Disc Embeddings of Planar Graphs

Zhi-Zhong Chen
Dept. of Mathematical Sciences
Tokyo Denki University
Hatoyama, Saitama 350-0394, Japan
Email: chen@r.dendai.ac.jp

Xin He *
Dept. of Computer Science and Engin.
State Univ. of New York at Buffalo
Buffalo, NY 14260, USA
Email: xinhe@cse.buffalo.edu

Abstract

Given a planar graph $G = (V, E)$ and a rooted forest $\mathcal{F} = (V_{\mathcal{F}}, A_{\mathcal{F}})$ with leaf set V, we wish to decide whether G has a plane embedding $\mathcal{G}$ satisfying the following condition: There are $|V_{\mathcal{F}}| - |V|$ pairwise non-crossing Jordan curves in the plane one-to-one corresponding to the non-leaf vertices of $\mathcal{F}$ such that for every non-leaf vertex f of $\mathcal{F}$, the interior of the curve $\mathcal{J}_f$ corresponding to f contains all the leaf descendants of f in $\mathcal{F}$ but contains no other leaves of $\mathcal{F}$. This problem arises from theoretical studies in geographic database systems. It is unknown whether this problem can be solved in polynomial time. This paper presents an almost linear-time algorithm for a nontrivial special case where the set of leaf descendants of each non-leaf vertex f in $\mathcal{F}$ induces a connected subgraph of G.

1 Introduction

It is a fundamental problem in mathematics (e.g., see [5, 12, 15, 17, 19, 26]) to embed a graph into a given surface while optimizing certain objectives required by applications. (Throughout this paper, a graph may have multiple edges but may never have selfloops, while a simple graph always has neither.) A graph is *planar* if it can be embedded on the plane so that any pair of edges can only intersect at their endpoints; a *plane* graph is a planar one together with such an embedding. A classical variant of the problem is to test whether a given graph is planar and in case it is, to find a planar embedding. This planarity problem can be solved in linear time [1, 2, 17].

A *Jordan curve* is a continuous non-self-intersecting curve in the plane whose origin and terminus coincide. Let $\mathcal{J}$ be a Jordan curve. Consider the set of all points in the plane that do not lie on $\mathcal{J}$. This set consists of two topologically connected regions; one region is bounded while the other is not. The bounded and the unbounded regions are called the *interior* and the *exterior* of $\mathcal{J}$, respectively. The closure of the interior of $\mathcal{J}$ is called the *closed interior* of $\mathcal{J}$. Two Jordan curves *cross* if both the interior and the exterior of each curve contain a point of the other curve. Note that two non-crossing Jordan curves may share a point.

In this paper, we initiate the study of the following new planarity problem. Let $(G, \mathcal{F})$ be a *planar-graph forest (PGF) pair*, i.e., a pair of a planar graph $G = (V, E)$ and a rooted

*Research supported in part by NSF Grant CCR-9912418.

forest $\mathcal{F} = (V_\mathcal{F}, A_\mathcal{F})$ such that V is also the set of the leaves of $\mathcal{F}$. A plane embedding $\mathcal{G}$ of G *satisfies* $\mathcal{F}$ if there are $|V_\mathcal{F}| - |V|$ pairwise non-crossing Jordan curves in the plane one-to-one corresponding to the non-leaf vertices of $\mathcal{F}$ such that for every non-leaf vertex f of $\mathcal{F}$, the interior of the curve $\mathcal{J}_f$ corresponding to f contains all the leaf descendants of f in $\mathcal{F}$ but contains no other leaves of $\mathcal{F}$. For example, in Figure 1.1, $\mathcal{G}$ is a plane embedding of G satisfying $\mathcal{F}$. However, if we modify forest $\mathcal{F}$ in the figure by replacing edge $\{4, f_1\}$ with edge $\{4, f_3\}$, then G has no plane embedding satisfying $\mathcal{F}$.

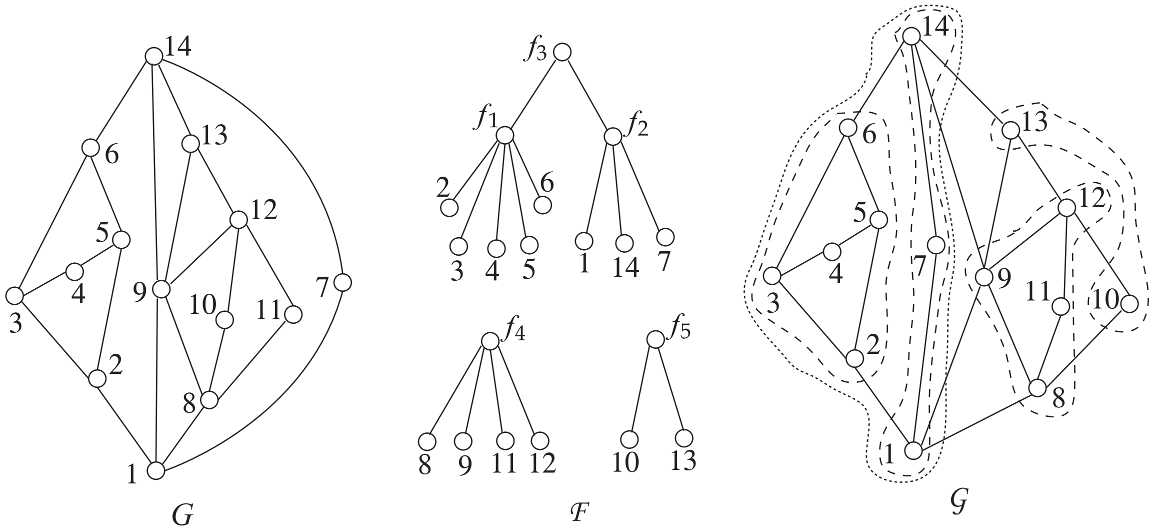

Figure 1.1: A plane graph G, a rooted forest $\mathcal{F}$, and a different embedding $\mathcal{G}$ of G satisfying $\mathcal{F}$. (The dotted curves are the required Jordan curves.)

Problem 1 (the disc-embedding problem)

- **Input**: A PGF pair $(G, \mathcal{F})$.
- **Question**: Does G have a plane embedding satisfying $\mathcal{F}$?

Unfortunately, we do not know whether the disc-embedding problem is NP-complete or solvable in polynomial time. Let n be the total number of vertices and edges in G and $\mathcal{F}$. The main result of this paper is an $O(n \log^2 n)$-time algorithm for a special case of the disc-embedding problem where the input PGF pair $(G, \mathcal{F})$ satisfies that for each non-leaf vertex f of $\mathcal{F}$, the set of all leaf descendants of f in $\mathcal{F}$ induces a connected subgraph of G. We call this special case the *connected disc-embedding* (CDE) problem. Our algorithm can actually find a plane embedding of G satisfying $\mathcal{F}$, if any exists. Note that the classical problem of simply finding a planar embedding is a further special case of the CDE problem where the forest in the input PGF pair has no edge at all. Therefore, the processing of the additional constraints imposed by $\mathcal{F}$ only incurs a quadratic-logarithmic factor of overhead.

The disc-embedding problem arises naturally from theoretical studies in geographic database systems [8, 9, 13, 20, 21]. In these studies, each geographical region managed by the database system is abstracted as a *closed disc homeomorph*, i.e., the closed interior of a Jordan curve. The database system needs to deal with certain basic topological relations such as the following:

1. Regions A and B are disjoint.
2. Regions A and B share only some isolated points of their boundaries.
3. Regions A and B share some curve segments (and possibly some isolated points) of their boundaries.
4. Region A contains region B as a subregion.

Given a set R and a topological relation between each pair of elements of R, we wish to (1) test if the elements of R can be drawn as closed disc homeomorphs in the plane in such a way that the given relations between them are satisfied, and (2) find such a drawing if so. This problem, known as the *topological inference problem*, has been studied extensively in the literature [8, 9, 13, 20, 21]. Despite much effort (and claims in the literature [25, 9]...), until recently it was open whether this problem is even decidable; in fact it is NP-complete [16, 22, 23]. On the other hand, in case the regions are districts on a political map, only the above four types of relations are possible and the distinction between the first two types of relations is not important and hence ignored. This political-map case of the problem is essentially the disc-embedding problem. To be a bit more specific, the districts on the map correspond to the vertices of the rooted forest $\mathcal{F}$ in the input PGF pair of the disc-embedding problem while the fourth-type relations between the districts on the map correspond to the edges of $\mathcal{F}$. The minimal districts on the map correspond to the vertices of the planar graph G in the input PGF pair of the disc-embedding problem while the third-type relations between the minimal districts correspond to the edges of G (and hence the first- or second-type relations between the minimal districts correspond to the edges of the complement of G). Thus, the CDE problem corresponds to the case where the minimal sub-districts of each district are connected by sharing boundaries. This case often occurs when the districts are small cities and their sub-districts.

The rest of the paper is organized as follows. Section 2 gives definitions needed in later sections. Section 3 reduces the CDE problem to its special case where the input planar graph G is biconnected. Section 4 presents an algorithm for this special case; the algorithm's implementation details are given in Section 5.

2 Definitions

Let $\mathcal{F}$ be a rooted forest $\mathcal{F}$. We view each vertex f in $\mathcal{F}$ as an ancestor and a descendant of f itself in $\mathcal{F}$. A *proper* ancestor (respectively, descendant) of f in $\mathcal{F}$ is an ancestor (respectively, descendant) f' of f in $\mathcal{F}$ with $f' \neq f$. A vertex f is *lower* (respectively, *higher*) than another vertex f' in $\mathcal{F}$ if f is a proper descendant (respectively, ancestor) of f' in $\mathcal{F}$. Let $f_1, \ldots, f_k$ be (possibly not distinct) vertices in $\mathcal{F}$. The *lowest common ancestor* of $f_1, \ldots, f_k$ in $\mathcal{F}$, denoted by $\mathrm{LCA}_{\mathcal{F}}(f_1, \ldots, f_k)$, is the lowest vertex in $\mathcal{F}$ that is an ancestor of f_1 through f_k in $\mathcal{F}$. Note that $\mathrm{LCA}_{\mathcal{F}}(f_1, \ldots, f_k)$ does not exist when no tree in $\mathcal{F}$ contains f_1 through f_k. We write $\mathrm{LCA}_{\mathcal{F}}(f_1, \ldots, f_k) = \perp$ when $\mathrm{LCA}_{\mathcal{F}}(f_1, \ldots, f_k)$ does not exist. For a vertex f of $\mathcal{F}$, let $\mathrm{Ds}_{\mathcal{F}}(f)$ denote the set of all descendants of f in $\mathcal{F}$; and let $\mathrm{Lf}_{\mathcal{F}}(f)$ denote the set of all leaves of $\mathcal{F}$ that are descendants of f.

Throughout this paper, a path or cycle in a graph is always simple, i.e., no vertex appears on it more than once. Let G be a graph. $V(G)$ and $E(G)$ denote the vertex-set and the edge-set of G, respectively. $|G|$ denotes the total number of vertices and edges in G. The *degree* of a vertex v in G is the number of edges incident to v in G. Let $E' \subseteq E(G)$ and $U \subseteq V(G)$. The *subgraph of G spanned by E'*, denoted by $G[E']$, is the graph $(V(G), E')$. $G - E'$ denotes the subgraph spanned by $E(G) - E'$. On the other hand, the *subgraph of G induced by E'* is the graph (V', E') where V' consists of the endpoints of the edges in E'. The *subgraph of G induced by U*, denoted by $G[U]$, is the graph (U, A_U) where A_U consists of all edges e in G such that both endpoints of e are contained in U. $G - U$ denotes $G[V(G) - U]$. If G is a *plane* graph, then each of $G[E']$, $G - E'$, $G[U]$, $G - U$, and the subgraph induced by E' *inherits* an embedding from G in the obvious way.

A *cut vertex* of G is a $v \in V(G)$ such that $G - \{v\}$ has more connected components than G. Similarly, a *bridge* of G is an $e \in E(G)$ such that $G - \{e\}$ has more connected components than G. G is *biconnected* if G is connected and has no cut vertex. A *biconnected component* of G is a maximal biconnected subgraph of G. For brevity, we call connected components *1-components*, and biconnected components *2-components*. A 1-component of G is *nontrivial* if it contains at least one edge. A 2-component of G is *nontrivial* if it contains at least two edges. Note that each nontrivial 2-component of G contains no bridge of G.

Let G be a plane graph. An *outer vertex* (respectively, *outer edge*) of G is a vertex (respectively, edge) on the boundary of the outer face of G. An *inner vertex* (respectively, *inner edge*) of G is a vertex (respectively, edge) that is not outer. An *outer non-bridge* of G is an outer edge that is not a bridge of G. Note that each outer non-bridge of G appears on the boundary of the outer face and also on the boundary of an inner face of G. For convenience, we define the *interior* of G to be the open set consisting of all points in the plane that do not belong to the outer face of G. Note that the interior of G is empty if and only if G is a forest. Moreover, the interior of G may be not topologically connected even if G is connected (for example, when a bridge of G is outer and connects two disjoint cycles of G). A *superface* of G is a plane subgraph H of G such that each inner face of H is also an inner face of G. Obviously, a plane subgraph H of G is a superface of G if and only if in the embedding G, no vertex of $V(G) - V(H)$ appears in the interior of H.

Let $(G, \mathcal{F})$ be a PGF pair. If G has two or more parallel edges between a pair of vertices, then we can delete all but one of the edges from G. Obviously, G has a plane embedding satisfying $\mathcal{F}$ before this modification if and only if G has a plane embedding satisfying $\mathcal{F}$ after this modification. Due to this reason, throughout the rest of this paper, a PGF pair always means one $(G, \mathcal{F})$ such that G is simple.

Let ψ be a partial function from a set X to a set Y. For each $x \in X$ such that $\psi(x)$ is undefined, we write $\psi(x) = \perp$. For a subset S of X, $\psi(S)$ denotes the set $\{\psi(x) \mid x \in S$ and $\psi(x) \neq \perp\}$. For each $y \in Y$, $\psi^{-1}(y)$ denotes the set $\{x \in X \mid \psi(x) = y\}$. For a subset S of Y, let $\psi^{-1}(S) = \cup_{y \in S} \psi^{-1}(y)$.

3 Reductions

This section reduces the CDE problem to its special case where the input graph is biconnected. For convenience, we say that a PGF pair $(G, \mathcal{F})$ is *connected* if for every non-leaf vertex f of $\mathcal{F}$, $G[\mathrm{Lf}_{\mathcal{F}}(f)]$ is connected. Recall that only connected PGF pairs are valid instances of the CDE problem. The following lemma gives an efficient algorithm for recognizing valid instances of the CDE problem.

Lemma 3.1 *Whether a PGF pair $(G, \mathcal{F})$ is connected can be decided in $O((|G| + |\mathcal{F}|)\log|\mathcal{F}|)$ time.*

Proof: Let $\mathcal{T}$ be a rooted tree. Define the *height* of a vertex f in $\mathcal{T}$ to be the number of edges on the longest path from f to a descendent of f in $\mathcal{T}$. The *height* of $\mathcal{T}$ is the height of its root. The *middle vertices* in $\mathcal{T}$ are those whose heights are $\lfloor \frac{h}{2} \rfloor$, where h is the height of $\mathcal{T}$.

Let $(G, \mathcal{T})$ be a PGF pair such that $\mathcal{T}$ is a rooted tree. As mentioned in the last section, we may assume that G is simple. Let h be the height of $\mathcal{T}$. If $h \leq 1$, then it is easy to decide whether $(G, \mathcal{T})$ is connected, in $O(|G|)$ time. So, suppose $h \geq 2$. Let $f_1, \ldots, f_k$ be the middle vertices in $\mathcal{T}$. For $i \in \{1, \ldots, k\}$, let $G_i = G[\mathrm{Lf}_{\mathcal{T}}(f_i)]$ and $\mathcal{T}_i$ be the subtree of $\mathcal{T}$ rooted at f_i. Let $A_0 = E(G) - (\cup_{i=1}^{k} E(G_i))$. The partition of $E(G)$ into $A_0, E(G_1), \ldots, E(G_k)$ can be done in $O(|G| + |\mathcal{F}|)$ time.

We test if each G_i is connected or not. If G_i is disconnected for some $i \in \{1, \ldots, k\}$, then $(G, \mathcal{T})$ is not connected. Otherwise, let G_0 be the graph obtained from G by merging the vertices in each $G_i \in \{G_1, \ldots, G_k\}$ into a single vertex f_i and deleting parallel edges and loops. G_0 is planar because G_1 through G_k are connected. Note that $G_0 = (\{f_1, \ldots, f_k\}, E_0')$, where E_0' is obtained from A_0 by identifying the vertices in each $G_i \in \{G_1, \ldots, G_k\}$ and further deleting parallel edges and loops. Let $\mathcal{T}_0$ be the rooted tree obtained from $\mathcal{T}$ by deleting all vertices whose heights are smaller than $\lfloor \frac{h}{2} \rfloor$.

For all $i \in \{0, 1, \ldots, k\}$, $(G_i, \mathcal{T}_i)$ is a PGF pair such that $\mathcal{T}_i$ is a rooted tree. So, we recursively decide whether $(G_i, \mathcal{T}_i)$ is connected for all $i \in \{0, 1, \ldots, k\}$. The maximum depth of recursion is $\lceil \log h \rceil = O(\log|\mathcal{F}|)$.

Note that $\sum_{i=0}^{k} |G_i|$ may equal to $|G| + k$. Since k can be large, it is not immediate that the overall time is $O((|G| + |\mathcal{F}|)\log|\mathcal{F}|)$. The key point is that for each $i \in \{1, \ldots, k\}$, f_i is neither a leaf nor the root in $\mathcal{T}$, but is a root in $\mathcal{T}_i$ and a leaf in $\mathcal{T}_0$. Thus, none of f_1 through f_k will become a middle vertex again in any recursive call. So, for every $d \in \{1, \ldots, \lceil \log h \rceil\}$, the total size of graphs in the PGF pairs at the dth depth of recursion is at most $2|G|$. In turn, the time needed at each depth of recursion is $O(|G|)$. Therefore, the overall time is $O((|G| + |\mathcal{F}|)\log|\mathcal{F}|)$.

Let $(G, \mathcal{F})$ be a PGF pair such that $\mathcal{F}$ is not a rooted tree. Let $\mathcal{T}_1, \ldots, \mathcal{T}_\ell$ be the rooted trees in $\mathcal{F}$, and $r_1, \ldots, r_\ell$ be their roots, respectively. We compute $G[\mathrm{Lf}_{\mathcal{F}}(r_1)], \ldots, G[\mathrm{Lf}_{\mathcal{F}}(r_\ell)]$; this takes $O(|G| + |\mathcal{F}|)$ time. Then, it suffices to decide whether all of $(G[\mathrm{Lf}_{\mathcal{F}}(r_1)], \mathcal{T}_1), \ldots, (G[\mathrm{Lf}_{\mathcal{F}}(r_\ell)], \mathcal{T}_\ell)$ are connected. ▮

Lemma 3.2 *Let $(G, \mathcal{F})$ be a connected PGF pair. Let $\mathcal{G}$ be a plane embedding of G. Then, $\mathcal{G}$ satisfies $\mathcal{F}$ if and only if for every non-leaf vertex f of $\mathcal{F}$, $\mathcal{G}[\mathrm{Lf}_{\mathcal{F}}(f)]$ is a superface of $\mathcal{G}$.*

Proof: The "only if" part is obvious. To see the "if" part, suppose that for every non-leaf vertex f of $\mathcal{F}$, $\mathcal{G}[\mathrm{Lf}_{\mathcal{F}}(f)]$ is a superface of $\mathcal{G}$. Obviously, for each edge e of $\mathcal{G}$, we can draw a Jordan curve $\mathcal{J}_e$ around e and close enough to e so that (1) no edge of $\mathcal{G}$ other than e and no vertex of $\mathcal{G}$ other than the endpoints of e appear in the interior of $\mathcal{J}_e$, (2) for every edge e' of $\mathcal{G}$ incident to no endpoint of e, $\mathcal{J}_e$ and $\mathcal{J}_{e'}$ share no point, and (3) for every edge e' of $\mathcal{G}$ incident to an endpoint of e, the intersection of the closed interiors of $\mathcal{J}_e$ and $\mathcal{J}_{e'}$ is a connected region containing the common endpoint of e and e'. (See Figure 3.1 for an example.) For each non-leaf vertex f of $\mathcal{F}$, the union of the interior of the superface $\mathcal{G}[\mathrm{Lf}_{\mathcal{F}}(f)]$ and the closed interiors of all $\mathcal{J}_e$ where e ranges over the outer edges of $\mathcal{G}[\mathrm{Lf}_{\mathcal{F}}(f)]$ is a connectd region; let $\mathcal{J}_f$ be the boundary of this region. Obviously, the vertices of $\mathrm{Lf}_{\mathcal{F}}(f)$ but no other leaves of $\mathcal{F}$ appear in the interior of $\mathcal{J}_f$. Moreover, if f and f' are distinct vertices of $\mathcal{F}$, then $\mathcal{J}_f$ and $\mathcal{J}_{f'}$ do not cross. So, $\mathcal{G}$ satisfies $\mathcal{F}$. ▮

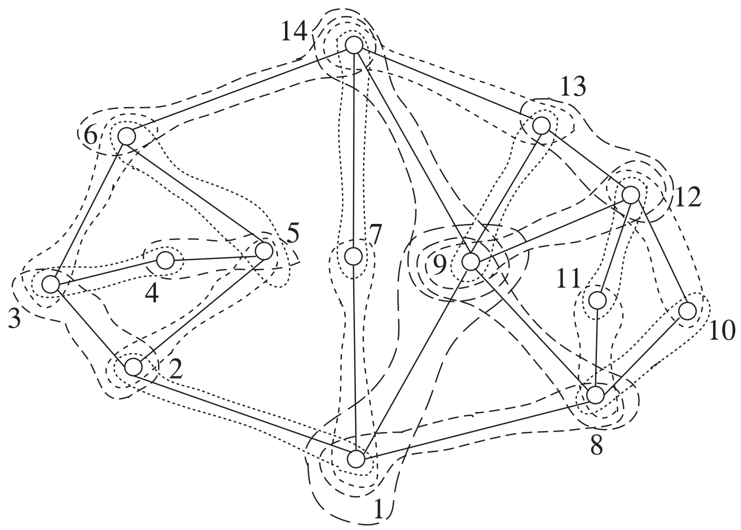

Figure 3.1: A plane graph $\mathcal{G}$. (The dotted curve surrounding each edge e is $\mathcal{J}_e$.)

By Lemma 3.2, the CDE problem can be redefined as follows.

Problem 2 (the connected disc-embedding (CDE) problem)

- **Input**: A connected PGF pair $(G, \mathcal{F})$.
- **Question**: Does G have a plane embedding $\mathcal{G}$ such that for every non-leaf vertex f of $\mathcal{F}$, $\mathcal{G}[\mathrm{Lf}_{\mathcal{F}}(f)]$ is a superface of $\mathcal{G}$.

Lemma 3.3 *Let $(G, \mathcal{F})$ be a connected PGF pair. Suppose that G is disconnected. Then, the answer of the CDE problem to the input $(G, \mathcal{F})$ is "yes" if and only if for every 1-component H of G, the answer of the CDE problem to the input $(H, \mathcal{F}_H)$ is "yes", where $\mathcal{F}_H$ is the forest consisting of all trees (1-components) in $\mathcal{F}$ whose leaves all are vertices of H too.*

Proof: Suppose that G has a plane embedding $\mathcal{G}$ such that for every non-leaf vertex f of $\mathcal{F}$, $\mathcal{G}[\mathrm{Lf}_{\mathcal{F}}(f)]$ is a superface F of $\mathcal{G}$. Since $\mathcal{G}[\mathrm{Lf}_{\mathcal{F}}(f)]$ is connected, some 1-component H of $\mathcal{G}$ contains all the vertices of F and every vertex in the interior of F in embedding $\mathcal{G}$ is also in H. Thus, for all 1-components H of $\mathcal{G}$, $\mathcal{H} = \mathcal{G}[V(H)]$ is a plane embedding of H such that for every non-leaf vertex f of $\mathcal{F}_H$, $\mathcal{H}[\mathrm{Lf}_{\mathcal{F}_H}(f)]$ is a superface of $\mathcal{H}$.

Conversely, suppose that for all 1-components H of G, H has a plane embedding $\mathcal{H}$ such that for every non-leaf vertex f of $\mathcal{F}_H$, $\mathcal{H}[\mathrm{Lf}_{\mathcal{F}_H}(f)]$ is a superface of $\mathcal{H}$. Then, combining the graphs H by identifying their outer faces yields a plane embedding $\mathcal{G}$ of G such that for every non-leaf vertex f of $\mathcal{F}$, $\mathcal{G}[\mathrm{Lf}_{\mathcal{F}}(f)]$ is a superface of $\mathcal{G}$. ▮

Lemma 3.3 gives a linear-time reduction of the CDE problem to its special case where the input planar graph is connected. We want to further reduce the CDE problem to its special case where the input planar graph is biconnected. In order to do this, we need to generalize the CDE problem as follows. Define a *graph-forest-set-function (GFSF)* quadruple to be a quadruple $(G, \mathcal{F}, \Gamma, \psi)$ satisfying the following three conditions:

- G is connected and $(G, \mathcal{F})$ is a connected PGF pair.
- Γ is a subset of $V(G)$ containing no cut vertex of G.
- ψ is a partial function mapping each $f \in \Gamma$ either to an ancestor of f in $\mathcal{F}$ or to nothing.

Let $(G, \mathcal{F}, \Gamma, \psi)$ be a GFSF quadruple. A plane embedding $\mathcal{G}$ of G *satisfies* $(\mathcal{F}, \Gamma, \psi)$ if the following holds:

- For every non-leaf vertex f of $\mathcal{F}$, $\mathcal{G}[\mathrm{Lf}_{\mathcal{F}}(f)]$ is a superface of $\mathcal{G}$ and every vertex of $\{y \in \Gamma \cap \mathrm{Lf}_{\mathcal{F}}(f) \mid \psi(y) = \perp \text{ or } \psi(y) \notin \mathrm{Ds}_{\mathcal{F}}(f)\}$ is an outer vertex of $\mathcal{G}[\mathrm{Lf}_{\mathcal{F}}(f)]$.

For example, the plane graph $\mathcal{G}$ in Figure 1.1 is an embedding of the planar graph G in Figure 1.1 satisfying $(\mathcal{F}, \Gamma, \psi)$, where $\mathcal{F}$ is the rooted forest in Figure 3.2, $\Gamma = \{4, 5, 10\}$, $\psi(4) = f_4$, $\psi(5) = f_6$, and $\psi(10) = \perp$.

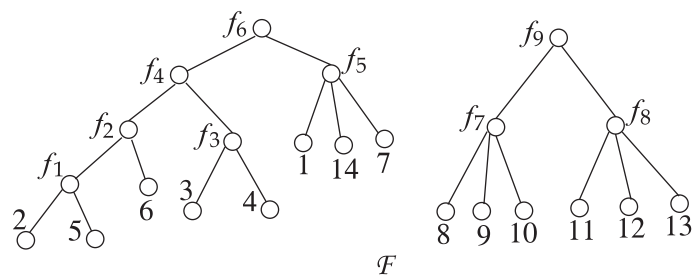

Figure 3.2: A rooted forest $\mathcal{F}$.

Now, we are ready to define:

Problem 3 (the generalized connected disc-embedding (GCDE) problem)

- **Input**: A GFSF quadruple $(G, \mathcal{F}, \Gamma, \psi)$.
- **Question**: Does G have a plane embedding satisfying $(\mathcal{F}, \Gamma, \psi)$?

In the remainder of this section, we show how to reduce the GCDE problem to its special case where the input planar graph G is biconnected. For convenience, we say that a vertex f of a rooted forest $\mathcal{F}$ is *uniparous* if f has exactly one child in $\mathcal{F}$. For a rooted forest $\mathcal{F}$ and a set $\mathcal{V}$ of vertices of $\mathcal{F}$, let $\mathcal{F}\langle\mathcal{V}\rangle$ be the forest obtained from $\mathcal{F}$ as follows: The vertex set of $\mathcal{F}\langle\mathcal{V}\rangle$ is $\mathcal{W} = \mathcal{V} \cup \{f \notin \mathcal{V} \mid f \text{ has at least two children } f' \text{ in } \mathcal{F} \text{ such that}$

$\mathrm{Ds}_{\mathcal{F}}(f') \cap \mathcal{V} \neq \emptyset\}$. A vertex $f \in \mathcal{W}$ is the parent of a vertex $f' \in \mathcal{W}$ in $\mathcal{F}\langle\mathcal{V}\rangle$ if and only if (1) f is an ancestor of f' in $\mathcal{F}$ and (2) no vertex in $\mathcal{W} - \{f, f'\}$ appears on the path between f and f' in $\mathcal{F}$. In Section 5, we will show that after an $O(|\mathcal{F}|)$-time preprocessing of $\mathcal{F}$, it takes $O(|\mathcal{V}| \log |\mathcal{F}|)$ time to compute $\mathcal{F}\langle\mathcal{V}\rangle$ for a given $\mathcal{V}$ (cf. Lemma 5.3).

Let x be a cut vertex of a graph G. Let $U_1, \ldots, U_k$ be the vertex-sets of the 1-components of $G - \{x\}$. $G[U_1 \cup \{x\}], \ldots, G[U_k \cup \{x\}]$ are called the *augmented components* of G induced by x. Let $\mathcal{F}$ be a rooted forest with leaf set $V(G)$. We say that a vertex f of $\mathcal{F}$ is *x-dangerous* if all except one of $U_1 \cup \{x\}, \ldots, U_k \cup \{x\}$ are subsets of $\mathrm{Lf}_{\mathcal{F}}(f)$.

Lemma 3.4 *Let $(G, \mathcal{F}, \Gamma, \psi)$ be a GFSF quadruple such that G contains a cut vertex x. Suppose that either $\mathcal{F}$ is not a rooted tree, or $\mathcal{F}$ is a rooted tree but the root of $\mathcal{F}$ is not uniparous. Let $G_1, \ldots, G_k$ be the augmented components of G induced by x. For each $i \in \{1, \ldots, k\}$, let $\Gamma_i = (\Gamma \cap V(G_i)) \cup \{x\}$. Then, the following hold:*

1. *Suppose that $\mathcal{F}$ has an x-dangerous vertex. Let f_d be the lowest x-dangerous vertex of $\mathcal{F}$. (Note that f_d must be unique because an x-dangerous vertex must be an ancestor of another x-dangerous vertex.) Without loss of generality, assume that $V(G_2), \ldots, V(G_k)$ are subsets of $\mathrm{Lf}_{\mathcal{F}}(f_d)$ but $V(G_1)$ is not. Then, a plane embedding $\mathcal{G}$ of G satisfies $(\mathcal{F}, \Gamma, \psi)$ if and only if the following hold:*

   (a) A plane embedding $\mathcal{G}_1$ of G_1 satisfies $(\mathcal{F}\langle V(G_1) \cup \psi_1(\Gamma_1)\rangle, \Gamma_1, \psi_1)$, where $\psi_1(x) = f_d$ and $\psi_1(y) = \psi(y)$ for all $y \in \Gamma_1 - \{x\}$.

   (b) For all $i \in \{2, \ldots, k\}$, a plane embedding $\mathcal{G}_i$ of G_i satisfies $(\mathcal{F}\langle V(G_i) \cup \psi_i(\Gamma_i)\rangle, \Gamma_i, \psi_i)$, where $\psi_i(x) = \perp$ and $\psi_i(y) = \psi(y)$ for all $y \in \Gamma_i - \{x\}$.

2. *Suppose that $\mathcal{F}$ has no x-dangerous vertex and $\mathcal{F}$ is a rooted tree. Let f_d be the root of $\mathcal{F}$. Assume that $\{y \in \Gamma \mid \psi(y) = \perp\}$ is nonempty and is a subset of $V(G_i)$ for some $i \in \{1, \ldots, k\}$. Without loss of generality, assume that $\{y \in \Gamma \mid \psi(y) = \perp\} \subseteq V(G_1)$. Then, a plane embedding $\mathcal{G}$ of G satisfies $(\mathcal{F}, \Gamma, \psi)$ if and only if Statements 1a and 1b hold.*

3. *Suppose that $\mathcal{F}$ has no x-dangerous vertex, and that $\mathcal{F}$ is not a rooted tree or $\{y \in \Gamma \mid \psi(y) = \perp\} = \emptyset$. Then, a plane embedding $\mathcal{G}$ of G satisfies $(\mathcal{F}, \Gamma, \psi)$ if and only if for all $i \in \{1, \ldots, k\}$, a plane embedding $\mathcal{G}_i$ of G_i satisfies $(\mathcal{F}\langle V(G_i) \cup \psi_i(\Gamma_i)\rangle, \Gamma_i, \psi_i)$, where $\psi_i(x) = \perp$ and $\psi_i(y) = \psi(y)$ for all $y \in \Gamma_i - \{x\}$.*

Proof: For each $i \in \{1, \ldots, k\}$, let $\mathcal{F}_i = \mathcal{F}\langle V(G_i) \cup \psi_i(\Gamma_i)\rangle$. A simple but useful observation is that for every $i \in \{1, \ldots, k\}$ and every non-leaf vertex f of $\mathcal{F}_i$, f is also a non-leaf vertex of $\mathcal{F}$ and $\{y \in \Gamma_i \cap \mathrm{Lf}_{\mathcal{F}_i}(f) \mid \psi_i(y) = \perp$ or $\psi_i(y) \notin \mathrm{Ds}_{\mathcal{F}_i}(f)\} \subseteq \{y \in \Gamma \cap \mathrm{Lf}_{\mathcal{F}}(f) \mid \psi(y) = \perp$ or $\psi(y) \notin \mathrm{Ds}_{\mathcal{F}}(f)\} \cup \{x\}$. We prove the statements in the lemma separately as follows.

Statement 1. We first prove the "only if" part. Suppose that a plane embedding $\mathcal{G}$ of G satisfies $(\mathcal{F}, \Gamma, \psi)$. For each $i \in \{1, \ldots, k\}$, let $\mathcal{G}_i = \mathcal{G}[V(G_i)]$. Fix an arbitrary $i \in \{1, \ldots, k\}$. We claim that $\mathcal{G}_i$ satisfies $(\mathcal{F}_i, \Gamma_i, \psi_i)$. To see this, consider a non-leaf vertex f of $\mathcal{F}_i$. $\mathcal{G}_i[\mathrm{Lf}_{\mathcal{F}_i}(f)]$ must be a superface of $\mathcal{G}_i$; otherwise, $\mathcal{G}[\mathrm{Lf}_{\mathcal{F}}(f)]$ would not be a superface of $\mathcal{G}$. It remains to prove that every vertex of $\{y \in \Gamma_i \cap \mathrm{Lf}_{\mathcal{F}_i}(f) \mid \psi_i(y) = \perp$ or $\psi_i(y) \notin \mathrm{Ds}_{\mathcal{F}_i}(f)\}$ is an outer vertex of $\mathcal{G}_i[\mathrm{Lf}_{\mathcal{F}_i}(f)]$. In turn, by the observation at the beginning of the proof, it suffices to prove that whenever $x \in \mathrm{Lf}_{\mathcal{F}_i}(f)$ and $x \notin \psi_i^{-1}(\mathrm{Ds}_{\mathcal{F}_i}(f))$,

x is an outer vertex of $\mathcal{G}_i[\mathrm{Lf}_{\mathcal{F}_i}(f)]$. So, suppose that $x \in \mathrm{Lf}_{\mathcal{F}_i}(f)$ and $x \notin \psi_i^{-1}(\mathrm{Ds}_{\mathcal{F}_i}(f))$. Note that x must be an outer vertex of each of $\mathcal{G}[V(G_2)], \ldots, \mathcal{G}[V(G_k)]$ or else $\mathcal{G}[\mathrm{Lf}_{\mathcal{F}}(f_d)]$ would not be a superface of $\mathcal{G}$ by planarity. Thus, if $i \notin \{2, \ldots, k\}$, then x is an outer vertex of $\mathcal{G}_i[\mathrm{Lf}_{\mathcal{F}_i}(f)]$. Assume that $i = 1$. Then, $x \in \mathrm{Lf}_{\mathcal{F}_1}(f)$ and $x \notin \psi_1^{-1}(\mathrm{Ds}_{\mathcal{F}_1}(f))$. Since $\psi_1(x) = f_d$, f must be a proper descendant of f_d in $\mathcal{F}_1$. In turn, f is a proper descendant of f_d in $\mathcal{F}$. So, x must be an outer vertex of $\mathcal{G}[\mathrm{Lf}_{\mathcal{F}}(f)]$; otherwise either f would be an x-dangerous vertex of $\mathcal{F}$ or $\mathcal{G}[\mathrm{Lf}_{\mathcal{F}}(f)]$ would not be a superface of $\mathcal{G}$ by planarity. Since $\mathcal{G}_1[\mathrm{Lf}_{\mathcal{F}_1}(f)]$ is a plane subgraph of $\mathcal{G}[\mathrm{Lf}_{\mathcal{F}}(f)]$, x is an outer vertex of $\mathcal{G}_1[\mathrm{Lf}_{\mathcal{F}_1}(f)]$.

We next prove the "if" part of Statement 1. Suppose that Statements 1a and 1b hold. For each $i \in \{2, \ldots, k\}$, since $\mathcal{F}_i$ is a rooted tree, x must be an outer vertex of $\mathcal{G}_i$. Let f_1 be the child of f_d in $\mathcal{F}_1$ that is an ancestor of x. $\mathcal{G}_1$ must have a face F_1 such that x appears on the boundary of F_1 and F_1 is not an inner face of $\mathcal{G}_1[\mathrm{Lf}_{\mathcal{F}_1}(f_1)]$; otherwise, x would be an inner vertex of $\mathcal{G}_1[\mathrm{Lf}_{\mathcal{F}_1}(f_1)]$. So, combining the graphs $\mathcal{G}_1$ through $\mathcal{G}_k$ by (1) putting $\mathcal{G}_2, \ldots, \mathcal{G}_k$ in face F_1 of $\mathcal{G}_1$ and (2) identifying all the appearances of vertex x, yields a plane embedding $\mathcal{G}$ of G satisfying $(\mathcal{F}, \Gamma, \psi)$.

Statement 2. The proof of the "only if" part is the same as that of Statement 1, except that the reason why x must be an outer vertex of each of $\mathcal{G}[V(G_2)], \ldots, \mathcal{G}[V(G_k)]$ is that all vertices in $\{y \in \Gamma \mid \psi(y) = \bot\}$ must be outer vertices of $\mathcal{G}$. The proof of the "if" part is the same as that of Statement 1.

Statement 3. We first prove the "only if" part. Suppose that a plane embedding $\mathcal{G}$ of G satisfies $(\mathcal{F}, \Gamma, \psi)$. For each $i \in \{1, \ldots, k\}$, let $\mathcal{G}_i = \mathcal{G}[V(G_i)]$. If x is an outer vertex of $\mathcal{G}$, then as in the proof of the "only if" part of Statement 1, we can easily see that $\mathcal{G}_i$ satisfies $(\mathcal{F}_i, \Gamma_i, \psi_i)$ for all $i \in \{1, \ldots, k\}$. So, assume that x is an inner vertex of $\mathcal{G}$. Then, by planarity, there is exactly one $i \in \{1, \ldots, k\}$ such that x is an inner vertex of $\mathcal{G}_i$. Without loss of generality, we may assume that x is an inner vertex of $\mathcal{G}_1$. Then, for all $i \in \{2, \ldots, k\}$, since x is an outer vertex of $\mathcal{G}_i$, we can prove that $\mathcal{G}_i$ satisfies $(\mathcal{F}_i, \Gamma_i, \psi_i)$ as in the proof of the "only if" part of Statement 1. We are done if $\mathcal{G}_1$ satisfies $(\mathcal{F}_1, \Gamma_1, \psi_1)$. So, assume that $\mathcal{G}_1$ does not satisfy $(\mathcal{F}_1, \Gamma_1, \psi_1)$. Then, similarly to the proof of the "only if" part of Statement 1, we can argue that x is an inner vertex of $\mathcal{G}_1[\mathrm{Lf}_{\mathcal{F}_1}(f_1)]$ where f_1 is the root ancestor of x in $\mathcal{F}_1$. So, $V(G) \subseteq \mathrm{Lf}_{\mathcal{F}}(f_1)$; otherwise, by planarity, either $\mathcal{G}[\mathrm{Lf}_{\mathcal{F}}(f_1)]$ would not be a superface of $\mathcal{G}$ or f_1 would be an x-dangerous vertex of $\mathcal{F}$. Consequently, $\mathcal{F}$ is a rooted tree. In turn, by assumption, $\{y \in \Gamma \mid \psi(y) = \bot\} = \emptyset$. Moreover, since $V(G) \subseteq \mathrm{Lf}_{\mathcal{F}}(f_1)$ and the root of $\mathcal{F}$ is not uniparous, f_1 is the root of $\mathcal{F}$.

Recall that root f_1 of $\mathcal{F}_1$ is not uniparous. Let f_2 be the child of f_1 in $\mathcal{F}_1$ that is an ancestor of x. There is at least one face F_1 in $\mathcal{G}_1$ such that x appears on the boundary of F_1 but F_1 is not an inner face of $\mathcal{G}_1[\mathrm{Lf}_{\mathcal{F}_1}(f_2)]$; otherwise, by planarity, either $\mathcal{G}[\mathrm{Lf}_{\mathcal{F}}(f_2)]$ would not be a superface of $\mathcal{G}$ or f_2 would be an x-dangerous vertex of $\mathcal{F}$. We construct a plane embedding $\mathcal{G}_1'$ of G_1 by modifying $\mathcal{G}_1$ by letting F_1 be the outer face and the original outer face be an inner face while keeping the rest faces unchanged. We claim that $\mathcal{G}_1'$ satisfies $(\mathcal{F}_1, \Gamma_1, \psi_1)$. To see this, first note that $\mathcal{G}_1'[\mathrm{Lf}_{\mathcal{F}_1}(f_1)]$ is $\mathcal{G}_1'$ itself, and every vertex of $\{y \in \Gamma_1 \cap \mathrm{Lf}_{\mathcal{F}_1}(f_1) \mid \psi_1(y) = \bot \text{ or } \psi_1(y) \notin \mathrm{Ds}_{\mathcal{F}_1}(f_1)\}$ is an outer vertex of $\mathcal{G}_1'[\mathrm{Lf}_{\mathcal{F}_1}(f_1)]$ because f_1 is the root of tree $\mathcal{F}_1$ and $\{y \in \Gamma_1 \mid \psi_1(y) = \bot\}$ consists of x only which is an outer vertex of $\mathcal{G}_1'$. Consider an arbitrary non-leaf non-root vertex f of $\mathcal{F}_1$. Note that for every cycle C of $\mathcal{G}_1$, if the closed interior of C in embedding $\mathcal{G}_1$ does not contain face F_1, then the interior of C in embedding $\mathcal{G}_1$ is the same as that in $\mathcal{G}_1'$. Combining this with the choice of F_1 and the fact that x is an outer vertex of $\mathcal{G}_1'$, we have that (1) $\mathcal{G}_1'[\mathrm{Lf}_{\mathcal{F}_1}(f)]$

is a superface of $\mathcal{G}'_1$ if and only if $\mathcal{G}_1[\mathrm{Lf}_{\mathcal{F}_1}(f)]$ is a superface of $\mathcal{G}_1$, and (2) every vertex of $\{y \in \Gamma_1 \cap \mathrm{Lf}_{\mathcal{F}_1}(f) \mid \psi_1(y) = \perp \text{ or } \psi_1(y) \notin \mathrm{Ds}_{\mathcal{F}_1}(f)\}$ is an outer vertex of $\mathcal{G}'_1[\mathrm{Lf}_{\mathcal{F}_1}(f)]$ if and only if every vertex of $\{y \in \Gamma_1 \cap \mathrm{Lf}_{\mathcal{F}_1}(f) \mid \psi_1(y) = \perp \text{ or } \psi_1(y) \notin \mathrm{Ds}_{\mathcal{F}_1}(f)\}$ is an outer vertex of $\mathcal{G}_1[\mathrm{Lf}_{\mathcal{F}_1}(f)]$. Thus, $\mathcal{G}'_1[\mathrm{Lf}_{\mathcal{F}_1}(f)]$ is a superface of $\mathcal{G}'_1$ and every vertex of $\{y \in \Gamma_1 \cap \mathrm{Lf}_{\mathcal{F}_1}(f) \mid \psi_1(y) = \perp \text{ or } \psi_1(y) \notin \mathrm{Ds}_{\mathcal{F}_1}(f)\}$ is an outer vertex of $\mathcal{G}'_1[\mathrm{Lf}_{\mathcal{F}_1}(f)]$. This finishes the proof of the "only if" part of Statement 3.

We next prove the "if" part of Statement 3. Suppose that for all $i \in \{1, \ldots, k\}$, a plane embedding $\mathcal{G}_i$ of G_i satisfies $(\mathcal{F}_i, \Gamma_i, \psi_i)$. If $\mathcal{F}$ is a rooted tree, then so are $\mathcal{F}_1$ through $\mathcal{F}_k$; in turn, x must be an outer vertex of $\mathcal{G}_1$ through $\mathcal{G}_k$, and we can obtain a plane embedding of G satisfying $(\mathcal{F}, \Gamma, \psi)$ by combining the graphs $\mathcal{G}_1$ through $\mathcal{G}_k$ by (1) identifying their outer faces and (2) identifying all the appearances of vertex x. So, assume that $\mathcal{F}$ is not a rooted tree.

Fix an arbitrary $i \in \{1, \ldots, k\}$ such that x is an inner vertex of $\mathcal{G}_i$. Let f_i be the root ancestor of x in $\mathcal{F}_i$. Since $\mathcal{G}_i$ satisfies $(\mathcal{F}_i, \Gamma_i, \psi_i)$, x is an outer vertex of superface $\mathcal{G}_i[\mathrm{Lf}_{\mathcal{F}_i}(f_i)]$. So, there is at least one face F_i in $\mathcal{G}_i$ such that x appears on the boundary of F_i but F_i is not an inner face of $\mathcal{G}_i[\mathrm{Lf}_{\mathcal{F}_i}(f_i)]$. We construct a plane embedding $\mathcal{G}'_i$ of G_i by modifying $\mathcal{G}_i$ by letting F_i be the outer face and the original outer face be an inner face while keeping the rest faces unchanged. We claim that $\mathcal{G}'_i$ satisfies $(\mathcal{F}_i, \Gamma_i, \psi_i)$. The proof of this claim is similar to that of the claim (that $\mathcal{G}'_1$ satisfies $(\mathcal{F}_1, \Gamma_1, \psi_1)$) in the "only if" part of Statement 3.

For convenience, let $\mathcal{G}'_i = \mathcal{G}_i$ for each $i \in \{1, \ldots, k\}$ such that x is an outer vertex of $\mathcal{G}_i$. Then, combining the graphs $\mathcal{G}'_1$ through $\mathcal{G}'_k$ by (1) identifying their outer faces and (2) identifying all the appearances of vertex x, yields a plane embedding $\mathcal{G}$ of G satisfying $(\mathcal{F}, \Gamma, \psi)$. This finishes the proof of the "if" part of Statement 3. ▮

Lemma 3.5 *Given a GFSF quadruple $(G, \mathcal{F}, \Gamma, \psi)$, we can construct another GFSF quadruple $(G, \mathcal{F}', \Gamma, \psi')$ in linear time such that (1) no vertex of $\mathcal{F}'$ is uniparous and (2) G has a plane embedding satisfying $(\mathcal{F}, \Gamma, \psi)$ if and only if G has a plane embedding satisfying $(\mathcal{F}', \Gamma, \psi')$.*

Proof: We initialize $\mathcal{F}' = \mathcal{F}$ and $\psi' = \psi$. Suppose that there is a path P in $\mathcal{F}'$ such that

- one endpoint f_1 of P is an ancestor of the other endpoint f_2 of P,
- f_1 is either a uniparous root of $\mathcal{F}'$, or a vertex having two or more children in $\mathcal{F}'$,
- f_2 is either a leaf vertex of $\mathcal{F}'$ or has two or more children in $\mathcal{F}'$, and
- every vertex on P other than f_1 and f_2 is uniparous.

If f_1 is a uniparous root of $\mathcal{F}'$, then we delete the vertices in $V(P) - \{f_2\}$ from $\mathcal{F}'$, and re-define $\psi'(x) = \perp$ for all $x \in \cup_{f_3 \in V(P) - \{f_2\}} \psi'^{-1}(f_3)$. Otherwise, we replace P by a single edge $\{f_1, f_2\}$, and re-define $\psi'(x) = f_1$ for all $x \in \cup_{f_3 \in V(P) - \{f_1, f_2\}} \psi'^{-1}(f_3)$. Obviously, G has a plane embedding satisfying $(\mathcal{F}, \Gamma, \psi)$ if and only if G has a plane embedding satisfying $(\mathcal{F}', \Gamma, \psi')$. We can repeat these modifications of $\mathcal{F}'$ and ψ' until $\mathcal{F}'$ has no uniparous vertex. The total time needed for all the modifications is linear. ▮

Lemma 3.6 *Suppose that $(G, \mathcal{F}, \Gamma, \psi)$ is a GFSF quadruple such that $\mathcal{F}$ is a rooted tree and $|\{x \in \Gamma \mid \psi(x) = \perp\}| \geq 2$. Then, the following hold:*

1. *Let G' be the graph obtained from G by adding a new vertex z and connecting z to every $x \in \Gamma$ with $\psi(x) = \perp$ via new edges. If G' is not planar, then G has no plane embedding satisfying $(\mathcal{F}, \Gamma, \psi)$. Moreover, z is not a cut vertex of G'.*

2. *Suppose that G' is planar. Then, we can construct another GFSF quadruple $(G', \mathcal{F}', \Gamma', \psi')$ in linear time such that (1) $\mathcal{F}'$ is a rooted tree without uniparous vertices, (2) $|\{x \in \Gamma' \mid \psi'(x) = \perp\}| = 1$, and (3) G has a plane embedding satisfying $(\mathcal{F}, \Gamma, \psi)$ if and only if G' has a plane embedding satisfying $(\mathcal{F}', \Gamma', \psi')$.*

Proof: Since $\mathcal{F}$ is a rooted tree, all vertices $x \in \Gamma$ with $\psi(x) = \perp$ must be outer vertices in every plane embedding of G satisfying $(\mathcal{F}, \Gamma, \psi)$. This implies Statement 1 immediately. The GFSF quadruple $(G', \mathcal{F}', \Gamma', \psi')$ required in Statement 2 is constructed as follows. $\mathcal{F}'$ is the rooted tree obtained from $\mathcal{F}$ by adding vertex z and a new root r and letting z and the original root be the children of r. Γ' is obtained from Γ by adding z and removing all vertex y with $\psi(y) = \perp$. $\psi'(z) = \perp$ and $\psi'(x) = \psi(x)$ for all $x \in \Gamma' - \{z\}$. If G has a plane embedding $\mathcal{G}$ satisfying $(\mathcal{F}, \Gamma, \psi)$, then the plane embedding of G' obtained from $\mathcal{G}$ by putting vertex z in the outer face and connecting it to all $x \in \Gamma$ with $\psi(x) = \perp$ satisfies $(\mathcal{F}', \Gamma', \psi')$. Conversely, if G' has a plane embedding $\mathcal{G}'$ satisfying $(\mathcal{F}', \Gamma', \psi')$, then z must be an outer vertex of $\mathcal{G}'$ and the plane embedding of G obtained from $\mathcal{G}'$ by deleting z and the edges incident to z satisfies $(\mathcal{F}, \Gamma, \psi)$. ▮

Using Lemmas 3.4, 3.5 and 3.6, we have a quadratic-time reduction from the GCDE problem to its special case where the input planar graph is biconnected. We next show a more efficient reduction.

For a connected graph G, let $\Phi(G)$ denote the bipartite graph whose vertices are the cut vertices and the 2-components of G and whose edges are those $\{x, B\}$ such that x is a cut vertex of G, B is a 2-component of G, and $x \in V(B)$. It is widely known that $\Phi(G)$ is indeed a tree. We call $\Phi(G)$ the *2-decomposition tree* of G.

Theorem 3.7 *Let n be the size of the input to the GCDE problem. Then, there is an $O(n \log^2 n)$-time reduction from the GCDE problem to its special case where the input graph is biconnected.*

Proof: The reduction is based on Lemma 3.4. Let $(G, \mathcal{F}, \Gamma, \psi)$ be a GFSF quadruple such that G is not biconnected. By applying Lemma 3.5, we may assume that no vertex of $\mathcal{F}$ is uniparous. Then, $|\mathcal{F}| = O(|G|)$. Moreover, by applying Lemma 3.6, we may further assume that if $\mathcal{F}$ is a rooted tree, then $|\{y \in \Gamma \mid \psi(y) = \perp\}| \le 1$.

As discussed in Section 5, we can process $\mathcal{F}$ in $O(|\mathcal{F}|)$ time so that after the processing, we can compute $\mathrm{LCA}_{\mathcal{F}}(f_1, f_2)$ in $O(1)$ time for two given vertices f_1 and f_2 of $\mathcal{F}$, and can compute $\mathcal{F}\langle\mathcal{V}\rangle$ in $O(|\mathcal{V}| \log |\mathcal{F}|)$ time for a given set $\mathcal{V}$ of vertices of $\mathcal{F}$. Hereafter, we assume that $\mathcal{F}$ has been so processed.

Using the 2-decomposition tree $\Phi(G)$ of G, we can compute a 2-component B_0 of G in linear time such that for every cut vertex x of G with $x \in V(B_0)$, each augmented component B of G induced by x with $V(B_0) \not\subseteq V(B)$ has at most one half of the cut vertices of G.

Consider the case where $\mathcal{F}$ is a rooted tree and $\{y \in \Gamma \mid \psi(y) = \perp\}$ consists of a unique vertex $y \notin V(B_0)$. A preprocessing is needed in this case. We describe the preprocessing first. Let $x \in V(B_0)$ be the cut vertex of G such that y is not contained in the augmented component B'_0 of G induced by x with $V(B_0) \subseteq V(B'_0)$. Let $B'_0, B'_1, \ldots, B'_q$ be the augmented components of G induced by x. We apply Lemma 3.4 to $(G, \mathcal{F}, \Gamma, \psi)$ with respect to x to construct a GFSF quadruple $(B'_j, \mathcal{F}'_j, \Gamma'_j, \psi'_j)$ for all $j \in \{0, 1, \ldots, q\}$. By Statements 1 and 2 in Lemma 3.4, $\{y \in \Gamma'_0 \mid \psi'_0(y) = \bot\} \subseteq \{x\}$. We then apply Lemmas 3.5 and 3.6 in this order to $(B'_0, \mathcal{F}'_0, \Gamma'_0, \psi'_0)$. After this, either we know that G has no plane embedding satisfying $(\mathcal{F}, \Gamma, \psi)$, or we have obtained a new GFSF quadruple $(B''_0, \mathcal{F}''_0, \Gamma''_0, \psi''_0)$ such that G has a plane embedding satisfying $(\mathcal{F}, \Gamma, \psi)$ if and only if (1) B''_0 has a plane embedding satisfying $(\mathcal{F}''_0, \Gamma''_0, \psi''_0)$ and (2) B'_j has a plane embedding satisfying $(\mathcal{F}'_j, \Gamma'_j, \psi'_j)$ for all $j \in \{1, \ldots, q\}$. In the former case, we are done. So, assume the latter case. The following observations follow from the proof of Lemmas 3.5 and 3.6 immediately:

1. $\mathcal{F}''_0$ is a rooted tree without uniparous vertices.

2. Either $\{y \in \Gamma''_0 \mid \psi''_0(y) = \bot\} = \emptyset$, or $\{y \in \Gamma''_0 \mid \psi''_0(y) = \bot\}$ consists of a unique vertex z and z is contained in the 2-component B of B''_0 with $V(B_0) \subseteq V(B)$. (Comment: $\{y \in \Gamma''_0 \mid \psi''_0(y) = \bot\}$ can be empty only when some non-root vertex f_d of $\mathcal{F}$ is x-dangerous and $V(B'_0)$ is not a subset of $\mathrm{Lf}_{\mathcal{F}}(f_d)$. In this case, the root of $\mathcal{F}'_0$ is not uniparous since the root of $\mathcal{F}$ is not uniparous.)

3. All vertices of B''_0 except z are also vertices of G, and all cut vertices of B''_0 are also cut vertices of G.

We further compute the 2-decomposition tree $\Phi(B''_0)$ of B''_0, and process $\mathcal{F}''_0$ in $O(|\mathcal{F}''_0|)$ time as discussed in Section 5 so that after the processing, we can compute $\mathrm{LCA}_{\mathcal{F}''_0}(f_1, f_2)$ in $O(1)$ time for two given vertices f_1 and f_2 of $\mathcal{F}''_0$, and can compute $\mathcal{F}''_0\langle\mathcal{V}\rangle$ in $O(|\mathcal{V}| \log |\mathcal{F}''_0|)$ time for a given set $\mathcal{V}$ of vertices of $\mathcal{F}''_0$. This finishes the description of the preprocessing. Obviously, the preprocessing takes $O(|G| \log |\mathcal{F}|)$ total time.

Next, we describe the common computations needed in all cases. To simplify the description, let $(H, \mathcal{F}_H, \Gamma_H, \psi_H)$ denote the GFSF quadruple $(B''_0, \mathcal{F}''_0, \Gamma''_0, \psi''_0)$ if $\mathcal{F}$ is a rooted tree and $\{y \in \Gamma \mid \psi(y) = \bot\}$ consists of a unique vertex $y \notin V(B_0)$, while let $(H, \mathcal{F}_H, \Gamma_H, \psi_H)$ denote the GFSF quadruple $(B, \mathcal{F}, \Gamma, \psi)$ otherwise.

Let C_0 be the 2-component of H with $V(B_0) \subseteq V(C_0)$. Root $\Phi(H)$ at C_0. Let $x_1, \ldots, x_p$ be the children of C_0 in $\Phi(H)$. For convenience, let $\ell_0 = 0$. For each $i \in \{1, \ldots, p\}$, let $C_{\ell_{i-1}+1}, \ldots, C_{\ell_i}$ be the augmented components of H induced by x_i that do not contain C_0 as a subgraph. Note that $\ell_i - \ell_{i-1}$ is the number of children of x_i in $\Phi(H)$. Obviously, we can order $x_1, \ldots, x_p$ so that the following hold:

- If $\mathcal{F}_H$ is not a rooted tree, then there is a $j \in \{\ell_{p-1}+1, \ldots, \ell_p\}$ such that some vertex $y_1 \in V(C_0)$ and some vertex $y_2 \in V(C_j)$ satisfy $\mathrm{LCA}_{\mathcal{F}_H}(y_1, y_2) = \bot$.

- If $\mathcal{F}_H$ is a rooted tree and the root of $\mathcal{F}_H$ is not the lowest common ancestor of all vertices of C_0, then there is a $j \in \{\ell_{p-1}+1, \ldots, \ell_p\}$ such that for some child f of the root of $\mathcal{F}_H$, $\mathrm{Lf}_{\mathcal{F}_H}(f) \cap V(C_0) = \emptyset$ and $\mathrm{Lf}_{\mathcal{F}_H}(f) \cap V(C_j) \neq \emptyset$.

Our strategy is to repeatedly apply Lemma 3.4 with respect to the cut vertices $x_1, \ldots, x_p$ in this order. We will see that this strategy works because of the preprocessing and the ordering of $x_1, \ldots, x_p$. Before applying Lemma 3.4 with respect to x_1, we make a copy $\mathcal{F}^c_H$ of $\mathcal{F}_H$. In the course of repeatedly applying Lemma 3.4 with respect to the cut vertices

$x_1, \dots, x_p$ in this order, $\mathcal{F}_H$ will never be changed, while $\mathcal{F}_H^c$ will be $\mathcal{F}_H\langle\mathcal{U}\rangle$ for some set $\mathcal{U}$ of vertices of $\mathcal{F}_H$ and will have no uniparous vertex. Note that for every pair of vertices f_1 and f_2 of $\mathcal{F}_H^c$, $\mathrm{LCA}_{\mathcal{F}_H^c}(f_1, f_2) = \mathrm{LCA}_{\mathcal{F}_H}(f_1, f_2)$.

Consider the application of Lemma 3.4 with respect to x_1. We first need to decide whether $\mathcal{F}_H^c$ has an x_1-dangerous vertex, and find the lowest x_1-dangerous vertex d_1 in $\mathcal{F}_H^c$ if there is any. Moreover, in case d_1 exists, we also want to compute h_1 where $h_1 = 0$ if $V(C_1) \cup \cdots \cup V(C_{\ell_1}) \subseteq \mathrm{Lf}_{\mathcal{F}_H^c}(d_1)$, while $h_1 = j \in \{1, \dots, \ell_1\}$ if $V(C_j) \not\subseteq \mathrm{Lf}_{\mathcal{F}_H^c}(d_1)$. To this end, for each $j \in \{1, \dots, \ell_1\}$, we compute the lowest common ancestor f_j of all the vertices of C_j in $\mathcal{F}_H$ in $O(|V(C_j)|)$ time. We then distinguish three cases as follows:

Case 1: For two or more $j \in \{1, \dots, \ell_1\}$, $f_j = \perp$. Obviously, $\mathcal{F}_H^c$ has no x_1-dangerous vertex.

Case 2: For exactly one $j \in \{1, \dots, \ell_1\}$, $f_j = \perp$. Then, $\mathcal{F}_H^c$ is not a rooted tree. We find all *C_j-pure vertices* in $\mathcal{F}_H^c$, i.e., those vertices f in $\mathcal{F}_H^c$ such that $\mathrm{Lf}_{\mathcal{F}_H^c}(f) \subseteq V(C_j)$. This can be done in $O(|V(C_j)|)$ time. We further check if two or more roots of $\mathcal{F}_H^c$ are not C_j-pure. If two or more roots of $\mathcal{F}_H^c$ are not C_j-pure, then $\mathcal{F}_H^c$ has no x_1-dangerous vertex and we are done. Otherwise, the root ancestor r of x_1 in $\mathcal{F}_H^c$ must be x_1-dangerous; so, we set $h_1 = j$. To find the lowest x_1-dangerous vertex in $\mathcal{F}_H^c$, starting at r and proceeding downwards, we try to find the lowest vertex f in $\mathcal{F}_H^c$ such that two or more children of f are not C_j-pure; f must be the lowest x_1-dangerous vertex in $\mathcal{F}_H^c$. In more details, if two or more children of r are not C_j-pure, then r is the lowest x_1-dangerous vertex in $\mathcal{F}_H^c$ and we are done. Otherwise, exactly one child r' of r in $\mathcal{F}_H^c$ is not C_j-pure; we proceed to r' and check if two or more children of r' are not C_j-pure.

Case 3: For all $j \in \{1, \dots, \ell_1\}$, f_j exists. Then, $\mathrm{LCA}_{\mathcal{F}_H}(f_1, \dots, f_{\ell_1})$ must exist because f_1 through f_{ℓ_1} are ancestors of x_1 in $\mathcal{F}_H$. Let $f = \mathrm{LCA}_{\mathcal{F}_H}(f_1, \dots, f_{\ell_1})$. If $\mathcal{F}_H^c$ is not a rooted tree, then f is the lowest x_1-dangerous vertex in $\mathcal{F}_H^c$ and we are done by setting $h_1 = 0$. So, suppose that $\mathcal{F}_H^c$ is a rooted tree. Let r be the root of $\mathcal{F}_H^c$. If r appears in $f_1, \dots, f_{\ell_1}$ at least twice, then $\mathcal{F}_H^c$ has no x_1-dangerous vertex and we are done. Moreover, if r does not appear in $f_1, \dots, f_{\ell_1}$, then f is the lowest x_1-dangerous vertex in $\mathcal{F}_H^c$ and we are done by setting $h_1 = 0$. So, we assume that r appears in $f_1, \dots, f_{\ell_1}$ exactly once. Consider the $j \in \{1, \dots, \ell_1\}$ with $f_j = r$. As in Case 2, we find all C_j-pure vertices in $\mathcal{F}_H^c$. After this, starting at r and proceeding downwards, we try to find the lowest vertex f' in $\mathcal{F}_H^c$ such that two or more children of f' are not C_j-pure. If $f' = r$, then $\mathcal{F}_H^c$ has no x_1-dangerous vertex; otherwise, f' is the lowest x_1-dangerous vertex in $\mathcal{F}_H^c$ and we are done by setting $h_1 = j$.

Now, after knowing d_1 and h_1, we can apply Lemma 3.4 to the GFSF quadruple $(H, \mathcal{F}_H, \Gamma_H, \psi_H)$ with respect to x_1. This yields a GFSF quadruple $(C_j, \mathcal{F}_j, \Gamma_j, \psi_j)$ for each $j \in \{1, \dots, \ell_1\}$, and a GFSF quadruple $(H_1, \mathcal{F}_{H_1}, \Gamma_{H_1}, \psi_{H_1})$ where H_1 is the augmented component of H induced by x_1 with $V(C_0) \subseteq V(H_1)$. Obviously, H_1, Γ_{H_1}, ψ_{H_1}, and $(C_1, \mathcal{F}_1, \Gamma_1, \psi_1)$ through $(C_{\ell_1}, \mathcal{F}_{\ell_1}, \Gamma_{\ell_1}, \psi_{\ell_1})$ can be obtained in $O(\sum_{j=1}^{\ell_1} |C_j| \log |\mathcal{F}_H|)$ total time. Instead of computing $\mathcal{F}_{H_1}$, we modify $\mathcal{F}_H^c$ in $O(\sum_{j=1}^{\ell_1} |C_j|)$ time by performing the following steps in turn:

- Delete all vertices f with $\mathrm{Lf}_{\mathcal{F}_H^c}(f) \subseteq \cup_{j=1}^{\ell_1} V(C_j)$.
- Delete every maximal path P such that one endpoint of P is a root of $\mathcal{F}_H^c$ and the rest vertices of P are uniparous vertices of $\mathcal{F}_H^c$.

- Replace every maximal path P whose internal vertices all are uniparous vertices of $\mathcal{F}^c_H$, with a single edge.

The point is that either $\mathcal{F}_{H_1}$ is not a rooted tree, or $\mathcal{F}_{H_1}$ is a rooted tree but its root is not uniparous. This is because of the preprocessing and the ordering of $x_1, \ldots, x_p$. Moreover, due to the same reason and Lemma 3.4, when $\mathcal{F}_{H_1}$ is a rooted tree and $\{y \in \Gamma_{H_1}(y) = \perp\} \neq \emptyset$, $\{y \in \Gamma_{H_1}(y) = \perp\}$ must be a subset of $V(C_0)$. So, we can again apply Lemma 3.4 to $(H_1, \mathcal{F}_{H_1}, \Gamma_{H_1}, \psi_{H_1})$ with respect to the cut vertex x_2. We want to do this in the same way as we applied Lemma 3.4 to $(H, \mathcal{F}_H, \Gamma_H, \psi_H)$ with respect to the cut vertex x_1. To see that this is possible, first observe that for every pair of vertices f_1 and f_2 of $\mathcal{F}_{H_1}$, $\mathrm{LCA}_{\mathcal{F}_{H_1}}(f_1, f_2) = \mathrm{LCA}_{\mathcal{F}_H}(f_1, f_2)$. Moreover, since only uniparous vertices of $\mathcal{F}_{H_1}$ are missing in $\mathcal{F}^c_H$, each x_2-dangerous vertex of $\mathcal{F}_{H_1}$ is also an x_2-dangerous vertex of $\mathcal{F}^c_H$, and vice versa. Thus, we can apply Lemma 3.4 to $(H_1, \mathcal{F}_{H_1}, \Gamma_{H_1}, \psi_{H_1})$ with respect to x_2 in the same way as we did to $(H, \mathcal{F}_H, \Gamma_H, \psi_H)$ with respect to x_1. Continuing this argument, we can claim that for every $i \in \{1, \ldots, p-1\}$, after applying Lemma 3.4 to $(H_i, \mathcal{F}_{H_i}, \Gamma_{H_i}, \psi_{H_i})$ with respect to x_{i+1}, we can again apply Lemma 3.4 to the GFSF quadruple constructed for the augmented component H_{i+1} of H_i induced by x_{i+1} with $V(C_0) \subseteq V(H_{i+1})$, with respect to x_{i+2}.

After applying Lemma 3.4 with respect to x_p, we have $(H_p, \mathcal{F}_{H_p}, \Gamma_{H_p}, \psi_{H_p})$. Note that $H_p = C_0$ and $\mathcal{F}_{H_p} = \mathcal{F}_H\langle V(C_0) \cup \psi_{H_p}(\Gamma_{H_p})\rangle$. For convenience, let $(C_0, \mathcal{F}_0, \Gamma_0, \psi_0) = (H_p, \mathcal{F}_{H_p}, \Gamma_{H_p}, \psi_{H_p})$.

In summary, by Lemma 3.4, G has a plane embedding satisfying $(\mathcal{F}, \Gamma, \psi)$ if and only if (1) B'_j has a plane embedding satisfying $(\mathcal{F}'_j, \Gamma'_j, \psi'_j)$ for each $j \in \{1, \ldots, q\}$ and (2) C_j has a plane embedding satisfying $(\mathcal{F}_j, \Gamma_j, \psi_j)$ for each $j \in \{0, 1, \ldots, \ell_p\}$. Moreover, the GFSF quadruples $(B'_1, \mathcal{F}'_1, \Gamma'_1, \psi'_1)$ through $(B'_q, \mathcal{F}'_q, \Gamma'_q, \psi'_q)$ and $(C_0, \mathcal{F}_0, \Gamma_0, \psi_0)$ through $(C_{\ell_p}, \mathcal{F}_{\ell_p}, \Gamma_{\ell_p}, \psi_{\ell_p})$ can be obtained in $O((\sum_{j=1}^{q} |B'_j| + \sum_{j=0}^{\ell_p} |C_j|) \log |\mathcal{F}|) = O(|G| \log |\mathcal{F}|)$ total time.

Observe that each cut vertex of G with respect to which we already applied Lemma 3.4 is not a cut vertex of any graph in $\{B'_1, \ldots, B'_q\} \cup \{C_0, \ldots, C_{\ell_p}\}$. Note that C_0 has no cut vertex but some of B'_1 through B'_q and C_1 through C_{ℓ_p} may have. For each graph in $\{B'_1, \ldots, B'_q\} \cup \{C_1, \ldots, C_{\ell_p}\}$ that has a cut vertex, we recursively apply Lemma 3.4 with respect to the cut vertices of the graph. The recursive calls can be organized into a rooted tree $T_{G,\mathcal{F},\Gamma,\psi}$. The root of $T_{G,\mathcal{F},\Gamma,\psi}$ is $(G, \mathcal{F}, \Gamma, \psi)$. Each internal node of $T_{G,\mathcal{F},\Gamma,\psi}$ is a GFSF quadruple $(G', \mathcal{F}', \Gamma', \psi')$; its children are the GFSF quadruples that are obtained from $(G', \mathcal{F}', \Gamma', \psi')$ by recursively applying Lemma 3.4 with respect to some cut vertices of G'. For example, the children of the root are $(B'_1, \mathcal{F}'_1, \Gamma'_1, \psi'_1)$ through $(B'_q, \mathcal{F}'_q, \Gamma'_q, \psi'_q)$ and $(C_0, \mathcal{F}_0, \Gamma_0, \psi_0)$ through $(C_{\ell_p}, \mathcal{F}_{\ell_p}, \Gamma_{\ell_p}, \psi_{\ell_p})$. Each leaf of $T_{G,\mathcal{F},\Gamma,\psi}$ is a GFSF quadruple in which the graph has no cut vertex. Since the recursion depth is $O(\log |G|)$, $T_{G,\mathcal{F},\Gamma,\psi}$ has $O(\log |G|)$ levels. The root of $T_{G,\mathcal{F},\Gamma,\psi}$ is at level 1. For each integer $i \geq 1$, the children of each GFSF quadruple at level i are at level $i + 1$. Let t be the total number of vertices in the 2-components of G. Note that the total number of vertices in B'_1 through B'_q and C_0 through C_{ℓ_p} does not exceed $t + 2$. From this fact, it is easy to see that for every level $i \geq 2$, the total number of vertices in the graphs in the GFSF quadruples at level i does not exceed t plus twice the number of GFSF quadruples at levels 1 through $i - 1$. It is well known that $t = O(|G|)$ (see, e.g., [3]). In turn, $|\Phi(G)| = O(|G|)$. Obviously, the total number of GFSF quadruples in $T_{G,\mathcal{F},\Gamma,\psi}$ does not exceed $|\Phi(G)|$. Thus, the computations at each level take $O(|G| \log |\mathcal{F}|)$ total time. Consequently, the overall computation takes $O(|G| \log |\mathcal{F}| \log |G|)$ time. ∎

4 Biconnected case

Throughout the rest of this paper, fix a GFSF quadruple $(G, \mathcal{F}, \Gamma, \psi)$ such that G is biconnected. We want to check if G has a plane embedding satisfying $(\mathcal{F}, \Gamma, \psi)$.

4.1 SPQR decompositions

A *two-terminal graph* [1] (TTG) $\mathcal{G} = (V, E)$ is an acyclic plane digraph with a pair (s, t) of specified vertices on the outer face such that s is the only *source* (i.e. with no entering edge) and t is the only *sink* (i.e. with no leaving edge). We denote s and t by $s(\mathcal{G})$ and $t(\mathcal{G})$, respectively. We also call s and t the *poles* of $\mathcal{G}$ and the rest vertices the *nonpoles* of $\mathcal{G}$. A *two-terminal subgraph* of $\mathcal{G}$ is a subgraph H of $\mathcal{G}$ such that H is a TTG and no edge of $\mathcal{G}$ is incident both to a nonpole of H and to a vertex outside H. As in previous studies, we visualize $\mathcal{G}$ in such a way that s is at the bottom, t is at the top, and each edge goes (possibly not vertically) upwards. Although $\mathcal{G}$ is a digraph, the directions on its edges are used only in the definition of its two-terminal subgraphs but have no meaning anywhere else. With this in mind, we will view $\mathcal{G}$ as an undirected graph from now on.

The poles of $\mathcal{G}$ divide the boundary of the outer face into two (possibly not disjoint) paths, which are called the *left side* and the *right side* of $\mathcal{G}$, respectively. A *side* of $\mathcal{G}$ is either the left or the right side of $\mathcal{G}$.

A *split pair* of $\mathcal{G}$ is either an unordered pair of adjacent vertices, or an unordered pair of vertices whose removal disconnects the graph $(V(\mathcal{G}), E(\mathcal{G}) \cup \{\{s(\mathcal{G}), t(\mathcal{G})\}\})$. A *split component* of a split pair $\{u, v\}$ is either an edge $\{u, v\}$ or a maximal two-terminal subgraph H (of $\mathcal{G}$) with poles u and v such that $\{u, v\}$ is not a split pair of H. A split pair $\{u, v\}$ of $\mathcal{G}$ is *maximal* if there is no other split pair $\{w_1, w_2\}$ in $\mathcal{G}$ such that $\{u, v\}$ is contained in a split component of $\{w_1, w_2\}$.

The *decomposition tree* T of $\mathcal{G}$ describes a recursive decomposition of $\mathcal{G}$ with respect to its split pairs [6, 7]. For clarity, we call the elements of $V(T)$ *nodes*. T is a rooted ordered tree whose nodes are of four types: S, P, Q, and R. Each node μ of T has an associated TTG, called the *skeleton* of μ and denoted by $\mathrm{Skl}(\mu)$. Also, it is associated with an edge in the skeleton of the parent χ of μ, called the *virtual edge* of μ in $\mathrm{Skl}(\chi)$. T is recursively defined as follows.

Trivial case: $\mathcal{G}$ consists of a single edge $\{s(\mathcal{G}), t(\mathcal{G})\}$. Then, T consists of a single Q-node whose skeleton is $\mathcal{G}$ itself.

Series case: $\mathcal{G}$ has one or more cut vertices. Let $x_1, \ldots, x_{k-1}$ $(k \geq 2)$ be the cut vertices of $\mathcal{G}$. Since $\mathcal{G}$ is a TTG, each $x_i \in \{x_1, \ldots, x_{k-1}\}$ is contained in exactly two 2-components $\mathcal{G}_i$ and $\mathcal{G}_{i+1}$ of $\mathcal{G}$ such that $s(\mathcal{G})$ is in $\mathcal{G}_1$ and $t(\mathcal{G})$ is in $\mathcal{G}_k$. The root of T is an S-node μ. $\mathrm{Skl}(\mu)$ is the path: $s(\mathcal{G}), e_1, x_1, e_2, \ldots, x_{k-1}, e_k, t(\mathcal{G})$.

Parallel case: $s(\mathcal{G})$ and $t(\mathcal{G})$ together constitute a split pair of $\mathcal{G}$ with split components $\mathcal{G}_1, \ldots, \mathcal{G}_k$ $(k \geq 2)$. Then, the root of T is a P-node μ. $\mathrm{Skl}(\mu)$ consists of k parallel edges between $s(\mathcal{G})$ and $t(\mathcal{G})$, denoted by $e_1, \ldots, e_k$.

[1] Previously called a *planar st-graph*.

Rigid case: None of the above cases applies. Let $\{s_1, t_1\}, \ldots, \{s_k, t_k\}$ be the maximal split pairs of $\mathcal{G}$. For $i = 1, \ldots, k$, let $\mathcal{G}_i$ be the union of all the split components of $\{s_i, t_i\}$. The root of T is an R-node μ. Skl(μ) is obtained from $\mathcal{G}$ by replacing each $\mathcal{G}_i \in \{\mathcal{G}_1, \ldots, \mathcal{G}_k\}$ with an edge $e_i = \{s_i, t_i\}$. Note that adding the edge $\{s(\mathcal{G}), t(\mathcal{G})\}$ to Skl(μ) yields a triconnected graph. Thus, the only embedding of Skl(μ) with both $s(\mathcal{G})$ and $t(\mathcal{G})$ on the outer face is itself [18].

In the last three cases, μ has children $\nu_1, \ldots, \nu_k$ (in this order), such that ν_i is the root of the decomposition tree of graph $\mathcal{G}_i$ for all $i \in \{1, \ldots, k\}$. The virtual edge of node ν_i is the edge e_i in Skl(μ). $\mathcal{G}_i$ is called the *pertinent graph* of ν_i and is denoted by Prt(ν_i). Note that $\mathcal{G}$ is the pertinent graph of the root of T.

Figure 4.1 illustrates the decomposition tree of the TTG G in Figure 1.1. The skeletons of the nodes μ and ν are also shown.

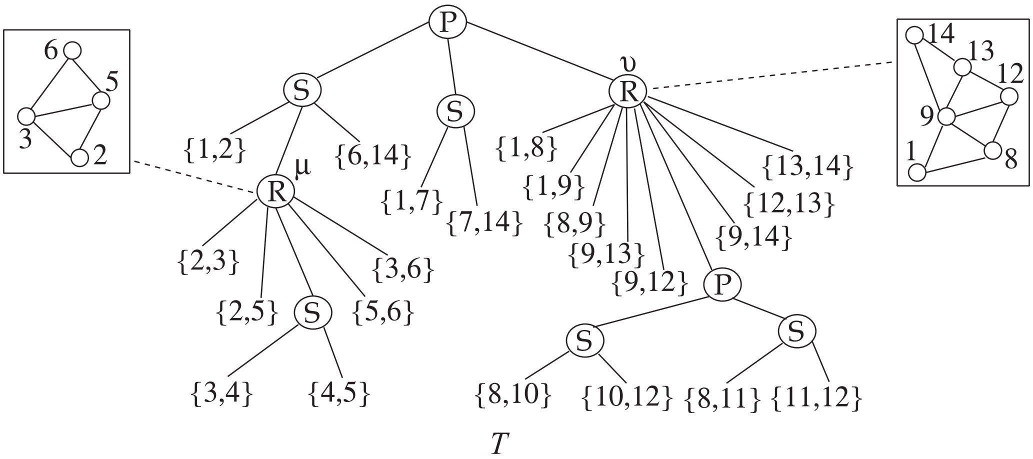

Figure 4.1: The decomposition tree T of the TTG G in Figure 1.1.

By the definition of T, no child of an S-node is an S-node and no child of a P-node is a P-node. Also note that, for an R-node μ, the two poles of Skl(μ) are not adjacent in Skl(μ). A *block* of $\mathcal{G}$ is the pertinent graph of a node in T. For a non-leaf node μ in T and each child ν of μ, we call Prt(ν) a *child block* of Prt(μ). The following fact is obvious.

Fact 4.1 *Each block of $\mathcal{G}$ is a superface of $\mathcal{G}$.*

Lemma 4.2 (see [7]) *T has $O(|\mathcal{G}|)$ nodes. Also, the total number of edges of the skeletons stored at the nodes of T is $O(|\mathcal{G}|)$.*

Lemma 4.3 *Let z_1 and z_2 be two vertices of G such that adding a new edge between z_1 and z_2 does not distroy the planarity of G. Then, it takes $O(|G|)$ time to convert G to a TTG $\mathcal{G}$ with poles z_1 and z_2 and to construct the corresponding decomposition tree T of $\mathcal{G}$. Moreover, every plane embedding of G whose outer face contains both z_1 and z_2 can be obtained by performing a sequence of $O(|G|)$ operations below:*

1. *Flip the skeleton of an R-node of T around the poles of the skeleton.*

2. *Permute the children of a P-node of T.*

Proof: The first assertion in the lemma is proved in [11], while the second is proved in [6, 7]. ▌

A *desired pair* for $(G, \mathcal{F}, \Gamma, \psi)$ is an unordered pair $\{z_1, z_2\}$ of two vertices of G such that if G has a plane embedding satisfying $(\mathcal{F}, \Gamma, \psi)$, then G has a plane embedding satisfying $(\mathcal{F}, \Gamma, \psi)$ whose outer face contains both z_1 and z_2.

Lemma 4.4 *Given $(G, \mathcal{F}, \Gamma, \psi)$, we can either find a desired pair $\{z_1, z_2\}$ for it in $O(|G| + |\mathcal{F}|)$ time, or find a new GFSF quadruple $(G, \mathcal{F}', \Gamma, \psi')$ and two vertices z_1', z_2' of G in $O(|G| + |\mathcal{F}|)$ time such that G has a plane embedding satisfying $(\mathcal{F}, \Gamma, \psi)$ if and only if G has a plane embedding satisfying $(\mathcal{F}', \Gamma, \psi')$ whose outer face contains both z_1' and z_2'. (Note that $\{z_1', z_2'\}$ is a desired pair for $(G, \mathcal{F}', \Gamma, \psi')$.)*

Proof: By Lemma 3.5, we may assume that no vertex of $\mathcal{F}$ has exactly one child. We distinguish two cases as follows:

Case 1: $\mathcal{F}$ is not a tree. Then, we can let $\{z_1, z_2\}$ be an edge of G such that there are two distinct roots f_1 and f_2 in $\mathcal{F}$ with $z_1 \in \mathrm{Lf}_{\mathcal{F}}(f_1)$ and $z_2 \in \mathrm{Lf}_{\mathcal{F}}(f_2)$. Note that if a plane embedding $\mathcal{G}$ of G satisfies $(\mathcal{F}, \Gamma, \psi)$ but edge $\{z_1, z_2\}$ is not outer, then we can modify $\mathcal{G}$ by letting an inner face with edge $\{z_1, z_2\}$ on its boundary be the outer face while letting the original outer face be an inner face; the resulting embedding still satisfies $(\mathcal{F}, \Gamma, \psi)$.

Case 2: $\mathcal{F}$ is a tree. We further distinguish three cases as follows.

Case 2.1: ψ is undefined for at least two elements of Γ. Then, we can let z_1 and z_2 be two such elements of Γ.

Case 2.2: ψ is defined for all elements of Γ. Then, we can let $\{z_1, z_2\}$ be an edge of G such that there are two distinct children f_1 and f_2 of the root of $\mathcal{F}$ with $z_1 \in \mathrm{Lf}_{\mathcal{F}}(f_1)$ and $z_2 \in \mathrm{Lf}_{\mathcal{F}}(f_2)$.

Case 2.3: ψ is undefined for exactly one element y of Γ. Let $\mathcal{F}'$ be the forest obtained from $\mathcal{F}$ by deleting the root r. Let ψ' be the function obtained from ψ by letting $\psi'(x) = \perp$ for all $x \in \psi^{-1}(r)$. Let $\{z_1', z_2'\}$ be an edge of G such that there are two distinct roots f_1 and f_2 in $\mathcal{F}'$ with $z_1' \in \mathrm{Lf}_{\mathcal{F}'}(f_1)$ and $z_2' \in \mathrm{Lf}_{\mathcal{F}'}(f_2)$. We claim that G has a plane embedding satisfying $(\mathcal{F}, \Gamma, \psi)$ if and only if G has a plane embedding satisfying $(\mathcal{F}', \Gamma, \psi')$ whose outer face contains both z_1' and z_2'.

To see the "only if" part of the claim, suppose that G has a plane embedding $\mathcal{G}$ satisfying $(\mathcal{F}, \Gamma, \psi)$. If both z_1' and z_2' are outer vertices of $\mathcal{G}$, then we are done because $\mathcal{G}$ trivially satisfies $(\mathcal{F}', \Gamma, \psi')$. Otherwise, we modify $\mathcal{G}$ by letting its outer face be an inner face and letting an inner face F be the outer face where the boundary of F contains both z_1' and z_2'. Obviously, the resulting embedding satisfies $(\mathcal{F}', \Gamma, \psi')$ and both z_1' and z_2' are outer vertices in the embedding. This establishes the "only if" part of the claim.

To see the "if" part of the claim, suppose that G has a plane embedding $\mathcal{G}'$ satisfying $(\mathcal{F}', \Gamma, \psi')$ such that both z_1' and z_2' are outer vertices of $\mathcal{G}'$. If y is an outer vertex of $\mathcal{G}'$, then $\mathcal{G}'$ trivially satisfies $(\mathcal{F}, \Gamma, \psi)$ and we are done. Otherwise, we modify $\mathcal{G}'$ by letting its outer face be an inner face and letting an inner face F' be the outer face where the boundary of F' contains both y and an edge connecting descendants of two distinct roots of $\mathcal{F}'$. (*Note:* If there were no such face F' in $\mathcal{G}'$, then y would be an inner vertex of $\mathcal{G}'[\mathrm{Lf}_{\mathcal{F}'}(f_r)]$ where f_r is the root ancestor of y in $\mathcal{F}'$, contradicting that $y \in \Gamma$, $\psi'(y) = \perp$ and $\mathcal{G}'$ satisfies $(\mathcal{F}', \Gamma, \psi')$.) Obviously, the resulting embedding satisfies $(\mathcal{F}, \Gamma, \psi)$. This completes the proof of the claim and hence the lemma. ▌

4.2 Basic ideas

Suppose that we have found a desired pair $\{z_1, z_2\}$ for $(G, \mathcal{F}, \Gamma, \psi)$ and have converted G to a TTG $\mathcal{G}$ with poles z_1 and z_2. Throughout Sections 4.2, 4.4, and 4.5, when we say that G has no plane embedding satisfying $(\mathcal{F}, \Gamma, \psi)$, we mean that G has no plane embedding satisfying $(\mathcal{F}, \Gamma, \psi)$ whose outer face contains both z_1 and z_2.

Let T be the decomposition tree of $\mathcal{G}$. Our algorithm processes the nodes of T in post-order. Processing a node μ is to perform Operations 1 and 2 in Lemma 4.3 on the subtree of T rooted at μ.

Before proceeding to the details of processing nodes, we need several definitions. For a block B of $\mathcal{G}$, $\mathcal{G} \ominus B$ denotes the plane subgraph of $\mathcal{G}$ obtained from $\mathcal{G}$ by deleting all vertices in $V(B) - \{s(B), t(B)\}$ and all edges in $E(B)$. For a plane subgraph H of $\mathcal{G}$ and a vertex f of $\mathcal{F}$, $H \cap \mathrm{Lf}_{\mathcal{F}}(f)$ denotes the plane subgraph of H induced by $V(H) \cap \mathrm{Lf}_{\mathcal{F}}(f)$.

Let μ be a node of T. Let $B = \mathrm{Prt}(\mu)$. A vertex f in $\mathcal{F}$ is *μ-through* if $B \cap \mathrm{Lf}_{\mathcal{F}}(f)$ contains a path between $s(B)$ and $t(B)$; is *μ-out-through* if $(\mathcal{G} \ominus B) \cap \mathrm{Lf}_{\mathcal{F}}(f)$ contains a path between $s(B)$ and $t(B)$; is *μ-critical* if it is both μ-through and μ-out-through; is *μ-inclusive* if $V(B) \subseteq \mathrm{Lf}_{\mathcal{F}}(f)$ and every vertex of $\Gamma \cap (V(B) - \{s(B), t(B)\})$ satisfies $\psi(x) \in \mathrm{Ds}_{\mathcal{F}}(f)$. Note that if a vertex f of $\mathcal{F}$ is μ-critical (respectively, μ-through, μ-out-through, or μ-inclusive), then so is every ancestor of f in $\mathcal{F}$. A vertex f of $\mathcal{F}$ is *extreme* μ-critical if f is μ-critical but no child of f in $\mathcal{F}$ is. Define extreme μ-through vertices, extreme μ-out-through vertices, and extreme μ-inclusive vertices similarly. For example, if T is the decomposition tree in Figure 4.1, $\mathcal{F}$ is the rooted forest in Figure 3.2, $\Gamma = \{4, 5, 10\}$, $\psi(4) = f_4$, $\psi(5) = f_6$ and $\psi(10) = \perp$, then f_2 is extreme μ-through, f_6 is extreme μ-out-through, extreme μ-critical and extreme μ-inclusive, f_5 is extreme ν-out-through, and there is no ν-through vertex.

Fact 4.5 *For each node μ of T, there is at most one extreme μ-critical (respectively, μ-through, μ-out-through, or μ-inclusive) vertex in $\mathcal{F}$.*

Proof: We only prove the fact for the case of extreme μ-critical vertex; the other three cases are similar. Towards a contradiction, assume that f_1 and f_2 are two distinct extreme μ-critical vertices in $\mathcal{F}$. Then, none of f_1 and f_2 is an ancestor of the other. So, $\mathrm{Lf}_{\mathcal{F}}(f_1) \cap \mathrm{Lf}_{\mathcal{F}}(f_2) = \emptyset$ because $(G, \mathcal{F})$ is a PGF pair. On the other hand, since f_1 and f_2 are μ-critical, $\{s(\mathrm{Prt}(\mu)), t(\mathrm{Prt}(\mu))\} \subseteq \mathrm{Lf}_{\mathcal{F}}(f_1)$ and $\{s(\mathrm{Prt}(\mu)), t(\mathrm{Prt}(\mu))\} \subseteq \mathrm{Lf}_{\mathcal{F}}(f_2)$, a contradiction. ▮

For each node μ in T, if there is a μ-through (respectively, μ-out-through, μ-critical, or μ-inclusive) vertex in $\mathcal{F}$, then let x-thr(μ) (respectively, x-out-thr(μ), x-crt(μ), or x-inc(μ)) denote the extreme μ-through (respectively, μ-out-through, μ-critical, or μ-inclusive) vertex in $\mathcal{F}$; otherwise, let x-thr$(\mu) = \perp$ (respectively, x-out-thr$(\mu) = \perp$, x-crt$(\mu) = \perp$, or x-inc$(\mu) = \perp$).

For convenience, we define an *embedding* of a TTG $\mathcal{H}$ to be a TTG that has the same vertices, edges and poles as $\mathcal{H}$. Let μ be a node of T. Let f be a vertex of $\mathcal{F}$ with $V(\mathrm{Prt}(\mu)) \cap \mathrm{Lf}_{\mathcal{F}}(f) \neq \emptyset$. An embedding B of $\mathrm{Prt}(\mu)$ is *potential* for f if

- $B \cap \mathrm{Lf}_{\mathcal{F}}(f)$ is a superface of B and every vertex $y \in \Gamma \cap \mathrm{Lf}_{\mathcal{F}}(f) \cap V(B)$ with $\psi(y) = \perp$ or $\psi(y) \notin \mathrm{Ds}_{\mathcal{F}}(f)$ is an outer vertex of $B \cap \mathrm{Lf}_{\mathcal{F}}(f)$.

Note that since $(G, \mathcal{F})$ is a connected PGF pair, $B \cap \mathrm{Lf}_{\mathcal{F}}(f)$ can be disconnected only if f is not a μ-through vertex but is a μ-out-through vertex. Moreover, if $B \cap \mathrm{Lf}_{\mathcal{F}}(f)$ is disconnected, then it has exactly two connected components one of which contains $s(B)$ and the other contains $t(B)$.

An embedding B of $\mathrm{Prt}(\mu)$ is *potential* if it is potential for every vertex f in $\mathcal{F}$ with $V(B) \cap \mathrm{Lf}_{\mathcal{F}}(f) \neq \emptyset$.

Let B be a potential embedding of $\mathrm{Prt}(\mu)$. Let f be a μ-critical vertex in $\mathcal{F}$. The left (respectively, right) side of B is *surroundable* for f if

- every vertex of the left (respectively, right) side of B is contained in $\mathrm{Lf}_{\mathcal{F}}(f)$, and
- every vertex $y \in \Gamma \cap \mathrm{Lf}_{\mathcal{F}}(f) \cap V(B)$ with $\psi(y) = \perp$ or $\psi(y) \notin \mathrm{Ds}_{\mathcal{F}}(f)$ is an outer vertex of the plane graph obtained from $B \cap \mathrm{Lf}_{\mathcal{F}}(f)$ by adding a new edge $\{s(B), t(B)\}$ on the left (respetively, right) of $B \cap \mathrm{Lf}_{\mathcal{F}}(f)$.

Note that when μ is a P- or R-node of T, both sides of B are surroundable for f if and only if f is μ-inclusive. So, if μ is a P- or R-node of T and both sides of B are surroundable for f, then both sides of B are surroundable for all ancestors of f in $\mathcal{F}$. On the other hand, when μ is an S-node of T, f is μ-inclusive if and only if both sides of B are surroundable for f and every $y \in \Gamma \cap V(\mathrm{Skl}(\mu)) - \{s(B), t(B)\}$ satisfies $\psi(y) \in \mathrm{Ds}_{\mathcal{F}}(f)$. So, when μ is an S-node of T, f may not be μ-inclusive even if both sides of B are surroundable for f.

A *surroundable embedding* of $\mathrm{Prt}(\mu)$ is a potential embedding B of $\mathrm{Prt}(\mu)$ such that at least one side of B is surroundable for all μ-critical vertices of $\mathcal{F}$.

Lemma 4.6 *Suppose that an embedding $\mathcal{G}'$ of $\mathcal{G}$ satisfies $(\mathcal{F}, \Gamma, \psi)$. Then, for every node μ of T, the plane subgraph of $\mathcal{G}'$ that is an embedding of* $\mathrm{Prt}(\mu)$ *is a surroundable embedding of* $\mathrm{Prt}(\mu)$.

Proof: Obvious. ∎

Lemma 4.7 *Suppose that an embedding $\mathcal{G}_1$ of $\mathcal{G}$ satisfies $(\mathcal{F}, \Gamma, \psi)$. Let μ be a node of T. Let B_μ be the plane subgraph of $\mathcal{G}_1$ that is an embedding of* $\mathrm{Prt}(\mu)$. *Let B'_μ be a surroundable embedding of* $\mathrm{Prt}(\mu)$. *Then, the following hold:*

1. *Suppose that $\mathcal{F}$ has no μ-critical vertex. Then, the embedding $\mathcal{G}_2$ of $\mathcal{G}$ obtained from $\mathcal{G}_1$ by replacing B_μ with B'_μ satisfies $(\mathcal{F}, \Gamma, \psi)$.*
2. *Suppose that $\mathcal{F}$ has μ-critical vertices. Let P be a side (the left or right side) of B_μ that is surroundable for all μ-critical vertices of $\mathcal{F}$. Let B''_μ be obtained by flipping B'_μ around its poles. Let $\mathcal{G}_3$ be the embedding of $\mathcal{G}$ obtained from $\mathcal{G}_1$ by replacing B_μ with B'_μ if either (i) P is the right side of B_μ and the right side of B'_μ is surroundable for all μ-critical vertices of $\mathcal{F}$ or (ii) P is the left side of B_μ and the left side of B'_μ is surroundable for all μ-critical vertices of $\mathcal{F}$, while replacing B_μ with B''_μ otherwise. Then, $\mathcal{G}_3$ satisfies $(\mathcal{F}, \Gamma, \psi)$.*

Proof: We prove the two statements separately as follows.

Statement 1. Towards a contradiction, assume that $\mathcal{G}_2$ does not satisfy $(\mathcal{F}, \Gamma, \psi)$. Then, we can find a vertex f_2 in $\mathcal{F}$ and a cycle C in $\mathcal{G}_2$ such that $V(C) \subseteq \mathrm{Lf}_{\mathcal{F}}(f_2)$ but some vertex $x \in (V(\mathcal{G}_2) - \mathrm{Lf}_{\mathcal{F}}(f_2)) \cup (\Gamma - \psi^{-1}(\mathrm{Ds}_{\mathcal{F}}(f_2)))$ appears in the interior of C in embedding $\mathcal{G}_2$.

Since C is a cycle and f_2 is not μ-critical, C is a cycle in either $B'_\mu \cap \mathrm{Lf}_{\mathcal{F}}(f_2)$ or $\mathcal{G}_2 \ominus B'_\mu$. Let U_1 (respectively, U_2) be the set of all vertices appearing in the interior of C in embedding $\mathcal{G}_1$ (respectively, $\mathcal{G}_2$). Then, $x \in U_2$.

Case 1: C is a cycle in $B'_\mu \cap \mathrm{Lf}_{\mathcal{F}}(f_2)$. Since B'_μ is a superface of $\mathcal{G}_2$ by Fact 4.1, no vertex of $V(\mathcal{G}_2) - V(B'_\mu)$ appears in the interior of C in embedding $\mathcal{G}_2$. So, U_2 is also the set of all vertices appearing in the interior of C in embedding B'_μ. In turn, $x \in (V(B'_\mu) - \mathrm{Lf}_{\mathcal{F}}(f_2)) \cup (\Gamma - \psi^{-1}(\mathrm{Ds}_{\mathcal{F}}(f_2)))$ and x appears in the interior of C in embedding B'_μ. However, this contradicts the assumption that B'_μ is a surroundable embedding of $\mathrm{Prt}(\mu)$.

Case 2: C is a cycle in $\mathcal{G}_2 \ominus B'_\mu$ and $V(B'_\mu) \cap U_2 = \emptyset$. Then, U_2 is also the set of all vertices appearing in the interior of C in embedding $\mathcal{G}_2 \ominus B'_\mu$. In turn, since $\mathcal{G}_2 \ominus B'_\mu$ and $\mathcal{G}_1 \ominus B_\mu$ are the same plane graph, U_2 is also the set of all vertices appearing in the interior of C in embedding $\mathcal{G}_1 \ominus B_\mu$. Thus, $x \in (V(\mathcal{G}_1) - \mathrm{Lf}_{\mathcal{F}}(f_2)) \cup (\Gamma - \psi^{-1}(\mathrm{Ds}_{\mathcal{F}}(f_2)))$ and $x \in U_1$. However, this contradicts the assumption that $\mathcal{G}_1$ is an embedding of $\mathcal{G}$ satisfying $(\mathcal{F}, \Gamma, \psi)$.

Case 3: C is a cycle in $\mathcal{G}_2 \ominus B'_\mu$ and $V(B'_\mu) \cap U_2 \neq \emptyset$. Then, since B'_μ is a block of $\mathcal{G}_2$, $V(B'_\mu) \subseteq U_2$. In turn, $V(B_\mu) \subseteq U_1$. Since $V(C) \subseteq \mathrm{Lf}_{\mathcal{F}}(f_2)$ and $\mathcal{G}_1[\mathrm{Lf}_{\mathcal{F}}(f_2)]$ is a superface of $\mathcal{G}_1$, we have that $V(B_\mu) \subseteq \mathrm{Lf}_{\mathcal{F}}(f_2)$ and f_2 is both μ-through and μ-out-through. However, this contradicts the assumption that $\mathcal{F}$ has no μ-critical vertex.

Since one of Cases 1, 2 and 3 must occur and we have obtained a contradiction in each case, $\mathcal{G}_2$ has to satisfy $(\mathcal{F}, \Gamma, \psi)$. This establishes Statement 1.

Statement 2. We assume that P is the right side of B_μ and the left side of B'_μ is surroundable for all μ-critical vertices of $\mathcal{F}$; the other cases are similar. Then, $\mathcal{G}_3$ is the embedding of $\mathcal{G}$ obtained from $\mathcal{G}_1$ by replacing B_μ with B''_μ.

Towards a contradiction, assume that $\mathcal{G}_3$ does not satisfy $(\mathcal{F}, \Gamma, \psi)$. Then, we can find a vertex f_3 in $\mathcal{F}$ and a cycle C in $\mathcal{G}_3$ such that $V(C) \subseteq \mathrm{Lf}_{\mathcal{F}}(f_3)$ but some vertex $y \in (V(\mathcal{G}_3) - \mathrm{Lf}_{\mathcal{F}}(f_3)) \cup (\Gamma - \psi^{-1}(\mathrm{Ds}_{\mathcal{F}}(f_3)))$ appears in the interior of C in embedding $\mathcal{G}_3$. Let U_1 (respectively, U_3) be the set of all vertices appearing in the interior of C in embedding $\mathcal{G}_1$ (respectively, $\mathcal{G}_3$). If either C is a cycle in $B''_\mu \cap \mathrm{Lf}_{\mathcal{F}}(f_3)$, or C is a cycle in $\mathcal{G}_3 \ominus B''_\mu$ but $V(B''_\mu) \cap U_3 = \emptyset$, then we can obtain a contradiction as in Cases 1 and 2 in the proof of Statement 1. So, we only need to consider the following cases:

Case 4: C is a cycle in $\mathcal{G}_3 \ominus B''_\mu$ and $V(B''_\mu) \cap U_3 \neq \emptyset$. Then, as in Case 3 in the proof of Statement 1, we can claim that $V(B_\mu) \subseteq U_1$. This together with the fact that $\mathcal{G}_1 \ominus B_\mu$ and $\mathcal{G}_3 \ominus B''_\mu$ are the same plane graph implies that $U_3 = U_1$. Moreover, since $V(C) \subseteq \mathrm{Lf}_{\mathcal{F}}(f_3)$ and $\mathcal{G}_1[\mathrm{Lf}_{\mathcal{F}}(f_3)]$ is a superface of $\mathcal{G}_1$, $U_1 \subseteq \mathrm{Lf}_{\mathcal{F}}(f_3)$ and $U_1 \cap (\Gamma - \psi^{-1}(\mathrm{Ds}_{\mathcal{F}}(f_3))) = \emptyset$. Thus, $U_3 \subseteq \mathrm{Lf}_{\mathcal{F}}(f_3)$ and $U_3 \cap (\Gamma - \psi^{-1}(\mathrm{Ds}_{\mathcal{F}}(f_3)) = \emptyset$. However, this contradicts the existence of $y \in U_3$.

Case 5: C is neither a cycle in B''_μ nor a cycle in $\mathcal{G}_3 \ominus B''_\mu$. Then, the intersection of C with B''_μ is a path P_1 between $s(B''_\mu)$ and $t(B''_\mu)$, and the intersection of C with $\mathcal{G}_3 \ominus B''_\mu$ is a path P_2 between $s(B''_\mu)$ and $t(B''_\mu)$. So, f_3 is μ-critical. Let P_3 be the right side of B''_μ. P_3 is surroundable for f_3 because P_3 is the left side of B'_μ by assumption. Let C_1 (respectively, C_3) be the cycle of $\mathcal{G}$ formed by P (respectively, P_3) and P_2 together. Let U_4 (respectively, U_5) be the set of all vertices appearing in the interior of C_1 (respectively, C_3) in embedding $\mathcal{G}_1$ (respectively, $\mathcal{G}_3$).

Case 5.1: $V(B_\mu) \subseteq V(C_1) \cup U_4$. Then, P_2 is on the left of B_μ in embedding $\mathcal{G}_1$. In turn, P_2 is on the left of B''_μ in embedding $\mathcal{G}_3$. Consequently, $V(B''_\mu) \subseteq V(C_3) \cup U_5$ and $U_3 \subseteq U_5$.

By the fact that $\mathcal{G}_1 \ominus B_\mu$ and $\mathcal{G}_3 \ominus B''_\mu$ are the same plane graph, we now have $U_4 = U_5$. Moreover, since $V(C_1) \subseteq \mathrm{Lf}_\mathcal{F}(f_3)$ and $\mathcal{G}_1[\mathrm{Lf}_\mathcal{F}(f_3)]$ is a superface of $\mathcal{G}_1$, $U_4 \subseteq \mathrm{Lf}_\mathcal{F}(f_3)$ and $U_4 \cap (\Gamma - \psi^{-1}(\mathrm{Ds}_\mathcal{F}(f_3))) = \emptyset$. So, $U_3 \subseteq U_5 \subseteq \mathrm{Lf}_\mathcal{F}(f_3)$ and $U_3 \cap (\Gamma - \psi^{-1}(\mathrm{Ds}_\mathcal{F}(f_3))) \subseteq U_5 \cap (\Gamma - \psi^{-1}(\mathrm{Ds}_\mathcal{F}(f_3))) = \emptyset$. However, this contradicts the existence of $y \in U_3$.

Case 5.2: $V(B_\mu) \not\subseteq V(C_1) \cup U_4$. Then, P_2 is on the right of B_μ in embedding $\mathcal{G}_1$. In turn, P_2 is on the right of B''_μ in embedding $\mathcal{G}_3$. Moreover, $U_4 = U_5$ and $U_5 \subseteq U_3$. Consider the plane graph H obtained from B''_μ by adding a new edge $e = \{s(B''_\mu), t(B''_\mu)\}$ on the right of B''_μ. Let C_4 be the cycle of H formed by e and P_1 together. Let U_6 be the set of all vertices appearing in the interior of C_4 in embedding H. Then, $U_3 = U_5 \cup U_6$. Since P is surroundable for f_3, $V(P) \subseteq \mathrm{Lf}_\mathcal{F}(f_3)$. In turn, $V(C_1) \subseteq \mathrm{Lf}_\mathcal{F}(f_3)$. This together with the fact that $\mathcal{G}_1[\mathrm{Lf}_\mathcal{F}(f_3)]$ is a superface of $\mathcal{G}_1$ implies that $U_4 \subseteq \mathrm{Lf}_\mathcal{F}(f_3)$ and $U_4 \cap (\Gamma - \psi^{-1}(\mathrm{Ds}_\mathcal{F}(f_3))) = \emptyset$. Thus, $y \notin U_4 = U_5$ and hence $y \in U_3 - U_5 = U_6$.

Let $s(B''_\mu), u_1, \ldots, u_k, t(B''_\mu)$ be the common vertices of P_1 and P_3, and assume that they appear on P_3 in this order. It is possible that $k = 0$. For convenience, let $u_0 = s(B''_\mu)$ and $u_{k+1} = t(B''_\mu)$. For each $i \in \{0, \ldots, k\}$, let $P_{1,i}$ (respectively, $P_{3,i}$) be the subpath of P_1 (respectively, P_3) between u_i and u_{i+1}. Let I be the set of all $i \in \{0, \ldots, k\}$ such that $P_{1,i}$ and $P_{3,i}$ together form a cycle in B''_μ. For each $i \in I$, let W_i be the set of all vertices appearing in the interior of the cycle formed by $P_{1,i}$ and $P_{3,i}$ together in embedding B''_μ. Then, $U_6 = \cup_{i \in I} W_i \cup (V(P_3) - \{u_0, \ldots, u_{k+1}\})$. So, either $y \in W_i$ for some $i \in I$, or $y \in V(P_3) - \{u_0, \ldots, u_{k+1}\}$. On the other hand, since $P_{1,i}$ and $P_{3,i}$ together form a cycle, $V(P_{1,i}) \cup V(P_{3,i}) \subseteq \mathrm{Lf}_\mathcal{F}(f_3)$ and $B''_\mu \cap \mathrm{Lf}_\mathcal{F}(f_3)$ is a superface of B''_μ, it holds that $W_i \subseteq \mathrm{Lf}_\mathcal{F}(f_3)$ and $W_i \cap (\Gamma - \psi^{-1}(\mathrm{Ds}_\mathcal{F}(f_3))) = \emptyset$. Thus, $y \in V(P_3) - \{u_0, \ldots, u_{k+1}\}$. Note that no vertex in $V(P_3) - \{u_0, \ldots, u_{k+1}\}$ appears on the boundary of the outer face of $H \cap \mathrm{Lf}_\mathcal{F}(f_3)$. Therefore, the existence of y contradicts the fact that P_3 is surroundable for f_3.

Since we have obtained a contradiction in each possible case, Statement 2 has to hold. ∎

Recall that our algorithm processes the nodes of T in post-order. Right before the processing of a non-leaf node μ, the following invariant holds for all children ν of μ in T:

- A surroundable embedding B_ν of $\mathrm{Prt}(\nu)$ has been fixed, except for a possible flipping around its poles. Moreover, one side of B_ν that is surroundable for all ν-critical vertices of $\mathcal{F}$ has been labeled "surroundable".

By Lemma 4.7, unless no embedding of $\mathcal{G}$ satisfies $(\mathcal{F}, \Gamma, \psi)$, $\mathrm{Prt}(\mu)$ has a surroundable embedding B_μ such that for each child ν of μ in T, either B_ν or its flipping is a superface of B_μ.

Since each leaf node μ of T is a Q-node, $\mathrm{Prt}(\mu)$ is an edge and hence is a surroundable embedding of itself; the processing of μ is completed by labeling the right side of $\mathrm{Prt}(\mu)$ "surroundable". In the next three subsections, let μ be a non-leaf node of T whose children have been processed. Let $\nu_1, \ldots, \nu_k$ be the children of μ in T, and $B_1, \ldots, B_k$ be the fixed surroundable embedding of $\mathrm{Prt}(\nu_1), \ldots, \mathrm{Prt}(\nu_k)$, respectively. Moreover, for each $i \in \{1, \ldots, k\}$, let e_i be the virtual edge of ν_i in $\mathrm{Skl}(\mu)$. Let B be the plane graph obtained from $\mathrm{Skl}(\mu)$ by replacing e_i with B_i for all $i \in \{1, \ldots, k\}$. Note that B is an embedding of $\mathrm{Prt}(\mu)$. For brevity, let K denote $\mathrm{Skl}(\mu)$.

For a vertex f of $\mathcal{F}$, let $A_\mu(f) = \{e_i \in \{e_1, \ldots, e_k\} \mid f \text{ is } \nu_i\text{-through}\}$. A vertex f of $\mathcal{F}$ is *μ-cyclic*, if $K[A_\mu(f)]$ contains a cycle. Note that if f is μ-cyclic, then so is every ancestor of f in $\mathcal{F}$. Obviously, if $\mathcal{F}$ has neither μ-cyclic nor μ-critical vertex, then B must be a surroundable embedding of $\mathrm{Prt}(\mu)$.

The next two lemmas are necessary to speed up the processing of μ when μ is a P- or R-node.

Lemma 4.8 *Suppose that μ is a P- or R-node. Let f_1 be a vertex in $\mathcal{F}$. Suppose that the parent f_2 of f_1 in $\mathcal{F}$ satisfies $A_\mu(f_1) = A_\mu(f_2)$. Assume that B' is an embedding of $\mathrm{Prt}(\mu)$ potential for f_1 such that for each $i \in \{1, \ldots, k\}$, the plane subgraph B'_i of B' that is an embedding of $\mathrm{Prt}(\mu)$ is either the same as B_i or the same as the flipping of B_i. Then, B' is also potential for f_2.*

Proof: Towards a contradiction, assume that B' is not potential for f_2. Then, there is a cycle C in B' with $V(C) \subseteq \mathrm{Lf}_{\mathcal{F}}(f_2)$ whose interior in embedding B' contains a vertex y such that $y \notin \mathrm{Lf}_{\mathcal{F}}(f_2)$ or $y \in \Gamma - \psi^{-1}(\mathrm{Ds}_{\mathcal{F}}(f_2))$. First suppose that there is an $i \in \{1, \ldots, k\}$ such that B'_i contains all the edges of C. Then, since B'_i is either the same as B_i or the same as the flipping of B_i, C is a cycle in B_i with $V(C) \subseteq \mathrm{Lf}_{\mathcal{F}}(f_2)$ whose interior in embedding B_i contains a vertex y such that $y \notin \mathrm{Lf}_{\mathcal{F}}(f_2)$ or $y \in \Gamma - \psi^{-1}(\mathrm{Ds}_{\mathcal{F}}(f_2))$. However, this contradicts that B_i is a potential embedding of $\mathrm{Prt}(\nu_i)$. Next suppose that there is no $i \in \{1, \ldots, k\}$ such that B'_i contains all the edges of F. Then, for every $i \in \{1, \ldots, k\}$ such that B'_i contains at least one edge of C, the intersection of C and B'_i is a path between $s(B'_i)$ and $t(B'_i)$; so $e_i \in A_\mu(f_2)$ and $e_i \in A_\mu(f_1)$ in turn. Consequently, there is a cycle C_1 in $B' \cap \mathrm{Lf}_{\mathcal{F}}(f_1)$ that goes through those $B'_i \in \{B'_1, \ldots, B'_k\}$ that C goes through. One of the following two cases must occur:

Case 1: Either y is a vertex of K, or y is a nonpole of a unique $B'_i \in \{B'_1, \ldots, B'_k\}$ such that C does not go through B'_i. Then, for every $j \in \{1, \ldots, k\}$ such that C goes through B'_j, since $\{s(B'_j), t(B'_j)\} \subseteq V(C)$, it holds that $y \notin V(B'_j)$. In turn, y appears in the interior of C_1 in embedding B' because C_1 goes through those $B'_j \in \{B'_1, \ldots, B'_k\}$ that C goes through. Note that $y \in \Gamma - \psi^{-1}(\mathrm{Ds}_{\mathcal{F}}(f_2)) \subseteq \Gamma - \psi^{-1}(\mathrm{Ds}_{\mathcal{F}}(f_1))$. Thus, y is an inner vertex of $B' \cap \mathrm{Lf}_{\mathcal{F}}(f_1)$ and $y \in \Gamma - \psi^{-1}(\mathrm{Ds}_{\mathcal{F}}(f_1))$. This contradicts that B' is an embedding of $\mathrm{Prt}(\mu)$ potential for f_1.

Case 2: y is a nonpole of a unique $B'_i \in \{B'_1, \ldots, B'_k\}$ such that C goes through B'_i. Due to the existence of cycles C and C_1, both f_1 and f_2 are ν_i-critical. Obviously, at least one side P of B'_i is contained in the closed interiors of C and C_1 in embedding B'. We may assume that P is the right side of B'_i; the other case is similar. Then, y is not a vertex of the left side of B'_i. Since B' is potential for f_1, $V(P) \subseteq \mathrm{Lf}_{\mathcal{F}}(f_1)$. On the other hand, at least one vertex of the left side of B'_i is not in $\mathrm{Lf}_{\mathcal{F}}(f_1)$ or else y would be an inner vertex of $B'_i \cap \mathrm{Lf}_{\mathcal{F}}(f_1)$ and hence of $B_i \cap \mathrm{Lf}_{\mathcal{F}}(f_1)$, contradicting that B_i is a surroundable embedding of $\mathrm{Prt}(\nu_i)$. Thus, the left side of B'_i is not a surroundable side of B_i for f_1. In turn, P has to be the surroundable side of B_i for all ν_i-critical vertices of $\mathcal{F}$. However, since y appears on the right of the subpath of C in B'_i, y is an inner vertex of the plane graph obtained from $B'_i \cap \mathrm{Lf}_{\mathcal{F}}(f_2)$ by adding a new edge $\{s(B'_i), t(B'_i)\}$ on the right of $B'_i \cap \mathrm{Lf}_{\mathcal{F}}(f_2)$; so, P is not a surroundable side of B_i for f_2, a contradiction. ∎

Lemma 4.9 *Suppose that μ is a P- or R-node. Let f_1 be a μ-critical vertex in $\mathcal{F}$. Suppose that the parent f_2 of f_1 in $\mathcal{F}$ satisfies $A_\mu(f_1) = A_\mu(f_2)$. Assume that B' is a potential embedding of $\mathrm{Prt}(\mu)$ such that for each $i \in \{1, \ldots, k\}$, the plane subgraph B'_i of B' that is* *an embedding of* $\mathrm{Prt}(\mu)$ *is either the same as* B_i *or the same as the flipping of* B_i*. Then, if the right (respectively, left) side of* B' *is surroundable for* f_1*, the right (respectively, left) side of* B' *is also surroundable for* f_2*.*

Proof: Towards a contradiction, assume that the right side of B' is surroundable for f_1 but is not surroundable for f_2. For $j \in \{1, 2\}$, let H_j be the plane graph obtained from $B' \cap \mathrm{Lf}_{\mathcal{F}}(f_j)$ by adding a new edge $\{s(B'), t(B')\}$ on the right of $B' \cap \mathrm{Lf}_{\mathcal{F}}(f_j)$. Then, since every vertex of the right side of B' is contained in $\mathrm{Lf}_{\mathcal{F}}(f_1)$ and $\mathrm{Lf}_{\mathcal{F}}(f_1) \subseteq \mathrm{Lf}_{\mathcal{F}}(f_2)$, every vertex of the right side of B' is contained in $\mathrm{Lf}_{\mathcal{F}}(f_2)$; so, at least one inner vertex y of H_2 is contained in $\Gamma - \psi^{-1}(\mathrm{Ds}_{\mathcal{F}}(f_2))$. Obviously, y is a nonpole of B'. Moreover, the right side of B' must contain y or else y would be an inner vertex of $B' \cap \mathrm{Lf}_{\mathcal{F}}(f_2)$, contradicting that B' is potential for f_2. We distinguish two cases as follows.

Case 1: y is a vertex of K. Then, μ must be an R-node. So, K is also the plane graph obtained from B' by replacing every $B'_i \in \{B'_1, \ldots, B'_k\}$ with an edge. Moreover, y must be an inner vertex of the plane graph obtained from $K[A_\mu(f_2)]$ by adding a new edge $\{s(K), t(K)\}$ on the right of $K[A_\mu(f_2)]$; otherwise y would not be an inner vertex of H_2. Since $A_\mu(f_2) = A_\mu(f_1)$, y must be an inner vertex of the plane graph obtained from $K[A_\mu(f_1)]$ by adding a new edge $\{s(K), t(K)\}$ on the right of $K[A_\mu(f_1)]$. In turn, y is an inner vertex of H_1. This contradicts that the right side of B' is surroundable for f_1.

Case 2: y is not a vertex of K. Then, there is a unique integer $i \in \{1, \ldots, k\}$ such that y is a nonpole of B'_i. Edge e_i cannot be an inner edge or an outer non-bridge of the plane subgraph of K induced by $A_\mu(f_2)$ or else y would be an inner vertex of H_1. On the other hand, y is an inner vertex of H_2. So, y must be an inner vertex of the plane graph obtained from $B'_i \cap \mathrm{Lf}_{\mathcal{F}}(f_2)$ by adding a new edge $\{s(B'_i), t(B'_i)\}$ on the right of $B'_i \cap \mathrm{Lf}_{\mathcal{F}}(f_2)$. Thus, the right side of B'_i is not surroundable for f_2. Moreover, at least one vertex of the left side of B'_i is not contained in $\mathrm{Lf}_{\mathcal{F}}(f_1)$ or else y would be an inner vertex of the plane graph obtained from $B'_i \cap \mathrm{Lf}_{\mathcal{F}}(f_1)$ by adding a new edge $\{s(B'_i), t(B'_i)\}$ on the right of $B'_i \cap \mathrm{Lf}_{\mathcal{F}}(f_1)$. This implies that the left side of B'_i is not surroundable for f_1 and the right side of B'_i has to be surroundable for all ν_i-critical vertices of $\mathcal{F}$, contradicting that the right side of B'_i is not surroundable for f_2 which is ν_i-critical. ▮

4.3 Processing of S-nodes

Suppose that μ is an S-node of T. Then, K is a path in which the edges $e_1, \ldots, e_k$ appear in this order. So, $\mathcal{F}$ has no μ-cyclic vertex. Consequently, B is potential if and only if so are B_1 through B_k. Moreover, we can modify B by flipping each $B_i \in \{B_1, \ldots, B_k\}$ whose left side has been labeled "surroundable". Then, the right side of B becomes surroundable for all μ-critical vertices of $\mathcal{F}$; we label the right side of B "surroundable".

4.4 Processing of P-nodes

Suppose that μ is a P-node of T. Then, K is a bunch of parallel edges. Let $I = \{i \in \{1, \ldots, k\} \mid \text{x-thr}(\nu_i) \neq \perp\}$. If $I = \emptyset$, then obviously B is a surroundable embedding of $\mathrm{Prt}(\mu)$ and the processing of μ is completed by labeling the right side of B "surroundable". Suppose that I consists of a single integer i. Then, B is a potential embedding of $\mathrm{Prt}(\mu)$. If $\text{x-crt}(\mu) = \perp$, we are done by labeling the right side of B "surroundable". So, assume that x-crt$(\mu) \neq \perp$. Then, x-crt(ν_i) = x-crt(μ). We modify B by (1) permuting $B_1, \ldots, B_k$ so that B_i is on the rightmost and (2) flipping B_i if its left side is labeled "surroundable". Then, the right side of B becomes surroundable for x-crt(μ). In turn, by Lemma 4.9 and the fact that $|I| = 1$, the right side of B is surroundable for all μ-critical vertices in $\mathcal{F}$. So, B is a surroundable embedding of Prt(μ) and the processing of μ is completed by labeling the right side of B "surroundable". Thus, we assume that $|I| \geq 2$. Then, for all $i \in I$, x-crt$(\nu_i) \neq \perp$.

Since $\{s(B), t(B)\} \subseteq \mathrm{Lf}_{\mathcal{F}}(f)$ for all μ-through vertices f in $\mathcal{F}$, every μ-cyclic or μ-critical vertex must appear on the path P from x-thr(μ) to the root ancestor f_r of x-thr(μ) in $\mathcal{F}$. If x-crt$(\mu) = \perp$, then it suffices to decide whether Prt(μ) has a potential embedding, and so we process μ by calling P-Procedure1 below. Otherwise, we process μ by calling P-Procedure2 below.

P-Procedure1: (See Figure 4.2 for an example.)

1. Compute the set $U = \{\text{x-thr}(\nu_i) \mid i \in I\}$. Let $\ell = |U|$.
2. Sort the vertices in U in non-increasing order of their pre-order numbers in $\mathcal{F}$; let $S = (f_1, f_2, \ldots, f_\ell)$ be the resulting list.

   (*Comment:* By Lemmas 4.3 and 4.8, we only need to check whether we can make B potential for all vertices in S by permuting and flipping some child blocks $B_i \in \{B_1, \ldots, B_k\}$ of B.)
3. For $j = 1, 2, \ldots, \ell$, compute the set $I_j = \{i \in I \mid \text{x-thr}(\nu_i) = f_j\}$.

   (*Comment:* Note that $|I_j| \geq 1$.)
4. Compute the set I'_1 of all $i \in I_1$ such that f_1 is not ν_i-inclusive.

   (*Comment:* f_1 is ν_i-inclusive if and only if x-inc$(\nu_i) \neq \perp$ and x-inc(ν_i) is a descendant of f_1 in $\mathcal{F}$.)
5. If $|I'_1| \geq 3$, then (B cannot be made potential for f_1 by permuting and flipping some child blocks of B, and so) stop immediately for failure.
6. Construct an auxiliary graph H_1 whose vertex set is I_1 and whose edges altogether form a path such that each element of I'_1 is an endpoint of the path.

   (*Comment:* H_1 itself is indeed a path.)
7. For $j = 2, 3, \ldots, \ell$ (in this order), perform the following substeps:

   (a) Let $i_{j,1}$ and $i_{j,2}$ be the endpoints of path H_{j-1}. Compute the set I'_j of all $i \in I_j \cup \{i_{j,1}, i_{j,2}\}$ such that f_j is not ν_i-inclusive.
   (*Comment:* $i_{j,1} = i_{j,2}$ if and only if $j = 2$ and $|I_1| = 1$. Note that for every $i \in I_{j-1} - \{i_{j,1}, i_{j,2}\}$, f_{j-1} is ν_i-inclusive; so f_j is ν_i-inclusive, too.)

   (b) If either (1) $|I'_j| \geq 3$, or (2) $i_{j,1} \neq i_{j,2}$ and $I'_j = \{i_{j,1}, i_{j,2}\}$, then ($B$ cannot be made potential for f_j by permuting and flipping some child blocks of B, and so) stop immediately for failure.
   (*Comment:* Since $|I_j| \geq 1$, when $i_{j,1} \neq i_{j,2}$ and $I'_j = \{i_{j,1}, i_{j,2}\}$, B cannot be made potential by permuting and flipping some child blocks of B.)

(c) Construct a supergraph H_j of H_{j-1} whose vertex set is $I_j \cup V(H_{j-1})$ and whose edges altogether form a path such that each element of I'_j is an endpoint of the path.

(*Comment:* H_j is a path and H_{j-1} is a subpath of H_j. Moreover, $V(H_j) = I_1 \cup \cdots \cup I_j$; so, $I = I_1 \cup \cdots \cup I_\ell = V(H_\ell)$.)

8. Modify B by (1) permuting its child blocks so that the blocks in $\{B_i \mid i \in I\}$ appear consecutively from left to right in the order specified by path H_ℓ, and (2) performing the following substeps for $j = 1, 2, \ldots, \ell$ (in this order):

   (a) Find the left endpoint $i_{j,1}$ and the right endpoint $i_{j,2}$ of path H_j.

   (b) If $i_{j,1} \neq i_{j,2}$ and the left side of $B_{i_{j,1}}$ is labeled "surroundable", then flip $B_{i_{j,1}}$ around its poles.

   (*Comment:* When $i_{j,1} \neq i_{j,2}$, $i_{j,1}$ cannot be the right endpoint of path $H_{j'}$ for all $j' > j$; so, $B_{i_{j,1}}$ cannot be flipped more than once.)

   (c) If $i_{j,1} \neq i_{j,2}$ and the right side of $B_{i_{j,2}}$ is labeled "surroundable", then flip $B_{i_{j,2}}$ around its poles.

   (*Comment:* When $i_{j,1} \neq i_{j,2}$, $i_{j,2}$ cannot be the left endpoint of path $H_{j'}$ for all $j' > j$; so, $B_{i_{j,2}}$ cannot be flipped more than once.)

9. Label an arbitrary side of B "surroundable".

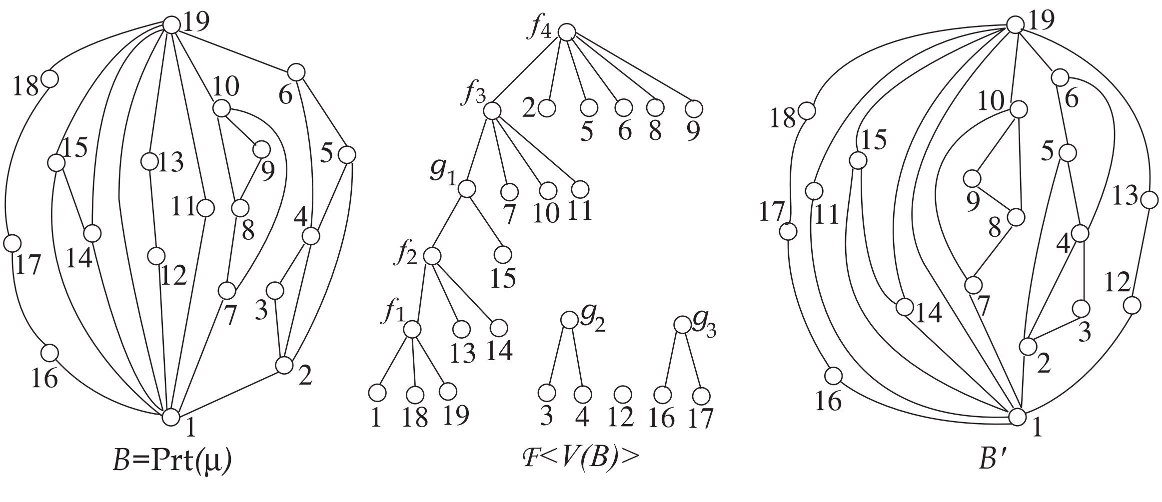

Figure 4.2: A block B of G, forest $\mathcal{F}\langle V(B)\rangle$, and an embedding B' of B computed by P-Procedure1. (*Comment:* B has seven child blocks $B_1, \ldots, B_7$ and they appear from left to right in this order. We assume that the right side of each child block is labeled "surroundable", $\Gamma \cap V(B) = \{11, 15\}$, $\psi(11) = \perp$ and $\psi(15) = f_3$. Then, $I = \{2, 3, 5, 6, 7\}$. The set U computed in Step 1 is $\{f_1, \ldots, f_4\}$. The sets I_1 through I_4 computed in Step 3 are $\{3\}$, $\{2\}$, $\{5, 6\}$, $\{7\}$, respectively. $I'_1 = \emptyset$ and H_1 is the path: 3. $I'_2 = \{2\}$ and H_2 is the path: 2, 3. $I'_3 = \{5, 6\}$ and H_3 is the path: 5, 2, 3, 6. $I'_4 = \{5, 7\}$ and H_4 is the path: 5, 2, 3, 6, 7. Note that B_6 and B_7 are flipped in Step 8.)

P-Procedure2: (See the next example.)

1. Compute the set $U = \{\text{x-thr}(\nu_i) \mid i \in I\} \cup \{\text{x-crt}(\mu)\}$. Let $\ell = |U|$.

2. Perform Steps 2 through 4 of P-Procedure1.

3. If either (1) $|I'_1| \geq 3$, or (2) $|I'_1| = 2$ and $f_1 = \text{x-crt}(\mu)$, then (B cannot be made either potential or surroundable for f_1 by permuting and flipping some child blocks of B, and so) stop immediately for failure.

4. Same as Step 6 of P-Procedure1.

5. Compute the integer $h \in \{1, \ldots, \ell\}$ such that $f_h = \text{x-crt}(\mu)$.

   (*Comment:* It is possible that $I_h = \emptyset$. However, for every $j \in \{1, \ldots, h-1, h+1, \ldots, \ell\}$, $I_j \neq \emptyset$.)

6. For $j = 2, 3, \ldots, h-1$ (in this order), perform Substeps 7a through 7c.

7. For $j = h$, perform the following substeps:

   (a) Same as Substep 7a of P-Procedure1.

   (b) If $|I'_j| \geq 2$, then (B cannot be made surroundable for f_h by permuting and flipping some child blocks of B, and so) stop immediately for failure.

   (c) Same as Substep 7c of P-Procedure1.

8. If $|I'_h| = 0$ or H_h has only one vertex, then let $i_{h,1}$ be an arbitrary endpoint of H_h. Otherwise, let $i_{h,1}$ be the endpoint of H_h that is not contained in I'_h.

9. Modify B by (1) permuting its child blocks so that $B_{i_{h,1}}$ is the rightmost block and the blocks in $\{B_i \mid i \in I_1 \cup \cdots \cup I_h\}$ appear consecutively from right to left in the order specified by path H_h, and (2) performing Substeps 8a through 8c of P-Procedure1 for $j = 1, 2, \ldots, h$ (in this order).

10. If H_h has only one vertex and the left side of $B_{i_{h,1}}$ is labeled "surroundable", then modify B by flipping $B_{i_{h,1}}$ around its poles.

    (*Comment:* If H_h has only one vertex, then $B_{i_{h,1}}$ is not flipped in Step 9. Otherwise, $i_{h,1} \notin I'_h$, f_h is $\nu_{i_{h,1}}$-inclusive, and both sides of $B_{i_{h,1}}$ are surroundable for all μ-critical vertices.)

11. For $j = h+1, h+2, \ldots, \ell$ (in this order), perform the following substeps:

    (a) If H_{j-1} has only one vertex, then ($i_{h,1}$ is the vertex of H_{j-1}) let $i_{j,2} = i_{h,1}$. Otherwise, let $i_{j,2}$ be the endpoint of path H_{j-1} that is different from $i_{h,1}$.

    (b) Compute the set I'_j of all $i \in I_j \cup \{i_{j,2}\}$ such that f_j is not ν_i-inclusive.

    (c) If either $|I'_j| \geq 2$ or $I'_j = \{i_{j,2}\}$, then (B cannot be made surroundable for f_j by permuting and flipping some child blocks of B, and so) stop immediately for failure.

    (*Comment:* Since $|I_j| \geq 1$, when $I'_j = \{i_{j,2}\}$, B cannot be made surroundable for f_j by permuting and flipping some child blocks of B.)

(d) Construct a supergraph H_j of H_{j-1} whose vertex set is $I_j \cup V(H_{j-1})$ and whose edges altogether form a path such that the elements of $\{i_{h,1}\} \cup I'_j$ are endpoints of the path.

12. Modify B by (1) permuting those child blocks in $\{B_i \mid i \in \{1,\ldots,k\} - (I_1 \cup \cdots \cup I_h)\}$ so that the blocks in $\{B_i \mid i \in I\}$ appear consecutively from right to left in the order specified by path H_ℓ, and (2) performing the following substeps for $j = h+1, h+2, \ldots, \ell$ (in this order):

    (a) Find the left endpoint $i_{j,2}$ of path H_j.
    (*Comment:* $i_{h,1}$ is the right endpoint of path H_j.)

    (b) If the left side of $B_{i_{j,2}}$ is labeled "surroundable", then flip $B_{i_{j,2}}$ around its poles.

13. Label the right side of B "surroundable".

Example 1 Consider the block B and forest $\mathcal{F}\langle V(B)\rangle$ in Figure 4.2. Suppose that x-crt$(\mu) = f_3$, $\Gamma = \{11, 15\}$, and $\psi(11) = \psi(15) = f_3$. Let B'' be the embedding of B obtained from B' by (1) moving the leftmost child block to the rightmost and (2) flipping the whole block. B'' is what P-Procedure2 computes.

It is easy to verify that after μ is processed without failure, the invariant in Section 4.2 hold for μ.

4.5 Processing R-nodes

Suppose that μ is an R-node of T. Since μ is an R-node, an embedding of K must be either K itself or its flipping around its poles. The processing of μ is based on the following lemma.

Lemma 4.10 *Suppose that* Prt(μ) *has a potential embedding. Let $f_1, \ldots, f_\ell$ be some vertices of $\mathcal{F}$ none of which is an ancestor of another. Then, $K[\cup_{1\le j\le \ell} A_\mu(f_j)]$ is a superface of K.*

Proof: Since Prt(μ) has a potential embedding, $K[A_\mu(f_j)]$ must be a superface of K. Moreover, since $\mathrm{Lf}_{\mathcal{F}}(f_i) \cap \mathrm{Lf}_{\mathcal{F}}(f_j) = \emptyset$ when $i \neq j$, each nontrivial 1-component C_i of $K[A_\mu(f_i)]$ and each nontrivial 1-component C_j of $K[A_\mu(f_j)]$ must be vertex-disjoint. Thus, each nontrivial 1-component C of $K[\cup_{1\le j\le \ell} A_\mu(f_j)]$ must be a nontrivial 1-component of some $K[A_\mu(f_j)]$; so C is a superface of K. Consequently, $K[\cup_{1\le j\le \ell} A_\mu(f_j)]$ is a superface of K. ∎

Corollary 4.11 *Suppose that* Prt(μ) *has a potential embedding. Let $f_1, \ldots, f_\ell$ be some vertices of $\mathcal{F}$. Then, $K[\cup_{1\le j\le \ell} A_\mu(f_j)]$ is a superface of K.*

Proof: Let U be the set of all $f_j \in \{f_1, \ldots, f_\ell\}$ such that f_j is not a descendant of f_i for every $i \in \{1, \ldots, j-1, j+1, \ldots, \ell\}$. Then, no vertex of U is an ancestor of another in $\mathcal{F}$. Moreover, $K[\cup_{1\le j\le \ell} A_\mu(f_j)]$ and $K[\cup_{f_j \in U} A_\mu(f_j)]$ are the same plane graph. So, the corollary follows from Lemma 4.10. ∎

If x-crt$(\mu) = \perp$, then it suffices to decide whether Prt(μ) has a potential embedding, and so we process μ by calling R-Procedure1 below. Otherwise, we process μ by calling R-Procedure2 below.

R-Procedure1: (See Figure 4.3 for an example.)

1. Compute the set $I = \{i \in \{1, \dots, k\} \mid \text{x-thr}(\nu_i) \neq \perp\}$.

2. Compute the set $U = \{\text{x-thr}(\nu_i) \mid i \in I\}$ and $\mathcal{F}\langle U \rangle$. Let W be the set of all vertices in $\mathcal{F}\langle U \rangle$.

3. Sort the vertices in W in non-increasing order of their pre-order numbers in $\mathcal{F}$; let $(f_1, \dots, f_\ell)$ be the resulting list.

   (*Comment:* If $1 \leq i < j \leq \ell$, then f_i is not an ancestor of f_j in $\mathcal{F}\langle U \rangle$. Moreover, by Lemmas 4.3 and 4.8, we only need to check whether we can make B potential for f_1 through f_ℓ by flipping some child blocks $B_i \in \{B_1, \dots, B_k\}$ of B.)

4. For $j = 1, 2, \dots, \ell$, compute the set $A_j = \{e_i \mid i \in I \text{ and } \text{x-thr}(\nu_i) = f_j\}$.

   (*Comment:* $A_1 \neq \emptyset$, because f_1 is a leaf vertex of $\mathcal{F}\langle U \rangle$ and hence belongs to U.)

5. Let $H_0 = K[\emptyset]$.

6. For $j = 1, 2, \dots, \ell$ (in this order), perform the following substeps:

   (a) Let $H_j = K[A_1 \cup \cdots \cup A_j]$.
   (*Comment:* We claim that for every nontrivial 1-component C of H_j, there is a vertex $f_h \in \{f_1, \dots, f_j\}$ such that $E(C) \subseteq A_\mu(f_h)$. The claim trivially holds when $j = 1$. Suppose that $j \geq 2$ and the claim holds for $1, 2, \dots, j-1$. Consider a nontrivial 1-component C of H_j. If C is also a 1-component of H_{j-1}, then the claim follows from the inductive hypothesis. So, assume that C is not a 1-component of H_{j-1}. Then, C has at least one edge of A_j. If we delete all edges of A_j from C, we get one or more 1-components of H_{j-1}. Each such 1-component C' of H_{j-1} contains at least one vertex y that is an endpoint of some edge of A_j. If C' is nontrivial, then by the inductive hypothesis, there is a vertex $f_{j'} \in \{f_1, \dots, f_{j-1}\}$ such that $E(C') \subseteq A_\mu(f_{j'})$. Since $y \in \text{Lf}_{\mathcal{F}}(f_j)$, $y \in \text{Lf}_{\mathcal{F}}(f_{j'})$ and $f_{j'}$ cannot be an ancestor of f_j in $\mathcal{F}$ for $j' < j$, f_j must be an ancestor of $f_{j'}$. Thus, $E(C') \subseteq A_\mu(f_j)$. This completes the proof of the claim.
   By the claim and the fact that for every edge $e_i \in A_j$, none of $f_1, \dots, f_{j-1}$ is ν_i-through, we have that for each nontrivial 1-component C of H_j with $E(C) \cap A_j \neq \emptyset$, C is a 1-component of $K[A_\mu(f_j)]$.)

   (b) If at least one of the following does not hold, then (B cannot be made potential for f_j by flipping some child blocks of B, and so) stop immediately for failure.

   - Every inner face of H_j is also an inner face of K.
     (*Comment:* If this does not hold, then Prt(μ) has no potential embedding by Corollary 4.11.)

- For every inner edge e_i of H_j that is not an inner edge of H_{j-1}, f_j is ν_i-inclusive.
  (*Comment:* Towards a contradiction, assume that this does not hold for some e_i but $\mathrm{Prt}(\mu)$ has a potential embedding. Let C be the nontrivial 1-component of H_j with $e_i \in E(C)$. Since every inner face of H_j is also an inner face of K, C must be a superface of K. Moreover, since e_i is an inner edge of H_j but is not an inner edge of H_{j-1}, C has at least one edge of A_j. Thus, by the comment on Substep 6a, C is a nontrivial 1-component and also a superface of $K[A_\mu(f_j)]$. In turn, e_i must be an inner edge of $K[A_\mu(f_j)]$. Now, since either x-inc$(\nu_i) = \perp$ or x-inc(ν_i) is not a descendant of f_j in $\mathcal{F}$, it is easy to see that we cannot make B potential for f_j by flipping some child blocks of B, a contradiction.
  Also note that for every nontrivial 1-component C of H_j with $E(C) \cap A_j \neq \emptyset$ and for every inner edge e_i of C that is also an inner edge of H_{j-1}, f_j must be ν_i-inclusive, because some $f_h \in \{f_1, \ldots, f_{j-1}\}$ with $f_h \in \mathrm{Ds}_{\mathcal{F}}(f_j)$ is ν_i-inclusive.)
- For every inner vertex $y \in \Gamma$ of H_j that is not an inner vertex of H_{j-1}, $\psi(y) \neq \perp$ and $\psi(y)$ is a descendant of f_j in $\mathcal{F}$.
  (*Comment:* If this does not hold, then B cannot be made potential for f_j. The proof is similar to that in the last comment.
  Also note that for every nontrivial 1-component C of H_j with $E(C) \cap A_j \neq \emptyset$ and for every inner vertex $y \in \Gamma$ of C that is also an inner vertex of H_{j-1}, it holds that $\psi(y) \neq \perp$ and $\psi(y)$ is a descendant of some $f_h \in \{f_1, \ldots, f_{j-1}\}$ with $f_h \in \mathrm{Ds}_{\mathcal{F}}(f_j)$ in $\mathcal{F}$.)

(c) For each outer non-bridge e_i of H_j that is not an outer non-bridge of H_{j-1}, if either (1) the left side of B_i is labeled "surroundable" but the inner face F of H_j whose boundary contains e_i is on the right of e_i in embedding K, or (2) the right side of B_i is labeled "surroundable" but F is on the left of e_i in embedding K, then modify B by flipping B_i around its poles.
(*Comment:* Let C be the 1-component of H_j with $e_i \in E(C)$. As in the comment on Substep 6a, we can claim that C is a nontrivial 1-component and also a superface of $K[A_\mu(f_j)]$. In turn, e_i must be an outer non-bridge of $K[A_\mu(f_j)]$. Thus, the flipping in Substep 6c is necessary in order to make B potential for f_j.)

7. Label an arbitrary side of B "surroundable".

R-Procedure2: (See the next example.)

1. Same as Step 1 of R-Procedure1.
2. Compute the set $U = \{\text{x-thr}(\nu_i) \mid i \in I\} \cup \{\text{x-crt}(\mu)\}$ and $\mathcal{F}\langle U \rangle$. Let W be the set of all vertices in $\mathcal{F}\langle U \rangle$.
3. Sort the non-μ-critical vertices in W in non-increasing order of their pre-order numbers in $\mathcal{F}$; let $(f_1, \ldots, f_{h-1})$ be the resulting list. Moreover, sort the μ-critical vertices

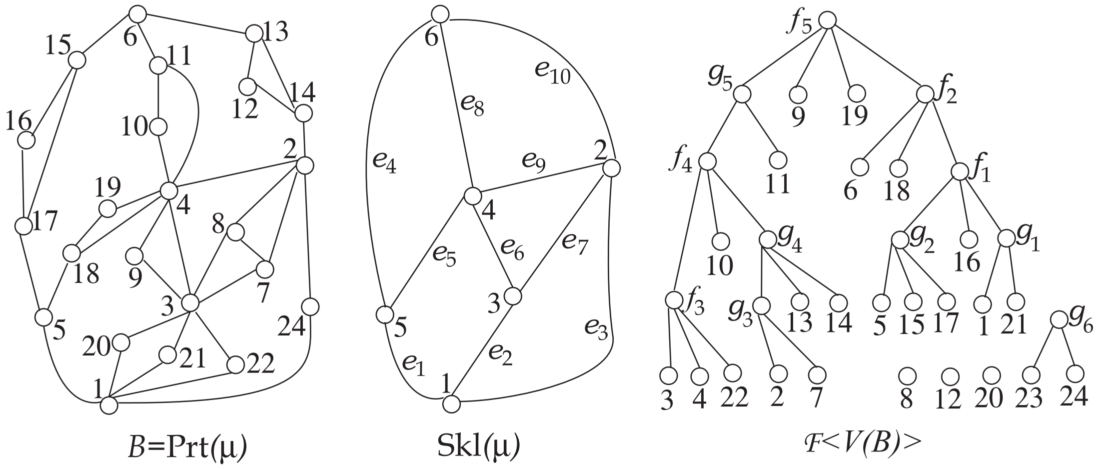

Figure 4.3: A block B of G, the skeleton of B, and forest $\mathcal{F}\langle V(B)\rangle$. (*Comment:* B has ten child blocks $B_1, \ldots, B_{10}$. We assume that the right side of each child block is labeled "surroundable", $\Gamma \cap V(B) = \{16, 18, 20\}$, $\psi(18) = f_5$ and $\psi(16) = \psi(20) = \perp$. The set I computed in Step 1 is $\{1, 2, 4, \ldots, 10\}$. The set W computed in Step 2 is $\{f_1, \ldots, f_5\}$. The sets A_1 through A_5 computed in Step 4 are as follows: $A_1 = \{e_1\}$, $A_2 = \{e_4\}$, $A_3 = \{e_6\}$, $A_4 = \{e_7, e_9\}$ and $A_5 = \{e_2, e_5, e_8, e_{10}\}$. Note that B_2, B_7, and B_{10} are flipped in Substep 6c.)

in W in non-increasing order of their pre-order numbers in $\mathcal{F}$; let $(f_h, \ldots, f_\ell)$ be the resulting list.

(*Comment:* f_h = x-crt(μ). Possibly, $h = 1$. Moreover, if $1 \le i < j \le \ell$, then f_i is not an ancestor of f_j in $\mathcal{F}\langle U\rangle$. Furthermore, by Lemmas 4.3, 4.8 and 4.9, we only need to check whether an embedding of Prt(μ) potential for f_1 through f_ℓ and surroundable for f_h through f_ℓ can be obtained from B by flipping some child blocks $B_i \in \{B_1, \ldots, B_k\}$ of B.)

4. Perform Steps 4 and 5.

5. For $j = 1, 2, \ldots, h$ (in this order), perform Substeps 6a through 6c.

6. Try to find a side P of K satisfying the following conditions:

   - P is also a path in H_h.
   - Let H'_h be the plane graph obtained from H_h by adding a new edge $e = \{s(K), t(K)\}$ on the right of P if P is the right side of K, while on the left of P otherwise. Then, for every inner edge e_i of H'_h that is on P, f_h is ν_i-inclusive. Moreover, for every inner vertex $y \in \Gamma$ of H'_h that is on P, $\psi(y) \neq \perp$ and $\psi(y)$ is a descendant of f_h in $\mathcal{F}$.

   (*Comment:* Each inner edge (respectively, inner vertex) of H'_h that is not on P is also an inner edge (respectively, inner vertex) of H_h. Moreover, the outer non-bridges of H_h that are on P become inner edges of H'_h. Furthermore, the vertices on P that are not cut vertices of H_h become inner vertices of H'_h.)

If P does not exist, then B cannot be made surroundable for f_h by flipping some child blocks of B, and so stop immediately for failure.

(*Comment:* If both sides of K satisfy the above conditions, then f_h must be μ-inclusive because of Step 5, and hence both sides of B are surroundable. Otherwise, for every embedding B' of B, only the side of B' corresponding to P can possibly be surroundable.)

7. For each outer non-bridge e_i of H'_h that is on P, if either (1) the left side of B_i is labeled "surroundable" but the inner face F of H'_h whose boundary contains e_i is on the right of e_i in embedding H'_h, or (2) the right side of B_i is labeled "surroundable" but F is on the left of e_i in embedding H'_h, then modify B by flipping B_i around its poles.

   (*Comment:* Since e_i must be an outer bridge of H_h, B_i was not flipped in Step 5.)

8. Let K' be the plane graph obtained from K by adding a new edge $e = \{s(K), t(K)\}$ on the right of P if P is the right side of K, while on the left of P otherwise. For $j = h+1, h+2, \ldots, \ell$ (in this order), perform the following substeps:

   (a) Let $H'_j = K'[A_1 \cup \cdots \cup A_j \cup \{e\}]$.
   (*Comment:* Since $f_h \in \mathrm{Ds}_{\mathcal{F}}(f_j)$, every edge e_i on P belongs to $A_\mu(f_j)$. So, for the 1-component C of H'_j with $e \in E(C)$, $C - \{e\}$ is a 1-component of $K[A_1 \cup \cdots \cup A_j]$. Thus, by the claim in the comment on Substep 6a of R-Procedure1, $E(C) - \{e\} \subseteq A_\mu(f_j)$.)

   (b) If at least one of the following does not hold, then (B cannot be made surroundable for f_j by flipping some child blocks of B, and so) stop immediately for failure.

      - Every inner face of H'_j is also an inner face of K'.
        (*Comment:* Every inner face of H'_j except the one enclosed by e and path P must be an inner face of $K[A_1 \cup \cdots \cup A_j]$, too. Also refer to the comment on the corresponding item in Substep 6b of R-Procedure1.)
      - For every inner edge e_i of H'_j that is not an inner edge of H'_{j-1}, f_j is ν_i-inclusive.
        (*Comment:* Each inner edge of H'_j that is not on P is also an inner edge of $K[A_1 \cup \cdots \cup A_j]$. Moreover, the outer non-bridges of $K[A_1 \cup \cdots \cup A_j]$ that are on P become inner edges of H'_j. Also refer to the comment on the corresponding item in Substep 6b of R-Procedure1.)
      - For every inner vertex $y \in \Gamma$ of H'_j that is not an inner vertex of H'_{j-1}, $\psi(y) \neq \perp$ and $\psi(y)$ is a descendant of f_j in $\mathcal{F}$.
        (*Comment:* Each inner inner vertex of H'_j that is not on P is also an inner vertex of $K[A_1 \cup \cdots \cup A_j]$. Moreover, the vertices on P that are not cut vertices of $K[A_1 \cup \cdots \cup A_j]$ become inner vertices of H'_j. Also refer to the comment on the corresponding item in Substep 6b of R-Procedure1.)

   (c) For each outer non-bridge e_i of H'_j that is not an outer non-bridge of H'_{j-1}, if either (1) the left side of B_i is labeled "surroundable" but the inner face F of H'_j whose boundary contains e_i is on the right of e_i in embedding K', or (2) the right side of B_i is labeled "surroundable" but F is on the left of e_i in embedding K', then modify B by flipping B_i around its poles.

9. If P is the left side of K, then label the left side of B "surroundable"; otherwise, label the right side of B "surroundable".

Example 2 Consider the block B and forest $\mathcal{F}\langle V(B)\rangle$ in Figure 4.3. Suppose that x-crt$(\mu) = f_2$, $\Gamma = \{16, 18, 20\}$, $\psi(16) = \psi(18) = f_5$, and and $\psi(20) = \perp$. Then, the sets W and A_1 through A_5 are as before. However, not only B_2, B_7, and B_{10} are flipped in Step 5, but also B_1 and B_4 are flipped in Substep 8c.

It is easy to verify that after μ is processed without failure, the invariant in Section 4.2 holds for μ.

5 Implementation

This section proves the following theorem:

Theorem 5.1 *The algorithm in Section 4 runs in $O(|G| \log |G|)$ time.*

5.1 Preprocessing

We identify the nodes in T by their pre-order numbers. By this, given two nodes μ and ν in T, we can determine in $O(1)$ time whether μ is an ancestor of ν, or ν is an ancestor of μ, or $\mathrm{LCA}_T(\mu, \nu) \in \{\mu, \nu\}$. We preprocess T in linear time to set up a data structure so that, given two nodes μ and ν, we can find $\mathrm{LCA}_T(\mu, \nu)$ in $O(1)$ time [14, 24]. Similar pre-processing is also done for the forest $\mathcal{F}$ so that the same queries can be answered in $O(1)$ time too.

For each vertex f in $\mathcal{F}$, let $\mathcal{F}_f$ denote the subtree of $\mathcal{F}$ rooted at f. We associate with f an interval $[l_f, r_f]$, where l_f (respectively, r_f) is the vertex in $\mathcal{F}_f$ with the smallest (respectively, largest) pre-order number. The intervals $[l_f, r_f]$ for all vertices f in $\mathcal{F}$ are pre-computed and stored, in $O(|\mathcal{F}|)$ time. Note that f is an ancestor of g in $\mathcal{F}$ if and only if the interval $[l_f, r_f]$ contains the interval $[l_g, r_g]$.

Let $\mathcal{V}$ be a set of vertices in $\mathcal{F}$. $\mathcal{F}[\mathcal{V}]$ denotes the forest obtained from $\mathcal{F}$ as follows: The vertex set of $\mathcal{F}[\mathcal{V}]$ is $\mathcal{V}$. A vertex $f \in \mathcal{V}$ is the parent of a vertex $g \in \mathcal{V}$ in $\mathcal{F}[\mathcal{V}]$ if and only if (1) f is an ancestor of g in $\mathcal{F}$ and (2) no vertex in $\mathcal{V} - \{f, g\}$ appears on the path between f and g in $\mathcal{F}$.

Lemma 5.2 *Given a set $\mathcal{V}$ of vertices of $\mathcal{F}$, it takes $O(|\mathcal{V}| \log |\mathcal{F}|)$ time to construct $\mathcal{F}[\mathcal{V}]$.*

Proof: For two intervals $[l, r]$ and $[l', r']$, if either $l < l'$, or $l = l'$ and $r > r'$, then we write $[l, r] \prec [l', r']$. Suppose $\mathcal{V} = \{f_1, \ldots, f_k\}$. For each f_i, let $[l_i, r_i]$ be its associated interval. We can sort $\mathcal{V}$ so that whenever a vertex f_i precedes another vertex f_j in the sorted $\mathcal{V}$, $[l_i, r_i] \prec [l_j, r_j]$. So, without loss of generality, assume $[l_1, r_1] \prec \ldots \prec [l_k, r_k]$.

We process $f_1, f_2, \ldots, f_k$ in this order. Initially, f_1 is the root of the first tree in $\mathcal{F}[\mathcal{V}]$. It will be an invariant that, right before the processing of f_i with $i \geq 2$, the path P from f_{i-1} to the root of the tree in $\mathcal{F}[\mathcal{V}]$ containing f_{i-1} is always available. We call P the *frontier path.*

Now consider the processing of f_i where $i \geq 2$. Note that f_{i-1} is the unique leaf of the frontier path P. Let f_j be the root of P. We check if $[l_j, r_j]$ contains $[l_i, r_i]$. If no, then f_i is the root of a new tree in $\mathcal{F}[\mathcal{V}]$ and we update the frontier path P to be a single vertex f_i. Otherwise, we do a binary search on P to find the lowest vertex f_{i_p} on P such that $[l_{i_p}, r_{i_p}]$ contains $[l_i, r_i]$. We then let f_i be a new child of f_{i_p} and update P accordingly.

The processing of each vertex in $\mathcal{V}$ takes $O(\log |\mathcal{F}|)$ time. So the total time needed to construct $\mathcal{F}[\mathcal{V}]$ is $O(|\mathcal{V}| \log |\mathcal{F}|)$. ▮

Lemma 5.3 *Given a set $\mathcal{V}$ of vertices of $\mathcal{F}$, it takes $O(|\mathcal{V}| \log |\mathcal{F}|)$ time to construct $\mathcal{F}\langle\mathcal{V}\rangle$.*

Proof: We may assume that each vertex f of $\mathcal{F}$ knows the root of the tree in $\mathcal{F}$ containing f, because this can be achieved by an $O(|\mathcal{F}|)$-time pre-processing. Then, we can further assume that the vertices of $\mathcal{V}$ appear in a single tree in $\mathcal{F}$; otherwise we can partition $\mathcal{V}$ suitably and work on each subset in the resulting partition.

Let $\mathcal{W}$ be the vertex set of $\mathcal{F}\langle\mathcal{V}\rangle$. Recall that $\mathcal{W} = \mathcal{V} \cup \{f \notin \mathcal{V} \mid f$ has at least two children g in $\mathcal{F}$ with $\mathrm{Ds}_{\mathcal{F}}(g) \cap \mathcal{V} \neq \emptyset\}$. Since $\mathcal{F}\langle\mathcal{V}\rangle = \mathcal{F}[\mathcal{W}]$ by definition, Lemma 5.2 can be applied if we know $\mathcal{W}$. Obviously, $\mathcal{W} = \mathcal{V} \cup \{\mathrm{LCA}_{\mathcal{F}}(f, g) \mid f \in \mathcal{V}$ and $g \in \mathcal{V}\}$. To compute $\mathcal{W}$, we first sort the vertices of $\mathcal{V}$ in increasing order of their post-order numbers; let $f_1, \ldots, f_k$ be the resulting list. We claim that $\{\mathrm{LCA}_{\mathcal{F}}(f_i, f_j) \mid 1 \leq i \neq j \leq k\} = \{\mathrm{LCA}_{\mathcal{F}}(f_i, f_{i+1}) \mid 1 \leq i \leq k-1\}$. The lemma follows from the claim immediately. To see the claim, first note that $\{\mathrm{LCA}_{\mathcal{F}}(f_i, f_{i+1}) \mid 1 \leq i \leq k-1\} \subseteq \{\mathrm{LCA}_{\mathcal{F}}(f_i, f_j) \mid 1 \leq i \neq j \leq k\}$. To see that $\{\mathrm{LCA}_{\mathcal{F}}(f_i, f_j) \mid 1 \leq i \neq j \leq k\} \subseteq \{\mathrm{LCA}_{\mathcal{F}}(f_i, f_{i+1}) \mid 1 \leq i \leq k-1\}$, let f_i and f_j be two distinct vertices in $\mathcal{V}$ such that $\mathrm{LCA}_{\mathcal{F}}(f_i, f_j) \neq \perp$ and $j > i+1$. Let $g = \mathrm{LCA}_{\mathcal{F}}(f_i, f_j)$. It suffices to show that there is an $\ell \in \{i, i+1, \ldots, j\}$ with $g = \mathrm{LCA}_{\mathcal{F}}(f_\ell, f_{\ell+1})$. If $g \in \{f_i, f_j\}$, then $g = f_i$ and $g = \mathrm{LCA}_{\mathcal{F}}(f_i, f_{i+1})$, and we are done. So, suppose that $g \notin \{f_i, f_j\}$. Let g_1 be the child of g in $\mathcal{F}$ with $f_i \in \mathrm{Ds}_{\mathcal{F}}(g_1)$. Let ℓ be the largest integer such that $f_\ell \in \mathrm{Ds}_{\mathcal{F}}(g_1)$. Obviously, $i \leq \ell \leq j-1$. Moreover, it is easy to see that $g = \mathrm{LCA}_{\mathcal{F}}(f_\ell, f_{\ell+1})$. ▮

Lemma 5.4 *It takes $O(|G| \log |G|)$ total time to compute x-thr(μ) for all nodes μ in T.*

Proof: We process the nodes of T in post-order. Consider the processing of a node μ in T. Let $B = \mathrm{Prt}(\mu)$. Let $x = s(B)$ and $y = t(B)$. If x and y are in different trees in $\mathcal{F}$, then there is no μ-through vertex in $\mathcal{F}$ and we are done. So, suppose that x and y are in the same tree in $\mathcal{F}$. Let $f_1 = \mathrm{LCA}_{\mathcal{F}}(x, y)$. If μ is a Q-node, then f_1 is x-thr(μ). So, assume that μ is not a Q-node.

Let $\nu_1, \ldots, \nu_k$ be the children of μ in T. For each $i \in \{1, \ldots, k\}$, let e_i be the virtual edge of ν_i in $\mathrm{Skl}(\mu)$, and $g_i = $ x-thr(ν_i). Note that when μ is processed, we already know g_i. Let $\mathcal{V} = \{f_1\} \cup \{g_i \mid i \in \{1, \ldots, k\}$ and $g_i \neq \perp\}$. Let $P = f_1, f_2, \ldots, f_p$ be the path from f_1 to the root ancestor of f_1 in $\mathcal{F}\langle\mathcal{V}\rangle$. If $f = $ x-thr(μ) exists, it must be on P. We compute P in $O(|\mathrm{Skl}(\mu)|)$ time, and then do a binary search on P to find f. In more details, we proceed in $\lfloor \log p \rfloor + 1 \leq \lfloor \log |G| \rfloor + 1$ stages. In the first stage, we determine whether $f_{\lceil p/2 \rceil}$ is a μ-through vertex or not as follows. For each $i \in \{1, \ldots, k\}$ with $g_i \neq \perp$, if $f_{\lceil p/2 \rceil}$ is an ancestor of g_i, then $f_{\lceil p/2 \rceil}$ is a ν_i-through vertex and we mark e_i in $\mathrm{Skl}(\mu)$. We then test if $s(B)$ and $t(B)$ can reach each other in $\mathrm{Skl}(\mu)$ via marked edges only. If yes, $f_{\lceil p/2 \rceil}$ is a μ-through vertex; otherwise, it is not. This completes the first stage. Clearly, this stage takes $O(|\mathrm{Skl}(\mu)|)$ time. In the second stage, if $f_{\lceil p/2 \rceil}$ is a μ-through vertex, we test whether $f_{\lceil\lceil p/2\rceil/2\rceil}$ is a μ-through vertex or not. Otherwise, we test whether $f_{\lceil(p-\lceil p/2\rceil)/2\rceil}$ is a μ-through vertex or not. Continue this until we get x-thr(μ).

The total time needed to find x-thr(μ) is thus $O(|\mathrm{Skl}(\mu)|\log|G|)$. So the total time needed for all nodes μ is $O(|G|\log|G|)$ as to be shown. ▮

Lemma 5.5 *It takes $O(|G|\log|G|)$ total time to compute x-out-thr(μ) for all nodes μ in T.*

Proof: We process the nodes of T in pre-order. Before a node μ in T is processed, we already know x-out-thr(μ). While processing μ, we compute x-out-thr(ν) for all children ν of μ. Initially, before the root ρ of T is processed, we know x-out-thr(ρ) $=\perp$.

Consider the processing of a node μ. Let $g_0=$ x-out-thr(μ). Let $\nu_1,\ldots,\nu_k$ be the children of μ. For each $i\in\{1,\ldots,k\}$, let $g_i=$ x-thr(ν_i), and e_i be the virtual edge of ν_i in Skl(μ).

Case 1: μ is a P-node. Let A be the multiset $\{g_i \mid 0\le i\le k \text{ and } g_i\ne\perp\}$. Let $|A|$ be the number of (possibly not distinct) elements in A. If $|A|=0$, then x-out-thr(ν_i) $=\perp$ for all $i\in\{1,\ldots,k\}$. If $|A|=1$ and $A=\{g_0\}$, then x-out-thr(ν_i) $=g_0$ for all $i\in\{1,\ldots,k\}$. If $|A|=1$ and $A=\{g_i\}$ with $i\in\{1,\ldots,k\}$, then x-out-thr(ν_i) $=\perp$ and x-out-thr(ν_j) $=g_i$ for all $j\in\{1,\ldots,i-1,i+1,\ldots,k\}$. Next, suppose $|A|\ge 2$. Then, there are two vertices f_1 and f_2 in A such that f_2 is an ancestor of f_1 in $\mathcal{F}$ and A consists of some vertices on the path from f_1 to f_2 in $\mathcal{F}$. We sort A so that its elements appear in non-increasing order of their pre-order numbers. Then, f_1 and f_2 are the first and the last elements in A. Let f_3 be the second element in A. If $f_1=g_0$, then x-out-thr(ν_i) $=g_0$ for all $i\in\{1,\ldots,k\}$. If $f_1\ne g_0$ and $f_1=f_3$, then x-out-thr(ν_i) $=f_1$ for all $i\in\{1,\ldots,k\}$. If $f_1\ne f_3$ and $f_1=g_i\ne g_0$, then x-out-thr(ν_i) $=f_3$ and x-out-thr(ν_j) $=f_1$ for all $j\in\{1,\ldots,i-1,i+1,\ldots,k\}$.

Case 2: μ is an R-node. This is the nontrivial case. If $g_0=\perp$, then let $D=\mathrm{Skl}(\mu)$; otherwise, let D be the graph obtained from Skl(μ) by adding a new edge e_0 between the poles of Skl(μ). Since μ is an R-node, D is a simple graph. Let $\mathcal{U}$ be the set $\{g_i \mid i\in\{0,\ldots,k\} \text{ and } g_i\ne\perp\}$, and $\mathcal{W}=\mathcal{U}\cup V(D)$. For each $g_i\in\mathcal{U}$, we call e_i a *g_i-edge*. Note that there may be two or more g_i-edges in D.

First, we claim that for every path $P=x_1,e_{i_1},\ldots,x_l,e_{i_\ell},x_{\ell+1}$ in D with $\ell\ge 1$ and $g_{i_j}\ne\perp$ for all $j\in\{1,\ldots,\ell\}$, $\mathrm{LCA}_{\mathcal{F}}(g_{i_1},\ldots,g_{i_\ell})\in\{g_{i_1},\ldots,g_{i_\ell}\}$. This claim is proved by induction on ℓ as follows. It trivially holds when $\ell=1$. Suppose $\ell\ge 2$ and assume that $\mathrm{LCA}_{\mathcal{F}}(g_{i_1},\ldots,g_{i_{\ell-1}})=g_{i_h}\in\{g_{i_1},\ldots,g_{i_{\ell-1}}\}$. Since $x_\ell\in \mathrm{Lf}_{\mathcal{F}}(g_{i_{\ell-1}})\cap \mathrm{Lf}_{\mathcal{F}}(g_{i_\ell})$, both $g_{i_{\ell-1}}$ and g_{i_ℓ} are ancestors of x in $\mathcal{F}$. In turn, both g_{i_h} and g_{i_ℓ} are ancestors of x in $\mathcal{F}$ because g_{i_h} is an ancestor of $g_{i_{\ell-1}}$ in $\mathcal{F}$. Thus, $\mathrm{LCA}_{\mathcal{F}}(g_{i_h},g_{i_\ell})\in\{g_{i_h},g_{i_\ell}\}$. Now, the claim follows from the fact that $\mathrm{LCA}_{\mathcal{F}}(g_{i_1},\ldots,g_{i_\ell})=\mathrm{LCA}_{\mathcal{F}}(g_{i_h},g_{i_\ell})$.

Obviously, for every $j\in\{1,\ldots,k\}$, if x-out-thr(ν_j) $\ne\perp$, then there is a path $P=x_1,e_{i_1},\ldots,x_l,e_{i_\ell},x_{\ell+1}$ in $D-\{e_j\}$ between the two poles x_1 and $x_{\ell+1}$ of ν_j such that x-out-thr(ν_j) $=\mathrm{LCA}_{\mathcal{F}}(g_{i_1},\ldots,g_{i_\ell})$. So, by the above claim, for every $i\in\{1,\ldots,k\}$, if x-out-thr(ν_i) $\ne\perp$, then x-out-thr(ν_i) $\in\mathcal{U}$. Thus, we construct $\mathcal{F}[\mathcal{W}]$. Note that $V(D)$ is the set of all leaves in $\mathcal{F}[\mathcal{W}]$. By Lemma 5.2, this takes $O(|\mathcal{W}|\log|G|)=O(|\mathrm{Skl}(\mu)|\log|G|)$ time.

For each non-leaf vertex a in $\mathcal{F}[\mathcal{W}]$, let $D(a)$ be the subgraph of D induced by the set of all edges e_i in D such that g_i is a descendant of a in $\mathcal{F}[\mathcal{W}]$. Note that if a and b are two vertices in $\mathcal{F}[\mathcal{W}]$ with $\mathrm{LCA}_{\mathcal{F}[\mathcal{W}]}(a,b)\notin\{a,b\}$, then $D(a)$ and $D(b)$ are vertex disjoint because the existence of a vertex v in both $D(a)$ and $D(b)$ would imply that $v \in \mathrm{Lf}_{\mathcal{F}}(a) \cap \mathrm{Lf}_{\mathcal{F}}(b)$.

For every $i \in \{1, \ldots, k\}$, let $a_i = \mathrm{LCA}_{\mathcal{F}[\mathcal{W}]}(x_i, y_i)$, where $\{x_i, y_i\} = e_i$. We claim that if $a_i \neq \perp$ and $a_i \neq g_i$, then x-out-thr$(\nu_i) = a_i$. To see this, suppose that $a_i \neq \perp$ and $a_i \neq g_i$. Recall that $(G, \mathcal{F})$ passes the connectivity-checking. Since $\{x_i, y_i\} \subseteq \mathrm{Lf}_{\mathcal{F}}(a_i)$ but $\{x_i, y_i\} \not\subseteq \mathrm{Lf}_{\mathcal{F}}(b)$ for every child b of a_i in $\mathcal{F}[\mathcal{W}]$, x_i and y_i are reachable from each other in $G[\mathrm{Lf}_{\mathcal{F}}(a_i)]$ and hence in $D(a_i)$, but are not in $D(b)$. By this, if $g_i = \perp$, then clearly x-out-thr$(\nu_i) = a_i$. In case $g_i \neq \perp$, $g_i =$ x-thr(ν_i) must be an ancestor of a_i in $\mathcal{F}[\mathcal{W}]$ and hence x-out-thr$(\nu_i) = a_i$, too. This establishes the claim. Based on this claim, for every $i \in \{1, \ldots, k\}$, if $a_i \neq \perp$ and $a_i \neq g_i$, we set x-out-thr$(\nu_i) = a_i$; otherwise, we initialize x-out-thr$(\nu_i) = \perp$.

Next, we process the vertices of $\mathcal{F}[\mathcal{U}]$ in post-order. Note that $\mathcal{F}[\mathcal{U}]$ can be obtained from $\mathcal{F}[\mathcal{W}]$ by deleting all leaves. Consider the processing of a vertex a in $\mathcal{F}[\mathcal{U}]$. In case a is a leaf vertex in $\mathcal{F}[\mathcal{U}]$, we construct $D(a)$ and compute the 2-components of $D(a)$; after this, for every nontrivial 2-component C of $D(a)$ and every edge e_i in C, if x-out-thr$(\nu_i) = \perp$, then we reset x-out-thr$(\nu_i) = a$. Next, suppose that a is a non-leaf vertex in $\mathcal{F}[\mathcal{U}]$. Let $b_1, \ldots, b_\ell$ be the children of a in $\mathcal{F}[\mathcal{U}]$. $D(a)$ can be obtained from $D(b_1), \ldots, D(b_\ell)$ by adding the a-edges of D. Using the data structures for incremental biconnectivity testing discussed in [27], we can enumerate the edges e_i of $D(a)$ such that e_i belongs to a nontrivial 2-component of $D(a)$ but belongs to no nontrivial 2-component of $D(b_j)$ for all $j \in \{1, \ldots, \ell\}$, in $O(1)$ time per edge. For each such edge e_i, if x-out-thr$(\nu_i) = \perp$, then reset x-out-thr$(\nu_i) = a$.

After all vertices in $\mathcal{F}[\mathcal{U}]$ have been processed, x-out-thr(ν_1) through x-out-thr(ν_k) have been correctly obtained. This completes Case 2.

Case 3: μ is an S-node. This is a special case of Case 2 where $\mathrm{Skl}(\mu)$ is a path. So it can be handled by the same algorithm in Case 2.

The time needed to process a node μ is $O(|\mathrm{Skl}(\mu)| \log |G|)$. So the time needed to process all nodes is $O(|G| \log |G|)$. ▮

Lemma 5.6 *It takes $O(|G| \log |G|)$ total time to compute x-crt(μ) for all nodes μ in T.*

Proof: For each μ, let $f_1 =$ x-thr(μ) and $f_2 =$ x-out-thr(μ). Then x-crt$(\mu) = \mathrm{LCA}_{\mathcal{F}}(f_1, f_2)$. The lemma follows from Lemmas 5.4 and 5.5 immediately. ▮

Lemma 5.7 *It takes $O(|G|)$ total time to compute x-inc(μ) for all nodes μ in T.*

Proof: We process the nodes μ of T in post-order. If μ is a Q-node, then obviously x-inc$(\mu) = \mathrm{LCA}_{\mathcal{F}}(s(\mathrm{Prt}(\mu)), t(\mathrm{Prt}(\mu)))$, which takes $O(1)$ time to compute. Suppose that μ is not a Q-node. Let $\nu_1, \ldots, \nu_k$ be the children of μ in T. Let $g_i =$ x-inc(ν_i) for $i \in \{1, \ldots, k\}$, which has been computed. If $g_i = \perp$ for some $i \in \{1, \ldots, k\}$, then obviously x-inc$(\mu) = \perp$. So, assume that $g_i \neq \perp$ for all $i \in \{1, \ldots, k\}$. We compute x-inc(μ) in $O(k)$ time as follows.

Case 1: μ is an S-node. Let $x_1, \ldots, x_h$ be the nonpoles of $\mathrm{Skl}(\mu)$ that are also contained in Γ. If $\psi(x_j) = \perp$ for some $j \in \{1, \ldots, h\}$, then $\mathcal{F}$ has no μ-inclusive vertex and we set x-inc$(\mu) = \perp$. Otherwise, we can set x-inc$(\mu) = \mathrm{LCA}_{\mathcal{F}}(g_1, \ldots, g_k, \psi(x_1), \ldots, \psi(x_h))$.

Case 2: μ is a P-node. Obviously, x-inc$(\mu) = \mathrm{LCA}_{\mathcal{F}}(g_1, \ldots, g_k)$.

Case 3: μ is an R-node. Let $y_1, \ldots, y_p$ be the nonpoles of $\mathrm{Skl}(\mu)$ that are also contained in Γ. If $\psi(y_j) = \perp$ for some $j \in \{1, \ldots, p\}$, then x-inc$(\mu) = \perp$; otherwise, x-inc$(\mu) = \mathrm{LCA}_{\mathcal{F}}(g_1, \ldots, g_k, \psi(v_1), \ldots, \psi(v_p))$.

By Lemma 4.2, the total time needed to compute all x-inc(μ) is $O(|G|)$. ∎

5.2 Details of implementation

Obviously, the processing of an S-node μ described in Section 4.3 can be done in $O(|\mathrm{Skl}(\mu)|)$ time. It is also easy to implement the two procedures in Section 4.4 in $O(|\mathrm{Skl}(\mu)|)$ time, using the pre-computed information. We describe the implementation details of the procedures in Section 4.5 below.

We inherit the notations in Section 4.5. We may assume that the vertices of $\mathcal{G}$ are $1, 2, \ldots, |V(\mathcal{G})|$.

Consider the implementation of R-Procedure1. Obviously, I can be computed in $O(|\mathrm{Skl}(\mu)|)$ time. U can be computed from I via bucket sort in $O(|I|) = O(|\mathrm{Skl}(\mu)|)$ time. Given U, $\mathcal{F}\langle U\rangle$ can be computed in $O(|\mathrm{Skl}(\mu)| \log |\mathcal{F}|)$ time by Lemma 5.3. Note that $|W| = O(|U|)$. So, S can be obtained from W via bucket sort in $O(|W|) = O(|\mathrm{Skl}(\mu)|)$ time. Similarly, $A_1, \ldots, A_\ell$ can be computed from I via bucket sort in $|I| = O(|\mathrm{Skl}(\mu)|)$ time.

In light of the preprocessing in Section 5.1, the hard tasks in Step 6 of R-Procedure1 are the following:

- Construct H_j.

  [*Implementation details:* Although H_j is a plane graph, it is unnecessary to compute its plane embedding. Instead, we only compute $E(H_j)$ and the number $\#_c(H_j)$ of connected components in H_j. Since $V(H_j) = V(\mathrm{Skl}(\mu))$, we can assume that $V(H_j)$ is available for free. Obviously, we can compute $\#_c(H_0)$ and $E(H_0)$ in $O(|\mathrm{Skl}(\mu)|)$ time. We use a union-find data structure to maintain a collection of disjoint dynamic subsets of $V(H_j)$. Each subset of $V(H_j)$ in the data structure consists of the vertices of a 1-component of H_j. The data structure for H_0 can be trivially constructed in $O(|\mathrm{Skl}(\mu)|)$ time. Given $\#_c(H_{j-1})$ and the data structure for H_{j-1} with $j \geq 1$, we can compute $\#_c(H_j)$ and the data structure for H_j in time almost linear in $|A_j|$. Thus, the total time needed to compute $\#_c(H_0), E(H_0), \ldots, \#_c(H_\ell), E(H_\ell)$ is almost linear in $|\mathrm{Skl}(\mu)|$.]

- Check whether every inner face of H_j is also an inner face of $\mathrm{Skl}(\mu)$.

  [*Implementation details:* Note that H_0 has no inner face. Suppose that $j \geq 1$ and every inner face of H_{j-1} is indeed also an inner face of $\mathrm{Skl}(\mu)$. We may assume that $\#_c(H_j)$, $|V(H_j)|$ and $|E(H_j)|$ have been computed as described in the last item. Given $\#_c(H_j)$, $|V(H_j)|$ and $|E(H_j)|$, we can use Euler's formula to compute the number $\#_f(H_j)$ of inner faces in H_j in $O(1)$ time. Similarly, we can assume that $\#_f(H_{j-1})$ has been computed before. Note that each inner face of H_{j-1} is also an inner face of H_j. So, it suffices to check whether $\#_f(H_j) - \#_f(H_{j-1})$ equals the number n_j of inner faces of $\mathrm{Skl}(\mu)$ that are also inner faces of H_j but are not inner faces of H_{j-1}. We will describe the computation of n_j later.]

- Compute the set X_j of all inner edges of H_j that are not inner edges of H_{j-1}.

- Compute the set Y_j of all inner vertices of H_j contained in Γ that are not inner vertices of H_{j-1} contained in Γ.

- Compute the set Z_j of all outer non-bridges of H_j that are not outer non-bridges of H_{j-1}.

It remains to consider how to compute n_j, X_j, Y_j, Z_j for $j = 1, 2, \ldots, \ell$. Right before this computation, we perform the following preprocessing:

- Compute the number $\#_f(\mathrm{Skl}(\mu))$ of inner faces of $\mathrm{Skl}(\mu)$.
- Index the inner faces of $\mathrm{Skl}(\mu)$ with the numbers $1, 2, \ldots, \#_f(\mathrm{Skl}(\mu))$.
- For $i = 1, 2, \ldots, \#_f(\mathrm{Skl}(\mu))$, creat a list F_i of all edges in the ith inner face of $\mathrm{Skl}(\mu)$, and a counter $c_f(i)$; initialize $c_f(i)$ to be the number of edges in F_i.
- For each edge e_h of $\mathrm{Skl}(\mu)$, creat a counter $c_e(h)$, initialize $c_e(h) = 0$, and creat a pointer to each counter $c_f(i)$ such that e_h is an edge in F_i.

  (*Comment:* e_h points to at most two counters, $c_e(h)$ will be used to memorize the number of inner faces in H_j whose boundaries contain e_h.)
- For each vertex y of $\mathrm{Skl}(\mu)$ contained in Γ, creat a counter $c_v(y)$ and initialize $c_v(y)$ to be the number of edges incident to vertex y in $\mathrm{Skl}(\mu)$.

Note that the above preprocessing can be done in $O(|\mathrm{Skl}(\mu)|)$ time. For $j = 1, 2, \ldots, \ell$ (in this order), we compute n_j, X_j, Y_j, Z_j by performing the following steps in turn:

1. Initialize $n_j = 0$, $X_j = Y_j = Z_j = \emptyset$.
2. For each edge $e_h \in A_j$, subtract 1 from each counter $c_f(i)$ to which e_h points; if the value of $c_f(i)$ just became 0, then (F_i is an inner face of H_j but is not an inner face of H_{j-1}, and so) add 1 to n_j, add 1 to each counter $c_e(h')$ such that $e_{h'}$ is an edge in F_i, and subtract 1 from each counter $c_v(y)$ such that $y \in \Gamma$ and y is incident to an edge in F_i.
3. For each edge $e_{h'}$ such that the value of $c_e(h')$ just became 2 in Step 2, (since $e_{h'}$ is an inner edge of H_j but not an inner edge of H_{j-1}) add $e_{h'}$ to X_j.
4. For each vertex $y \in \Gamma$ such that the value of $c_v(y)$ just became 0 in Step 2, (since $y \in \Gamma$ is an inner vertex of H_j but not an inner vertex of H_{j-1}) add y to Y_j.
5. For each edge $e_{h'}$ such that the value of $c_e(h')$ just became 1 at the end of Step 2, (since $e_{h'}$ is an outer non-bridge of H_j but not an outer non-bridge of H_{j-1}) add $e_{h'}$ to Z_j.

In the above computation, the following always hold:

- The value of no counter $c_f(i)$ can become negative.
- No edge $e_{h'}$ can be added to X_j more than once.
- No vertex y can be added to Y_j more than once.
- No edge $e_{h'}$ can be added to Z_j more than once.

From these observations, it is easy to see that the above computation takes $O(|\mathrm{Skl}(\mu)|)$ time. In turn, after the processing in Section 5.1, R-Procedure1 runs in $O(|\mathrm{Skl}(\mu)| \log |\mathcal{F}|)$ time. Similarly, R-Procedure2 runs in $O(|\mathrm{Skl}(\mu)| \log |\mathcal{F}|)$ time. So, by Lemma 4.2, Theorem 5.1 holds. Combining this theorem, Lemmas 3.2 and 3.3, and Theorem 3.7, we finally have the following:

Theorem 5.8 *The CDE problem can be solved in $O(n \log^2 n)$ time, where n is the length of the input.*

References

[1] 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.

[2] J. Boyer and W. Myrvold. Stop minding your P's and Q's: A simplified $O(n)$ planar embedding algorithm. In *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 140–146, 1999.

[3] Z.-Z. Chen. Approximating unweighted connectivity problems in parallel. *Information and Computation*, 171:125–136, 2001.

[4] Z.-Z. Chen, M. Grigni, and C. H. Papadimitriou. Map graphs. *Journal of the ACM*, 49(2):127-138, 2002. A preliminary version appeared as "Planar map graphs" in *Proceedings of the 30th Annual ACM Symposium on the Theory of Computing*, pages 514–523, 1998.

[5] Z-Z. Chen, X. He, and M.-Y. Kao. Common-face embeddings of planar graphs. *SIAM Journal on Computing*, 32(2):408-434, 2003. A preliminary version appeared as "Non-planar topological inference and political-map graphs" in *Proceedings of the 10th ACM-SIAM Symposium on Discrete Algorithms*, pages 195–204, 1999.

[6] G. Di Battista and R. Tamassia. On-line maintenance of triconnected components with *spqr*-trees. *Algorithmica*, 15:302–318, 1996.

[7] G. Di Battista and R. Tamassia. On-line planarity testing. *SIAM Journal on Computing*, 25(5):956–997, 1996.

[8] M.J. Egenhofer. Reasoning about binary topological relations. In O. Gunther and H.J. Schek, editors, *Proceedings of the 2nd International Symposium on Large Spatial Databases (Advances in Spatial Database)*, pages 143–160, 1991.

[9] M.J. Egenhofer and J. Sharma. Assessing the consistency of complete and incomplete topological information. *Geographical Systems*, 1:47–68, 1993.

[10] G. Ehrlich, S. Even, and R.E. Tarjan. Intersection graphs of curves in the plane. *Journal of Combinatorial Theory, Ser. B*, 21(1):8–20, 1976.

[11] S. Even and R.E. Tarjan. Computing an *st*-numbering. *Theoretical Computer Science*, 2:339–344, 1976.

[12] G.N. Frederickson. Using cellular graph embeddings in solving all pairs shortest paths problems. *Journal of Algorithms*, 19(1):45–85, 1995.

[13] M. Grigni, D. Papadias, and C.H. Papadimitriou. Topological inference. In *Proceedings of the 14th International Joint Conferences on Artificial Intelligence*, pages 901–906, 1995.

[14] D. Harel and R. Tarjan. Fast algorithms for finding nearest common ancestors. *SIAM Journal on Computing*, 13:338–355, 1984.

[15] G. Kant and X. He. Regular edge labeling of 4-connected plane graphs and its applications in graph drawing problems. *Theoretical Computer Science*, 172:175–193, 1997.

[16] J. Kratochvíl. String graphs II: Recognizing string graphs is NP-hard. *Journal of Combinatorial Theory, Ser. B*, 52(1):67–78, 1991.

[17] B. Mohar. Embedding graphs in an arbitrary surface in linear time. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pages 392–397, 1996.

[18] T. Nishizeki and N. Chiba. *Planar Graphs: Theory and Algorithms.* North-Holland, 1988.

[19] J. Nešetřil and M. Rosenfeld. Embedding graphs in Euclidean spaces, an exploration guided by Paul Erdős. *Geombinatorics*, 6(4):143–155, 1997.

[20] D. Papadias and T. Sellis. The qualitative representation of spatial knowledge in two-dimensional space. *Very Large Data Bases Journal*, 4:479–516, 1994.

[21] C.H. Papadimitriou, D. Suciu, and V. Vianu. Topological queries in spatial databases. In *Proceedings of the 15th ACM Symposium on Principles of Database Systems*, pages 81–92, 1996.

[22] M. Schaefer and D. Štefankovič. Decidability of string graphs. In *Proceedings of the 33rd Annual ACM Symposium Theory of Computing*, pages 241–246, 2001.

[23] M. Schaefer, E. Sedgwick, and D. Štefankovič. Recognizing string graphs in NP. In *Proceedings of the 34th Annual ACM Symposium Theory of Computing*, pages 1–6, 2002.

[24] B. Schieber and U. Vishkin. On finding lowest common ancestors: Simplification and parallelization. *SIAM Journal on Computing*, 17:1253–1262, 1988.

[25] T.R. Smith and K.K. Park. Algebraic approach to spatial reasoning. *International Journal on Geographical Information Systems*, 6:177-192, 1992.

[26] I.G. Tollis. Graph drawing and information visualization. *ACM Computing Surveys*, 28(4es):19, 1996.

[27] J. Westbrook and R.E. Tarjan. Maintaining bridge-connected and biconnected components on-line. *Algorithmica*, 7:433–464, 1992.