Optimal Labeling for Connectivity Checking in Planar Networks with Obstacles

Bruno Courcelle [a,1,2] Cyril Gavoille [a,1]
Mamadou Moustapha Kanté [a,1] Andrew Twigg [b,3,4]

[a] *Université de Bordeaux, LaBRI, CNRS*
351 cours de la Libération
33405 Talence, France

[b] *Computing Laboratory, Oxford University, UK*

Abstract

We consider the problem of determining in a planar graph G whether two vertices x and y are linked by a path that avoids a set X of vertices and a set F of edges. We attach labels to vertices in such a way that this fact can be determined from the labels of x and y, the vertices in X and the ends of the edges of F. For a planar graph with n vertices, we construct labels of size $O(\log n)$. The problem is motivated by the need to quickly compute alternative routes in networks under node or edge failures.

Key words: Connectivity Query; Labeling Scheme; Planar Graph.

1 Introduction

We are interested in constructing labeling schemes to answer 'extended connectivity queries' on a graph G. An extended connectivity query takes a pair of vertices u, v and a set of vertices X, and answers whether u, v are connected in G. We want to do this by precomputing the graph, and assigning a short *label* to every vertex. Then, given only the information in the labels for u, v, X, we want to answer the extended connectivity query. This problem

[1] Supported by the GRAAL project of "Agence Nationale pour la Recherche".
[2] Member of "Institut Universitaire de France".
[3] Supported by ITA project, US Army Research laboratory and the UK Ministry of Defence, W911NF-06-3-0001.
[4] Supported by St John's College Oxford Junior Research Fellowship.

is motivated by the need to make repeated and fast connectivity queries on networks that may suffer failures, or in emergency planning situations, where there is no time to recompute data structures when the network changes.

We will show how to compute labels of size $O(\log n)$ bits, so that we can answer extended connectivity queries efficiently on general planar graphs. This paper extends the result of Courcelle et al. [6], which showed how to solve the problem on 3-connected planar graphs. Extending the result to general planar graphs requires some extra machinery and techniques.

A *labeling scheme* for a property $P(x_1, ..., x_k)$ of vertices $x_1, ..., x_k$ of a graph G belonging to a class $\mathcal{C}$ consists of two algorithms: a labeling algorithm $\mathcal{A}$ and a query algorithm $\mathcal{B}$. Algorithm $\mathcal{A}$ takes as input a graph G in $\mathcal{C}$ and computes a label $L_G(x)$ for each vertex x of G. This label encodes, among other information, the name or the index of x, hence determines it in a unique way. Algorithm $\mathcal{B}$ takes a k-tuple t of bit sequences as input and reports, either that t is not $(L_G(x_1), ..., L_G(x_k))$ for any graph G in $\mathcal{C}$ and any vertices $x_1, ..., x_k$ of such a graph, or determines the validity of $P(x_1, ..., x_k)$ in some graph G belonging to $\mathcal{C}$, for vertices $x_1, ..., x_k$ of this graph such that $t = (L_G(x_1), ..., L_G(x_k))$. This algorithm has no other knowledge about G than the tuple t, and that $G \in \mathcal{C}$. The scheme $(\mathcal{A}, \mathcal{B})$ must be correct in the sense that $P(x_1, ..., x_k)$ must equal the output of $\mathcal{B}$ when given the labels $L_G(x_1), ..., L_G(x_k)$. Clearly, with sufficiently large labels we can encode the entire graph. So the aim is get short labels, ideally of size (measured in number of bits) polylogarithmic in n.

Answering connectivity queries is a fundamental problem in communication networks. In this case, given the labels $L_G(u), L_G(v)$, one should be able to determine quickly whether u, v are in the same connected component of G. Clearly (for undirected graphs) this is easy– each label can store with $O(\log n)$ bits the number of the maximal connected component containing that vertex. Our motivation in this article is to consider so-called *extended connectivity queries* of the following form. The extended connectivity query $Conn(u, v, X)$ asks whether u, v are connected in the graph $G \setminus X$, where X is a set of 'forbidden' vertices (the extension to consider forbidden edges will be easy). The motivation for this is to allow connectivity queries even when the network undergoes failures, and without recomputation of the labels. The set X is given to the query algorithm $\mathcal{B}$ as the set of its labels, and the set F by the labels of the endpoints of its edges, and given these labels, the query algorithm should be able to decide if a path exists from u to v in G, avoiding edges in F and vertices in X.

We now give more technical details before stating the main result and describing the proof method.

Notation. Most of the terminology is as in the book by Diestel [8]. We make precise some notations. All graphs are finite and loop-free. A graph is *simple* if it has no two edges with same ends, and same direction if the graph is directed. We denote by $V(G)$ (resp. $E(G)$) the vertex set (resp. the edge set) of a graph G, and by n its number of vertices.

For $m \in \mathbb{N}$, we let $[m]$ denote the set $\{1, 2, ..., m\}$; we let $[0] = \emptyset$. We denote by $G[U]$ the induced subgraph of G with vertex set $U \subseteq V(G)$ and we let $G \backslash U = G[V(G) - U]$. We denote by $G \backslash v$ the induced sub-graph $G \backslash \{v\}$. $G[F]$ is the sub-graph of G spanned by $F \subseteq E(G)$ hence $E(G[F]) = F$ and $V(G[F])$ is the set of ends of the edges in F. For $Y \subseteq E(G)$ we let $G - Y$ be the subgraph of G with $V(G - Y) = V(G)$ and $E(G - Y) = E(G) - Y$. Hence $G[F] \subseteq G[V(G[F])]$ and $G[E(G) - Y] \subseteq G - Y$; the inclusions may be strict.

The notation $x - y$ (resp. $x \to y$) indicates an undirected edge between x and y (resp. a directed edge from x to y). We denote by $E(x)$ the set of edges incident with x.

A *directed tree* is a tree with edges in any direction. A *rooted tree* is a directed tree with a unique node of indegree 0, called its *root*, from which every node is reachable by a (unique) directed path. A *directed* (resp. *rooted*) *forest* is a disjoint union of directed (resp. rooted) trees. Since we will discuss simultaneously graphs and trees representing their structure, it will be convenient to call *nodes* the vertices of trees.

A partial order $\leq_F$ on the nodes of a rooted forest F is defined as follows: $x \leq_F y$ if and only if every path from a root to x goes through y. Hence the roots are the maximal elements.

A vertex v of G is *separating* if $G' \backslash v$ has at least two connected components where G' is the connected component of v. A connected graph is *biconnected* if it has no separating vertex. A maximal biconnected subgraph (maximal for subgraph inclusion) is a *biconnected component* of the considered graph. We denote by $Bcc(G)$ the set of biconnected components of G. Two vertices u and v are *separated* by $X \subseteq V(G)$ if they are in different connected components of $G \backslash X$.

Let E be a set. A *circular sequence over* E is a non-empty sequence $s = (e_1, \ldots, e_n)$ of pairwise distinct elements of E. The term "circular" refers to equality: we define $(e_1, \ldots, e_n)$ and $(e_i, \ldots, e_n, e_1, \ldots, e_{i-1})$ as equal circular sequences. If $s_1 = (e_1, \ldots, e_p)$ and $s_2 = (f_1, \ldots, f_q)$ are sequences of pairwise distinct elements of E, we will denote by $s_1 \bullet s_2$ the concatenation of s_1 and s_2 and by $s_1 \circ s_2$ the circular sequence, one representation of which is $s_1 \bullet s_2 = (e_1, \ldots, e_p, f_1, \ldots, f_q)$.

If G is a graph, $u, v \in V(G)$, $X \subseteq (V(G) - \{u, v\})$ and $F \subseteq E(G)$, we let

$Conn(u, v, X, F)$ mean:

$Conn(u, v, X, F) \iff$ there exists a path between u and v that *avoids* X *and* F, i.e, a path in the graph $(G - F)\backslash X$.

We call this an *extended connectivity query* (implicitly in the subgraph of G defined by excluding X and F). We write it $Conn(u, v, X)$ if $F = \emptyset$. We call (X, F) the *data* of the query; its *size* is defined as $|X| + |F|$.

Let $P(x_1, \ldots, x_m, X_1, \ldots, X_m)$ be a graph property that depends on vertices $x_1, \ldots, x_m$ and sets of vertices $X_1, \ldots, X_q$. For a mapping $f : \mathbb{N} \to \mathbb{N}$, an $f(n)$-*labeling supporting* P *on a class* $\mathcal{C}$ *of* n*-vertex graphs* is a pair of algorithms $(\mathcal{A}, \mathcal{B})$ such that:

(1) For all $G \in \mathcal{C}$, $\mathcal{A}$ constructs a labeling $J : V(G) \to \{0,1\}^*$ that is injective and is such that $|J(x)| \leq f(n)$ for each $x \in V(G)$.
(2) $\mathcal{B}$ checks whether G satisfies $P(a_1, \ldots, a_p, U_1, \ldots, U_q)$ by using $J(a_1), \ldots, J(a_p)$ and $J(U_1), \ldots, J(U_q)$ where $J(U) = \{J(x) \mid x \in U\}$.

We now state our main theorem.

Theorem 1.1 (Main Theorem) *For every simple undirected planar graph with* n *vertices, we can construct in time* $O(n \cdot \log(n))$ *an* $O(\log(n))$*-labeling supporting extended connectivity queries. Queries are answered in time* $O(m^2)$ *where* $m = |X| + |F|$.

We now sketch the main ideas of the proof. For a plane graph G, we let G^+ be the plane graph obtained by the addition of one new vertex in the middle of each face and of edges between this vertex and those vertices of G incident with that face.

If G is biconnected, the graph G^+ is simple and can be embedded in the plane with integer coordinates and edges represented by straight-line segments by using Schnyder's algorithm [15]; we fix such an embedding.

For $X \subseteq V(G)$, we define its *barrier* $Bar(X)$ as a set of edges of G^+ such that u and v in $V(G) - X$ are separated by X in G if and only if they are separated in $\mathbb{R}^2$ by $Bar(X)$ (Section 2).

If, from labels attached to the vertices of X we can deduce the set of straight-line segments forming $Bar(X)$, and if we also know the coordinates of u and v, we can test whether u and v are separated in $\mathbb{R}^2$ by $Bar(X)$ in time $O(p \cdot \log(p))$ where p is the number of segments forming $Bar(X)$. We show that $p \leq 3 \cdot |X|$ (Section 4).

To form the label $L(x)$ for each vertex x of G, we attach its coordinates in the fixed embedding and those of a bounded number of neighbor vertices of G and of vertices of G^+ representing faces of G. This can be done because every planar graph is the union of three edge disjoint forests (Section 3). However, this proof only works for 3-connected graphs G, or rather for graphs such that every two vertices are incident with a bounded number of faces.

We use an additional treatment, first for biconnected graphs decomposed into 3-connected components (Section 6), and then for connected graphs decomposed into biconnected components, which gives the main theorem (Section 7). These decompositions are expressed as trees. By using a labeling scheme due to Courcelle and Vanicat [4], we can recognize certain cases where u and v are separated by exactly one or two vertices of the given set X. If those separation criteria do not apply, then we are reduced to connectivity queries in 3-connected components, and the geometric method described aboev can be applied.

The proofs are done for the particular case where $F = \emptyset$, i.e., where only vertices are forbidden. However, by subdividing each edge by a single vertex the problem with forbidden edges reduces to the case of only forbidden vertices. This reduction is done at the end of Section 7.

The main theorem extends to queries $Conn(u, v, X, F, H)$ where H is a set of edges inserted between vertices in $V(G) - X$ (we do not require that $((G - F)\backslash X) + H$ is planar, only that G is planar). The query $Conn(u, v, X, F, H)$ means

$Conn(u, v, X, F, H) \iff$ there exists a path between u and v in the graph $((G - F)\backslash X) + H$ defined as $(G - F)\backslash X$ augmented with edges defined by H.

The labeling defined by Theorem 1.1 supports these queries, but the answers take time $O(|H|^2 \cdot \log(m))$ with help of a data structure built for fixed (X, F) in expected time $O(m \cdot \log(m))$ where $m = |X| + |F|$.

The following is a second extension and is left as an open question.

Open Question 1 *Can we label the vertices of a planar graph with labels of size $O(\log(n))$ and for (X, F, H) and $u \in V(G) - X$ in order to decide the number of connected components of $G' = ((G - F)\backslash X) + H$ and the number of vertices of the connected component of G' that contains u ? The answer should be obtained in polynomial-time in $|X| + |F| + |H|$.*

2 Plane Graphs

We review definitions and basic facts about plane graphs. Our main references are the books by Diestel [8] and by Mohar and Thomassen [12].

Definition 2.1 (Embeddings in the Plane) *A* planar embedding *(or from now,* embedding*) of a graph* $G = \langle V, E \rangle$ *is a pair of mappings* $\mathcal{E} = (p, s)$ *such that the mapping* $p : V \to \mathbb{R}^2$ *associates with a vertex* $u \in V$ *the point* $p(u)$ *representing it in the plane, the mapping* $s : E \to \mathcal{P}(\mathbb{R}^2)$ *associates with every edge* e *linking* u *and* v *a closed curve segment with ends* $p(u)$ *and* $p(v)$*, such that for* e *and* $f \neq e$ *in* E*, we have* $x \in s(e) \cap s(f)$ *if and only if* $x = p(u)$ *and* u *is incident with* e *and with* f*. We call it a* straight-line embedding *if each* $s(e)$ *is a straight-line segment.*

A plane graph *is the equivalence class of a planar embedding of a planar graph with respect to homeomorphism. We will write a plane graph as a triple* $\langle V, E, F \rangle$ *where* F *is the set of faces.*

The notion of a plane graph is thus combinatorial. It consists of a graph $G = \langle V, E \rangle$ and for each $u \in V$ of the circular sequence $E^0(u)$ of edges incident with u, for the anti-clockwise orientation of the plane , and a *corner* belonging to the external face (we call (e', u, e) a *corner at* u if e' follows e in $E^0(u)$; each corner belongs to a face; the notion of a corner is relative to a plane graph). We only consider embeddings of graphs in the plane, not in the sphere; for this reason we distinguish the external face with help of some corner. Notice that several plane graphs may have the same underlying planar graph G even if G is 3-connected. See [12] for more details about embeddings of graphs in the plane.

Let C be a cycle in a plane graph G and $\mathcal{E} = (p, s)$ be an embedding of G. Let u be a vertex of G not belonging to C. We say that u is *inside* C if $p(u)$ is in the bounded component of $\mathbb{R}^2 - \mathcal{E}(C)$, where $\mathcal{E}(C)$ denotes the union of the curve segments $s(e)$ for the edges e in C. This property does not depend on $\mathcal{E}$. It will be used for plane graphs, independently of embeddings. We say that two vertices u and v are *separated by* C if exactly one of them is inside C. This means that for every embedding $\mathcal{E}$ of G, vertices u and v are in different connected components of $\mathbb{R}^2 - \mathcal{E}(C)$.

Definition 2.2 (Augmented Graph) *Let* $G = \langle V, E, F \rangle$ *be a connected plane graph. We associate with it a connected planar graph* G^+ *called its* augmented graph. *The graph* G^+ *is* $\langle V \cup F, E \cup E' \rangle$ *where* E' *is a set of edges linking each face* f *to its incident vertices. More precisely we have in* E' *an edge* $u - f$ *for each corner* (e, u, e') *of a face* f*. We may have several edges between* u *and* f *because a face* f *may have several corners at vertex* u *if* u *is a separating vertex. A face of* G *is called a* face-vertex of G^+*.*

For a simple planar graph G with n vertices, the maximum number of faces m is obtained when G is triangulated and $m = 2n - 4$. Hence G^+ has at most $3n - 4$ vertices. Every embedding $\mathcal{E}$ of G can be extended into an embedding $\mathcal{E}^+$ of G^+ in the obvious way: one defines $p(f)$ as any point in the open subset of $\mathbb{R}^2$ corresponding to face f and one draws lines between this point and the vertices adjacent to the vertex f of G^+. This is best explained by an example.

Example 2.3 *Figure 1 shows a plane graph G with vertices t, x, w, u, y, z, v represented by black dots and continuous edges. It also shows the graph G^+. Its face-vertices are small circles. The one marked A represents the external face. There are three parallel edges between A and x, because x is the separating vertex of three biconnected components.*

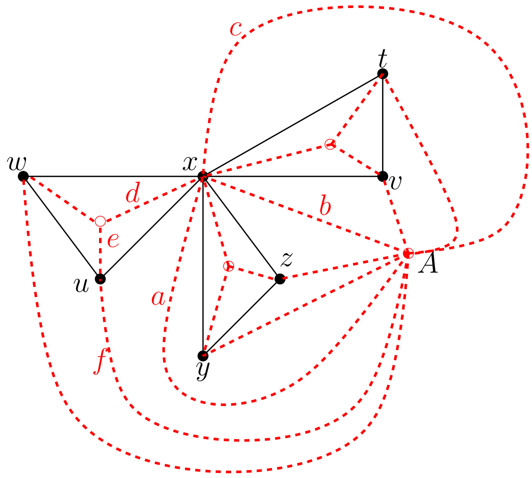

Fig. 1. An augmented graph G^+

In general several non-homeomorphic embeddings $\mathcal{E}^+$ can be associated with $\mathcal{E}$ because the edges incident with the external face of G can be drawn in different ways, even if G is 3-connected. Hence G^+ is a planar graph (and not a plane graph) associated with a plane graph G. The following lemma is straightforward to establish.

Lemma 2.4 *If G is a plane connected graph then the planar graph G^+ is triangulated. It is simple if and only if G is 2-connected.*

Definition 2.5 (The Barrier of a Set of Vertices) *As in Definition 2.2, we let $G^+ = \langle V \cup F, E \cup E' \rangle$ be associated with a plane graph $\langle V, E, F \rangle$. For $X \subseteq V$ we define the* barrier *of X as follows:*

> *$Bar(X)$ is the set of edges of G^+ that link a face $f \in F$ and a vertex $x \in X$ and such that there is in G^+ another edge linking f and some $y \in X$, possibly equal to x.*

If a face f has several corners at a vertex $x \in X$, then all edges of G^+ between f and x are in $Bar(X)$. This can happen if and only if x is a separating vertex.

A vertex of X may not be the end of any edge of $Bar(X)$. See Example 2.7.

If $\mathcal{E}^+ = (p, s)$ is a planar embedding of G^+ we define $Bar(X, \mathcal{E}^+)$ as the union of the segments $s(e)$ for $e \in Bar(X)$. Hence $Bar(X, \mathcal{E}^+)$ is a closed compact subset of $\mathbb{R}^2$. We say that $x, y \in \mathbb{R}^2$ are *separated by* $Bar(X, \mathcal{E}^+)$ if they are in different connected components of $\mathbb{R}^2 - Bar(X, \mathcal{E}^+)$.

Proposition 2.6 *Let G be a connected plane graph and $\mathcal{E}^+$ be a planar embedding of G^+. For every $X \subseteq V(G)$ and $u, v \in V(G) - X$ the vertices u and v are separated by X if and only if the corresponding points of the plane are separated by $Bar(X, \mathcal{E}^+)$.*

We first give examples.

Example 2.7 *We use the graph G of Figure 1. Then $Bar(\{x\}) = \{a, b, c\}$. It separates u and w from y and z and, from t and v. The barrier $Bar(\{y\})$ is empty. We have $Bar(\{u, x\}) = \{a, b, c, d, e, f\}$.*

Example 2.8 *Figure 2 shows the augmented graph H^+ of a graph H. It is simple since H is biconnected. So we can draw it with straight-lines. The barrier of $\{x, y\}$ consists of 6 (thick) dotted edges and separates u from v and w.*

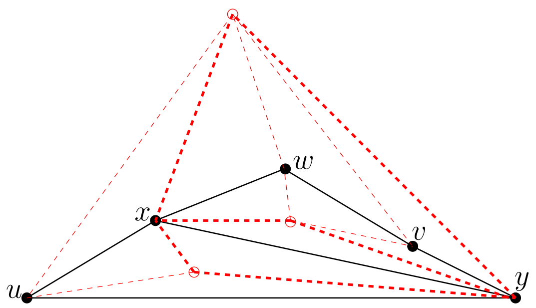

Fig. 2. An augmented graph H^+

Proof of Proposition 2.6. The "Only if direction". Assume u and v connected by a path in $G\backslash X$. They are connected by this path in G^+ and this path has no vertex in any edge of $Bar(X)$. Hence u and v are in the same connected component of $\mathbb{R}^2 - Bar(X, \mathcal{E}^+)$.

The "If direction". Let us assume that u and v are not connected in $G\backslash X$. Since $Y \subseteq X$ implies $Bar(Y) \subseteq Bar(X)$, it is enough to prove the result for a minimal separator X of u and v. Let X be so. The set $E(G)$ can be partitioned into $E(G) = E_u \cup E_v$ such that:

(1) $u \in V(G[E_u])$, $v \in V(G[E_v])$ and $V(G[E_u]) \cap V(G[E_v]) = X$;
(2) $G[E_u]$ and $G[E_v]$ are connected;

(3) The circular sequence of edges incident with each $x \in X$ can be written

$$E^\circ(x) = E_1(x) \circ E_2(x)$$

where $E_1(x)$ is a sequence enumerating the set of edges $E_u \cap E(x)$ and $E_2(x)$ is similar for the set $E_v \cap E(x)$.

Let us split x; that is we add a new vertex x' linked to x by a new edge denoted by e_x and we link to x', as opposed to x, the edges of $E_2(x)$. We make G into a plane graph G' with vertex set $V(G) \cup \{x' \mid x \in X\}$ and with circular sequences $E'^0(w)$ for each $w \in V(G')$ such that:

$$\begin{cases} E'^\circ(x) = E_1(x) \circ (e_x), \\ E'^\circ(x') = E_2(x) \circ (e_x) \end{cases} \qquad \text{for every } x \in X,$$

and

$$E'^\circ(x) = E^\circ(x) \qquad \text{if } x \in V(G) - X.$$

It is clear that G' is a plane graph, and that $E(X) := \{e_x \mid x \in X\}$ is a minimal edge-cut of G'. Hence $E(X)$ is a cycle in the dual plane graph G'^* (see Diestel [8, Proposition 4.6.1]) that separates u and v. (Notice that if $X = \{x_1\}$ then this cycle consists of two parallel edges.)

This cycle can be written as a circular sequence of edges $(e_{x_1}, \ldots, e_{x_p})$ for some enumeration $x_1, \ldots, x_p$ of X. Let $f_1, \ldots, f_p$ be the faces of G' such that in G'^* we have edge $e_i = \{f_i, f_{i+1}\}$ for $1 \leq i < p$ and $e_p = \{f_p, f_1\}$.

We denote by $\overline{f_1}, \ldots, \overline{f_p}$ the faces of G, resulting respectively from $f_1, \ldots, f_p$ by the contraction of edges e_x for all $x \in X$. It is clear that $\overline{f_i}$ is adjacent in G^+ to x_i and x_{i+1} for $i = 1, \ldots, p-1$ and that $\overline{f_p}$ is adjacent to x_p and x_1.

In any embedding $\mathcal{E}^+$ of G^+ the cycle formed by the circular sequence of vertices $(x_1, \overline{f_1}, x_2, \overline{f_2}, x_3, \ldots, x_p, \overline{f_p})$ separates u and v. □

Example 2.9 *A plane graph G is shown on Figure 3. Its vertices u and v are separated by $X = \{x, y, z\}$. Figure 4 shows the result of splitting x, y, z (edges e_x, e_y and e_z are dotted) together with the edges of the cycle $E(X)$ in the dual graph G'^*. The contraction of the dotted edges gives the desired cycle in G^+ (see Figure 5).*

3 Representation of Properties and Functions by Unary Functions

This section introduces the general notion of representation of an n-ary property and of an n-ary partial function by a fixed number of unary functions.

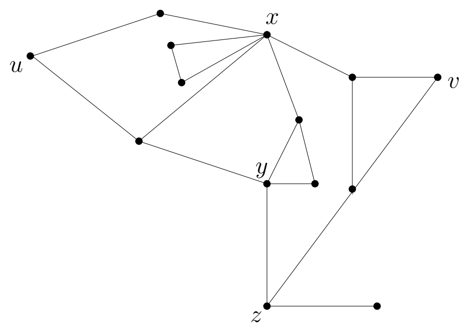

Fig. 3. A plane graph G; $X = \{x, y, z\}$

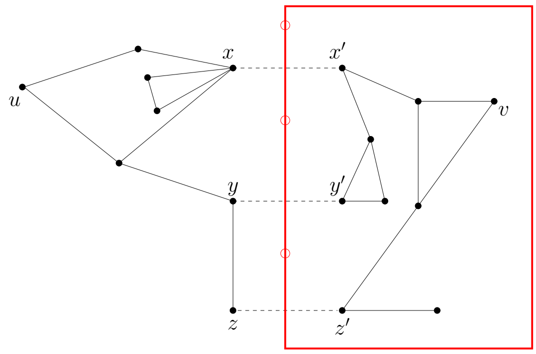

Fig. 4. The plane graph G' and the cycle $E(X)$ in its dual

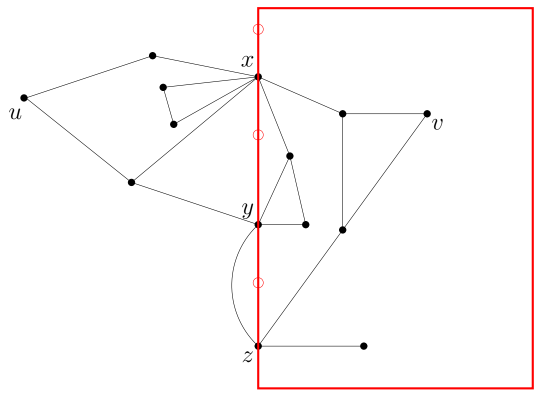

Fig. 5. The plane graph obtained by contracting the edges e_x, $x \in X$

This notion is then used for plane graphs.

Definition 3.1 (Representation by Unary Functions) *If $\mathcal{F}$ is a finite set of unary function symbols and $\mathcal{X}$ is a finite set of variables, we denote by $\mathcal{B}(\mathcal{F}, \mathcal{X})$ the set of quantifier-free formulas that are Boolean combinations of atomic formulas of the forms $x = y$, $x = f(y)$, $f(x) = g(y)$ for $x, y \in \mathcal{X}$ and $f, g \in \mathcal{F}$. (We may have $f = g$.) We do not allow formulas like $x = f(g(y))$, hence $\mathcal{B}(\mathcal{F}, \mathcal{X})$ is not the set of all quantifier-free formulas over $\mathcal{F}$ and $\mathcal{X}$.*

Let V be a set and $\bar{f} : V \to V$ be a total function for each $f \in \mathcal{F}$. We denote by $\overline{\mathcal{F}}$ the family $(\bar{f})_{f \in \mathcal{F}}$ and we let $\mathcal{X} = \{x_1, \ldots, x_m\}$. Every formula

$\varphi \in \mathcal{B}(\mathcal{F}, \mathcal{X})$ defines an m-ary relation $R_\varphi \subseteq V^m$ by:

$$R_\varphi = \{(a_1, \ldots, a_m) \in V^m \;\mid\; \varphi(a_1, \ldots, a_m) \text{ is true}\}.$$

We say that R_φ is represented by the functions of $\overline{\mathcal{F}}$ and the formula *φ.*

*We say that an m-*ary multivalued function, *i.e., a function $g : V^m \to \mathcal{P}(V)$, is* represented by the functions of $\overline{\mathcal{F}}$ and a formula φ *if the $(m+1)$-ary relation $y \in g(x_1, \ldots, x_m)$ is represented by $\overline{\mathcal{F}}$ and φ, where φ is a disjunction of formulas of the form $\psi \wedge (y = f(x_i))$ or $\psi \wedge (y = x_i)$ with $\psi \in \mathcal{B}(\mathcal{F}, \{x_1, \ldots, x_m\})$.*

If an m-ary multivalued function is represented by $\overline{\mathcal{F}}$ and φ where $\mathcal{F}$ is a finite set of functions, then $\varphi \in \mathcal{B}(\mathcal{F}, \{x_1, \ldots, x_m, y\})$. Thus $|g(x_1, \ldots, x_m)| \leq m \cdot (|\mathcal{F}| + 1)$ for all $x_1, \ldots, x_m \in V$. Definition 3.1 also covers the case of partial functions g for which $|g(x_1, \ldots, x_m)| \leq 1$.

We will use properties and functions, associated with graphs of specific classes (e.g. planar graphs) that are representable as defined above, for m and ψ fixed. We will also use the simultaneous representation of finitely many relations $P, Q, \ldots$ and partial functions $g, h, \ldots$ on a set V by a same set $\overline{\mathcal{F}}$ of unary functions and by formulas $\varphi_P, \varphi_Q, \ldots, \varphi_g, \varphi_h, \ldots$.

These definitions will be used as follows. For a class $\mathcal{C}$ of graphs (or of plane graphs) G, we will consider relations $P, Q, \ldots$,and functions $g, h, \ldots$ on $X(G) = V(G)$ (or on $X(G) = V(G) \cup F(G)$), like adjacency, incidence to a same face etc. We say that $P, Q, \ldots, g, h, \ldots$ are *representable by k functions in the graphs of $\mathcal{C}$* if there exist formulas $\varphi_P, \varphi_Q, \ldots, \varphi_g, \varphi_h, \ldots$ of appropriate forms such that for every $G \in \mathcal{C}$, there exists a k-tuple $\mathcal{F}$ of unary total functions on $X(G)$ that represent $P, Q, \ldots, g, h, \ldots$ in G by means, respectively of the formulas $\varphi_P, \varphi_Q, \ldots, \varphi_g, \varphi_h$.

Convention 3.2 *In all the constructions to be done below we will use partial functions $\bar{f} : X(G) \to X(G)$ such that $\bar{f}(x) \neq x$ for every x. We make them total by letting $\bar{f}(x) = x$ instead of "$\bar{f}(x)$ is undefined".*

This convention is useful to avoid the difficulty of defining the semantics of formulas with undefined terms. It makes no more complicated the explicit writing of formulas, as we will see in the next lemma. From now on we will say that a property can be represented by k partial functions (implicitly, such that $\bar{f}(x) \neq x$).

Lemma 3.3 *The adjacency in planar graphs is representable by* 3 *partial functions from vertices to vertices. The adjacency and edge directions in directed planar graphs are representable by* 6 *partial functions from vertices to vertices.*

Proof. We need only consider simple graphs (because we can replace a set of parallel edges by a single edge without changing adjacency).

Let G be a simple planar graph. Its edge set $E(G)$ can be partitioned into three sets E_1, E_2 and E_3 such that $G[E_i]$ is a forest for each i, that we can assume to be rooted (we orient G in an appropriate way). For $x, y \in V(G)$, we let $g_i(x) = y$ if and only if y is the father of x in $G[E_i]$. It is a partial function that we extend into a total one by Convention 3.2. Then x and y are adjacent if and only if:

$$x \neq y \ \wedge \ \left(\bigvee_{1 \leqslant i \leqslant 3} x = g_i(y) \ \vee \ y = g_i(x) \right).$$

The condition $x \neq y$ guarantees that if $x = g_i(y)$ then $g_i(y) \neq y$ hence that x is the father of y in $G[E_i]$ because $\overline{g_i}(y)$ is well-defined for the original partial function $\overline{g_i}$.

For representing edge directions, we replace each function g_i by two functions g_i^+ and g_i^- defined as follows:

$$\begin{array}{ll} g_i^+(x) = y & \text{if and only if } g_i(x) = y \text{ and there is an edge from } x \text{ to } y. \\ g_i^-(x) = y & \text{if and only if } g_i(x) = y \text{ and there is an edge from } y \text{ to } x. \end{array}$$

Notice that we have $g_i^+(x) = g_i^-(x) = y$ if there is a pair of directed opposite edges between x and y. Convention 3.2 is applicable to these functions. □

Because of Convention 3.2, formulas should be written with conditions of the form $g_i(x) \neq x$ conjuncted with each atomic formula containing the term $g_i(x)$. However we will omit such conditions for the purpose of readability. In the formula of Lemma 3.3 the clause $x \neq y$ replaces the condition $g_i(x) \neq x$.

Remark 3.4 *Lemma 3.3 extends easily to graphs of arboricity at most k, i.e., that are the union of k edge disjoint forests as follows. With k functions (resp. $2k$ functions) we represent adjacency (resp. adjacency and edge directions).*

For every pair of distinct vertices (x, y) in a plane graph, we let $Faces(x, y)$ denote the set of faces with which x and y are incident. We say that a plane graph is *m-face-bounded* if $|Faces(x, y)| \leq m$ for every $x, y \in V(G),\ x \neq y$. In particular, a biconnected graph obtained from a simple 3-connected graph by *edge subdivision*, i.e., by the replacement of some edges by paths (such graphs have unique embeddings in the sphere) is 2-face-bounded. In such a graph, two vertices are incident with two distinct faces if and only if they are adjacent or linked by a path with all intermediate vertices of degree two.

For a plane graph G, we let for $x, y \in V(G),\ x \neq y$

$$sf(x,y) \Longleftrightarrow |Faces(x,y)| \geq 1$$

which means that x and y are incident with a same face. This is the case of adjacent vertices. We let for $m \geq 1$:

$$m\text{-}face(x,y) \Longleftrightarrow |Faces(x,y)| \leqslant m$$

An *m-tuple of face selection functions* is an m-tuple $(Select_i)_{i\in[m]}$ of partial functions: $V(G) \times V(G) \to F(G)$ such that for all $x, y \in V(G)$:

$$\begin{aligned}
&Select_i(x,y) \neq Select_j(x,y) && \text{for } i \neq j,\\
&Select_i(x,y) \in Faces(x,y) && \text{for all } i,\\
&Faces(x,y) = \{Select_1(x,y), \ldots, Select_m(x,y)\} && \text{if } |Faces(x,y)| \leq m.
\end{aligned}$$

Note that we do not require $Select_i(x,y) = Select_i(y,x)$ for all i.

For adjacent vertices x and y, we let $left(x,y)$ be the face to the left of the edge $x-y$ (traversed from x to y). Clearly, $left(y,x)$ is the face to the right of $x-y$ and it can be equal to $left(x,y)$ if $x-y$ is an isthmus (or bridge edge). We call $left$ the *left-face function.*

Proposition 3.5 *For every simple connected plane graph, we can represent the adjacency and the left-face function with* 9 *functions on* $V(G) \cup F(G)$*, the adjacency and the same-face property with* 18 *functions. For every* m*, we can define an* m*-tuple of face selection functions and represent it by* $18 + 3m$ *functions including the* 18 *functions used for the same-face property.*

Proof. Let $G = \langle V, E, F \rangle$ be a simple connected plane graph. Let g_1, g_2 and g_3 be the three partial functions constructed by Lemma 3.3. They can represent adjacency. We consider next the left-face function. We let g_i^α be the six partial functions: $V \to F$ such that:

$$\begin{aligned}
g_i^{left}(x) &= left(x, g_i(x)),\\
g_i^{right}(x) &= left(g_i(x), x)
\end{aligned}$$

for $i = 1, 2, 3$, and $\alpha = left, right$ (we let $g_i^\alpha(x)$ be undefined if $g_i(x)$ is). Hence the function $left$ is represented by

$$left(x,y) = f \quad \text{if and only if} \quad \bigvee_{i\in[3]} \left(y = g_i(x) \wedge f = g_i^{left}(x)\right) \vee \bigvee_{i\in[3]} \left(x = g_i(y) \wedge f = g_i^{right}(y)\right)$$

This representation uses 9 functions.

For the same-face property we will use the planar graph $G^+ = \langle V \cup F, E \cup E' \rangle$. Let g_i^+ for $i = 1, 2, 3$ be three partial functions $V \cup F \to V \cup F$ representing the adjacency in G^+ (by Lemma 3.3). The same-face property in G can be expressed as follows for $x, y \in V,\ x \neq y$:

$$\bigvee_{1 \leq i,j \leq 3} g_i^+(x) = g_j^+(y) \in F \tag{1a}$$

$$\vee \bigvee_{1 \leq i,j \leq 3} g_i^+(x) \in F \wedge g_j^+(g_i^+(x)) = y \tag{1b}$$

$$\vee \bigvee_{1 \leq i,j \leq 3} g_i^+(y) \in F \wedge g_j^+(g_i^+(y)) = x \tag{1c}$$

$$\vee\ \exists f \in F \Big(\bigvee_{1 \leq i,j \leq 3} g_i^+(f) = x \wedge g_j^+(f) = y \Big). \tag{1d}$$

In order to handle the condition "$g_i^+(x) \in F$" we use the partial function: $V \to F$ defined by:

$$g_i'(x) = \texttt{ if } g_i^+(x) \in F \texttt{ then } g_i^+(x) \texttt{ else undefined}.$$

In order to handle the conditions "$g_i^+(x) \in F \wedge g_j^+(g_i^+(x)) = y$" we will use the partial functions: $g_{i,j}' : V \to V$ such that:

$$g_{i,j}' = \texttt{if } g_i^+(x) \in F \texttt{ and } g_j^+(g_i^+(x)) \texttt{ is defined then } g_j^+(g_i^+(x))$$
$$\texttt{else undefined}.$$

It remains to eliminate the existential quantification $\exists f \in F(\cdots)$ in formula (1d). We define an auxiliary planar graph H, with $V(H) = V(G)$ and an edge $x - y$ if and only if for some $i, j \in [3]$ and $f \in F$ we have $g_i^+(f) = x$ and $g_j^+(f) = y$. Such an edge can be drawn inside the face f in an embedding $\mathcal{E}$ of G. This shows that H is planar because one adds to each face at most 3 edges. Let h_1, h_2, h_3 be the associated functions by Lemma 3.3. Condition (1d) can thus be replaced by:

$$\bigvee_{1 \leq i \leq 3} h_i(x) = y \vee h_i(y) = x.$$

Hence with the 18 functions $g_i, g_i', g_{i,j}', h_i$ for $i, j \in [3]$ we can represent the adjacency and the same face property.

We now show how to define and represent an m-tuple of face selection functions. We will use cases (1a)-(1d) that characterize the same-face property. We first observe that they are mutually exclusive in the sense that each face of $Faces(x, y)$ is specified by one and only one of them.

It is convenient to fix a linear order on $F(G)$. Let $x, y \in V(G),\ x \neq y$ and $f \in F(G)$. We say that f has (x, y)-type t if $f \in Faces(x, y)$ and we have one of the following conditions:

(a) $f = g'_i(x) = g'_j(y)$ and $t = (a, i, j)$.
(b) $f = g'_i(x),\ y = g'_{i,j}(x)$ and $t = (b, i, j)$.
(c) $f = g'_i(y),\ x = g'_{i,j}(y)$ and $t = (c, i, j)$.
(d) f belongs to $F(x, y)$ defined as the set of faces in $Faces(x, y)$ that are not of the above forms (a), (b) or (c); we fix an enumeration $\{f_1, \dots, f_p, \dots\}$ of the set $F(x, y)$ inherited from the fixed enumeration of $F(G)$, and we let the (x, y)-type of f be $t = (d, j)$.

Note that the (y, x)-type of f is (a, j, i) or (c, i, j) or (b, i, j) or (d, j) if its (x, y)-type is respectively $(a, i, j),\ (b, i, j),\ (c, i, j)$ or (d, j). (We have $F(x, y) = F(y, x)$.)

We define as follows partial unary functions from $V(G) \to F(G)$, for $i \in [3]$ and $j \geq 1$:

$$h_{i,j}(x) = f \quad \text{if } h_i(x) \text{ is defined and } f \text{ is the } j\text{-}th \text{ element of } F(x, h_i(x)).$$

For every $x, y \in V(G)$, $x \neq y$ and $j \geq 1$, there is at most one face f of (x, y)-type (d, j) and it is characterized by the condition:

$$\bigvee_{1 \leq i \leq 3} \Big(\big(f = h_{i,j}(x) \wedge y = h_i(x)\big) \ \vee\ \big(f = h_{i,j}(y) \wedge x = h_i(y)\big) \Big). \qquad (2)$$

Similarly, for each $t \in \{a, b, c\} \times [3] \times [3]$ there is at most one face f of (x, y)-type t and it is characterized by a similar condition. For an example, if $t = (c, 1, 3)$ the corresponding condition is:

$$f = g'_1(y) \ \wedge\ x = g'_{1,3}(y).$$

Let us order types lexicographically. We get thus for each pair (x, y) of distinct vertices a linear order of the set $Faces(x, y)$. We let $Select_i(x, y)$ be the *i-th* element of this set. It is clear that for each $i \leq m$ one can express $f = Select_i(x, y)$ by a disjunction of formulas of the form:

$$f = g(z) \ \wedge\ \psi \qquad (3)$$

where $z \in \{x, y\},\ \mathcal{F}_m = \{g_i, g'_i, g'_{i,j}, h_i, h_{i,\ell} \mid i, j \in [3],\ \ell \in [m]\},\ \psi \in \mathcal{B}(\mathcal{F}_m, \{x, y\})$ and $g \in \mathcal{F}_m$. The set $\mathcal{F}_m$ consists of $18 + 3m$ functions. Hence we have specified an m-tuple of face-selection functions. □

Remark 3.6 *With* $18 + 3(m+1)$ *functions one can represent the property that two vertices* x *and* y *are incident with at most* m *faces. For doing so we use the expression of* $f = Select_{m+1}(x, y)$ *as a disjunction of formulas of the form of (3) (*$f = g(z) \wedge\ \psi$ *for* $z \in \{x, y\},\ g \in \mathcal{F}_{m+1},\ \psi \in \mathcal{B}(\mathcal{F}_{m+1}, \{x, y\})$*)*

and of the form of (2) and, we take the conjunction of the negations of all such formulas.

4 Tools from Computational Geometry

In this section we discuss some tools from computational geometry that can help us to decide whether two vertices of a planar graph G are separated by a subset of $V(G)$.

If $x, y \in \mathbb{R}^2$, we denote by $[x, y] \subseteq \mathbb{R}^2$ the straight-line segment with ends x and y. Two segments $[x, y]$ and $[x', y']$ are *non-crossing* if they only intersect at endpoints, i.e. $[x, y] \cap [x', y'] \subseteq \{x, y\} \cap \{x', y'\}$. A finite set Y of pairwise non-crossing straight-line segments is called a *subdivision of the plane*. The union $\bigcup Y$ of the segments in Y is a closed subset of $\mathbb{R}^2$. We need an algorithm for the following problem:

Input. A subdivision Y of the plane by segments with ends in $\mathbb{N}^2$ and $u, v \in \mathbb{N}^2 - \bigcup Y$.
Output. Are u and v separated by $\bigcup Y$? Equivalently are they in the same connected component of $\mathbb{R}^2 - \bigcup Y$?

The problem is called the *planar point location problem* [1,16].

Theorem 4.1 *[1, Theorem 6.8] Let Y be a subdivision of the plane consisting of m segments. One can construct in expected time $O(m \cdot \log(m))$ a data structure of size $O(m)$ from which one can test in time $O(\log(m))$, in the worst case, whether two elements of $\mathbb{N}^2 - \bigcup Y$ are separated by $\bigcup Y$.*

5 The Labeling of 2-Connected Face-Bounded Plane Graphs

In this section we prove the following particular case of the Main Theorem (Theorem 1.1) stated in the introduction. We denote by $\mathcal{C}_m$ the class of simple m-face bounded 2-connected planar graphs. In particular, a planar graph of degree at most d is d-face bounded.

Theorem 5.1 *Every n-vertex graph in $\mathcal{C}_m$ has an $O(m \cdot \log(n))$-labeling supporting connectivity queries in induced subgraphs defined by excluded vertices. Every subdivided 3-connected planar n-vertex graph has an $O(\log(n))$-labeling supporting such queries. The labeling can be built in time $O(n)$. The answers to queries can be obtained in time $O(|X|^2)$ where X is a set of at least two excluded vertices.*

We will use the following proposition.

Proposition 5.2 *For every simple planar* 2*-connected* n*-vertex graph one can construct in time* $O(n)$ *a corresponding plane graph* G*, the associated planar graph* G^+ *and a straight-line embedding of* G^+ *with positive integer coordinates in* $[3n-6]$.

Proof. The linear-time construction of G is a consequence of the well-known linear-time planarity testing algorithms (see [7]). The construction of G^+ follows then immediately. Since G is assumed 2-connected, the graph G^+ is simple and triangulated. It has at most $3n-4$ vertices. The last assertion follows from Schnyder's algorithm([15]) which defines a straight-line embedding of a simple planar graph H with coordinates in $[|V(H)|-2]$. □

We can now prove Theorem 5.1.

Proof of Theorem 5.1. Let G be a plane graph in $\mathcal{C}_m$ with n vertices. Let $\mathcal{E}$ be a straight-line embedding constructed by Proposition 5.2. Let $C(x) \in \mathbb{N}^2$ be the pair of coordinates of $x \in V(G) \cup F(G)$. Clearly $|C(x)| \leq 2 \cdot \lceil \log(n) \rceil + 2 \cdot \log(3)$.

By means of $p = 18 + 3m$ partial functions: $V(G^+) \to V(G^+)$ (cf. Proposition 3.5; they are extended into total ones by Convention 3.2 and still denoted by $f_1, \ldots, f_p$) we can specify the function $Faces : V(G)^2 \to \mathcal{P}(F(G))$ that associates with $(x, y) \in V(G)^2,\ x \neq y$, the set of faces with which they are both incident. Let us define for $x \in V(G)$:

$$D(x) = (C(x), C(f_1(x)), \ldots, C(f_p(x))) \tag{4}$$

of size $O(m \cdot \log(n))$. For every set $X \subseteq V(G)$ we can define from the family $(D(x))_{x \in X}$ the set of straight-line segments forming the embedding of $Bar(X)$ in $\mathbb{R}^2$ in time $O(|X|^2)$. It consists of the union of the segments from $\mathcal{E}$ corresponding to the edges of G^+ belonging to $Bar(X)$. If $G \in \mathcal{C}_m$ and $X \subseteq V(G)$ then $|Bar(X)| \leq m \cdot (3 \cdot |X| - 6)$.

To see this, consider the sub-graph $G' = G^+[Bar(X)]$. It is a plane bipartite graph with vertex set $X' \cup F$ for some $X' \subseteq X \subseteq V(G)$ and $F \subseteq F(G)$. We recall that a vertex of X may not occur in $Bar(X)$. Let H be the graph with vertex set X' and an edge between x and y whenever there is $f \in F$ such that $((x, f), f, (y, f))$ is a corner of G'. It is clear that H is planar, that $|E(H)| = |E(G')| = |Bar(X)|$ and that there are no more than m parallel edges in H between two vertices. It follows that $|E(H)| \leq m \cdot (3 \cdot |X'| - 6) \leq m \cdot (3 \cdot |X| - 6)$.

The data structure for the planar point location can be built in expected time $O(p \log(p))$ where $p = |Bar(X)|$. From Theorem 4.1 and Proposition 2.6 we can test in time $O(\log(p)) = O(\log(|X|))$ whether two vertices u, v given by $D(u)$ and $D(v)$ (actually $C(u)$ and $C(v)$ suffice) are connected in $G \backslash X$. □

In situations where $|X|$ is bounded by a fixed constant, we get the answer in constant time. In the next section (the most technical one of the article) we extend this result to the class of all biconnected simple planar graphs.

6 The Labeling of 2-Connected Planar Graphs

In this section we prove the main theorem stated in the introduction for the class of 2-connected planar graphs. Technical tools borrowed from Courcelle and Vanicat [4] and Di Battista and Tamassia [7] are presented respectively in Sections 6.1 and 6.2. They make it possible to overcome the following difficulty: since two vertices x and y may be incident with an unbounded number of faces, we may have in $Bar(X)$ an unbounded number of paths $x - f - y$, associated with all faces f incident with x and y. In order to build $Bar(X, \mathcal{E}^+)$, we need the coordinates $C(f)$ of all these faces but they cannot be encoded as lists $(C(f_1), \ldots, C(f_k))$ of bounded length attached to vertices x and y.

We overcome this by replacing each collection of paths $x - f - y$ by only one of them, whenever there are at least 3 faces incident with x and y. This way, we obtain the *reduced barrier* $RBar(X, \mathcal{E}^+) \subseteq Bar(X, \mathcal{E}^+)$. In certain cases it cannot witness that two vertices u and v are separated by X. This case is treated in a different way, using the decomposition of the graph into 3-connected components. The decomposition yields a tree T and the fact that two vertices u and v are separated by $\{x, y\}$ when x and y are attachment vertices of two different 3-connected components where lie u and v, can be checked in this tree by the technique of [4] without using the planar embedding of G.

We first recall the necessary results from [4] and then present the decomposition into 3-connected components with the help of bipolar orientations [9].

6.1 Labeling Schemes for Monadic Second Order Queries on Labeled Trees

Definition 6.1 (Monadic Second Order Queries on Labeled Trees) *Let A be a finite set of labels and $\mathcal{T}(A)$ be the set of finite directed or undirected trees, each node and edge of which has one or more labels from A, or no label* *at all. A tree T in $\mathcal{T}(A)$ will be represented by the following logical structure:*

$$S(T) = \langle N, edg, (nlab_a)_{a\in A}, (elab_a)_{a\in A} \rangle$$

where

(1) N is the set of nodes (we specify it as $N(T)$ if useful),
(2) edg is the binary edge relation (it is symmetric if T is undirected),
(3) $nlab_a(u)$ holds if and only if the node u is labeled by a,
(4) $elab_a(u, v)$ holds if and only if there is an edge from u to v labeled by a.

We will use monadic second order formulas $\varphi(x_1, \ldots, x_m)$ *with* individual free variables $x_1, \ldots, x_m$ *and written with the relation symbols edg, $nlab_a$, $elab_a$ for $a \in A$. We denote by $MS(A, \{x_1, \ldots, x_m\})$ the set of such formulas. They are first order formulas with variables ranging over sets. A formal definition can be found in [4].*

We only give an example significant for our purposes. The formula $\varphi(u, v, w)$ described below expresses in $S(T)$ that the unique path linking u and v goes through w. First we define the formula ψ with free set variable X and individual variables u, v:

$$u \in X \wedge v \in X \wedge \neg\exists Y[u \in Y \wedge v \notin Y \wedge \forall x, y(x \in Y \wedge x \in X \wedge y \in X \Rightarrow y \in Y)]$$

It is satisfied in $S(T)$ by X, u, v if and only if there is a path in T between u and v all nodes of which are in X. Then formula $\varphi(u, v, w)$ can be taken

$$\forall X[\psi(X, u, v) \Rightarrow w \in X]$$

For $\varphi \in MS(A, \{x_1, \ldots, x_m\})$ and $T \in \mathcal{T}(A)$ we let $P_\varphi \subseteq N(T)^m$ be defined as the set of m-tuples $(u_1, \ldots, u_m)$ such that $S(T) \models \varphi(u_1, \ldots, u_m)$. We call P_φ the query defined by φ. *The objective is to label each node u of T by $J(u)$ such that one can* answer the query P_φ, *that is, one can determine whether $P_\varphi(u_1, \ldots, u_m)$ is true or not, from $J(u_1), \ldots, J(u_m)$ only. We will say that this labeling* supports P_φ.

Theorem 6.2 ([4]) *Let A be a finite set of labels and let $\varphi_1, \ldots, \varphi_p$ be formulas in $MS(A, \{x_1, \ldots, x_m\})$. Let $T \in \mathcal{T}(A)$ be a tree with n nodes. One can construct in time $O(n \cdot \log(n))$ an $O(\log(n))$-labeling supporting $P_{\varphi_1}, \ldots, P_{\varphi_p}$.*

We conjecture that the construction of [4] can be done in time $O(n)$.

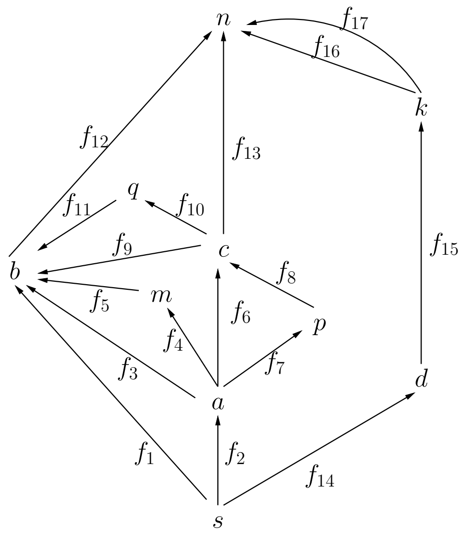

Fig. 6. A bipolar plane graph (cf. Example 6.8)

6.2 Bipolar Plane Graphs

Definition 6.3 (Bipolar Graphs and Bipolar Plane Graphs) *A* bipolar graph *is a directed graph G without circuits having a unique vertex of in-degree* 0*,* $s(G)$ *called its* South pole*, a unique vertex of out-degree* 0*,* $n(G)$ *called its* North pole *such that every* internal vertex*, i.e., every vertex in* $V_{Int}(G) := V(G) - \{s(G), n(G)\}$ *is on a directed path from* $s(G)$ *to* $n(G)$*.*

A directed plane graph G is bipolar *if, as a graph, it is bipolar, and has a planar embedding for which the two poles are incident with the external face.*

Bipolar graphs and bipolar orientations of undirected graphs are studied in [9]. A bipolar graph with adjacent poles is 2-connected. For every edge $x - y$ of a biconnected planar graph, there is an orientation G of this graph making it a bipolar plane graph with $s(G) = x$, $n(G) = y$. Such an orientation can be computed in time $O(n)$ (see [7]).

Lemma 6.4 ([17]) *For every planar embedding of a bipolar plane graph:*

(1) The incoming edges of each vertex x appear consecutively in the circular incidence sequence of x and so do the outgoing edges.
(2) The border of each face f consists of two disjoint directed paths from a vertex $s(f)$, called its South Pole, to a vertex $n(f)$, called its North pole.

If f is the external face, its two paths from $s(f)$ to $n(f)$ are called the *left-border* and the *right-border* of G. In the example of Figure 6, the left-border of G is the path (f_1, f_{12}) and its right-border is (f_{14}, f_{15}, f_{17}).

The circular incidence sequence of x is written $\overrightarrow{in}(x)\circ\overrightarrow{out}(x)$ where $\overrightarrow{in}(x)$ (resp. $\overrightarrow{out}(x)$) is the sequence of incoming (resp. outgoing) edges of x. This expression is possible by Lemma 6.4. We denote $\overrightarrow{out}(s(G))$ by $\overrightarrow{s}(G)$ and $\overrightarrow{in}(n(G))$ by $\overrightarrow{n}(G)$.

Definition 6.5 (Decomposition of Bipolar Plane Graphs) *Let R be a simple bipolar plane graph with m edges denoted $e_1, \ldots, e_m$. Let H, $G_1, \ldots, G_m$ be bipolar plane graphs. We write $H = R(G_1, \ldots, G_m)$ if and only if the following conditions (D1)-(D5) hold:*

(D1) $V(R) \cap V_{Int}(G_i) = \emptyset$ *and* $V_{Int}(G_i) \cap V_{Int}(G_j) = \emptyset$ *for all* $i, j \in [m]$, $i \neq j$.

(D2) e_i *is an edge of R from $s(G_i)$ to $n(G_i)$ for each $i \in [m]$; hence, the vertices of R are the poles of the graphs G_i.*

(D3) $V(H) = V(R) \cup V(G_1) \cup \cdots \cup V(G_m)$.

(D4) $E(H) = E(G_1) \cup \cdots \cup E(G_m)$ *and an edge links the same vertices in H and in the graph G_i to which it belongs. (By condition (D1), $E(G_i) \cap E(G_j) = \emptyset$ for $i \neq j$).*

Informally we could say that H is obtained from R by the replacement of an edge e_i by the graph G_i. Clearly, by these conditions, H is bipolar, $s(H) = s(R)$ and $n(H) = n(R)$. The next condition relates $H, R, G_1, \ldots, G_m$ as plane graphs, and not only as graphs as do Conditions (D1)-(D4).

(D5) *We require the following:*

(a) $\overrightarrow{in_H}(x) = \overrightarrow{in_{G_i}}(x)$ *and* $\overrightarrow{out_H}(x) = \overrightarrow{out_{G_i}}(x)$ *if* $x \in V_{Int}(G_i)$,

(b) $\overrightarrow{in_H}(x)$ *results from the replacement in* $\overrightarrow{in_R}(x)$ *of an incoming edge e from G_i by the sequence $\overrightarrow{n}(G_i)$ and similarly,*

(c) $\overrightarrow{out_H}(x)$ *is defined from* $\overrightarrow{out_R}(x)$ *and the sequences* $\overrightarrow{s}(G_i)$, *for all* $x \in V(R)$.

These conditions mean that planar embeddings are preserved in the replacement in R of e_i by G_i. If $H = R(G_1, \ldots, G_m)$ we say that H decomposes into *$G_1, \ldots, G_m$. We have:*

$$V_{Int}(R(G_1, \ldots, G_m)) = V_{Int}(R) \cup V_{Int}(G_1) \cup \cdots \cup V_{Int}(G_m).$$

The following particular decomposition will be useful. We write $H = G_1 // \cdots // G_m$ if $H = R(G_1, \ldots, G_m)$ and R consists of $m \geq 2$ parallel edges from $s(R)$ to $n(R)$ such that $\overrightarrow{n}(R) = (e_1, e_2, \ldots, e_m)$ and $\overrightarrow{s}(R) = (e_m, \ldots, e_2, e_1)$. We call H the *parallel-composition* of $G_1, \ldots, G_m$ (the operation $//$ is associative but not commutative).

Another particular case is also used in [7] and [2]. We write $H = G_1 \bullet G_2 \bullet \cdots \bullet G_m$ if $H = R(G_1, \ldots, G_m)$ and R consists of a directed path $(e_1, \ldots, e_m)$, $m \geq$

2 from $s(R)$ to $n(R)$. Then H is called the *series-composition* of $G_1, \ldots, G_m$. This operation is also associative and clearly not commutative.

A bipolar plane graph H is called a *//-graph* if it is of the form $G_1 // \cdots // G_m$ for bipolar plane graphs $G_1, \ldots, G_m, m \geq 2$. If it is not a //-graph it is called a *//-atom*.

A *factor* of a bipolar plane graph G is a subgraph H of G that is bipolar and

(1) contains all directed paths in G from $s(H)$ to $n(H)$,
(2) contains all edges of G incident with a vertex of $V_{Int}(H)$.

In such a case there exists a bipolar plane graph R such that G results from the replacement in R of some edge e by H. A factor that is a //-graph is called a *//-factor*.

Proposition 6.6 *(1) A //-graph is of the form $G_1 // \cdots // G_m$ for a unique sequence of //-atoms $G_1, \ldots G_m$.*

(2) A //-atom is an edge or is $R(G_1, \ldots, G_m)$ where $G_1, \ldots, G_m$ are //-factors or edges and R is a //-atom that is not an edge. The graph R and the sequence $(G_1, \ldots, G_m)$ are unique up to a permutation of $E(R)$.

Corollary 6.7 *Every bipolar plane graph has a unique decomposition in terms of the operation of parallel-composition and of substitutions $R(\cdots)$ for //-atoms R that are simple and are not edges.*

We call this decomposition *the decomposition* of the considered plane graph and the corresponding ordered tree its *decomposition tree*. This definition is illustrated by the following example.

Example 6.8 *A bipolar plane graph G with $V(G) = \{s, n, a, b, c, d, k, m, p, q\}$ and $E(G) = \{f_1, \ldots, f_{17}\}$ is shown in Figure 6. The graph G can be expressed by:*

$$G = R_1\Big(f_1, f_2, \big(f_3 // R_3(f_4, f_5)\big), \big(f_6 // R_4(f_7, f_8)\big), \big(f_9 // R_5(f_{10}, f_{11})\big), f_{12}, f_{13}\Big) \\ // R_2\Big(f_{14}, f_{15}, \big(f_{16} // f_{17}\big)\Big)$$

where $R_1, \ldots, R_5$ are shown on Figures 8 and 9. The corresponding tree is in Figure 7.

In decomposition trees (like the one of Figure 7) leaves correspond to the edges of the decomposed graph and on each branch parallel composition operations alternate with substitutions in //-atoms R.

A finer decomposition of bipolar plane graphs is defined in [7]: in this de-

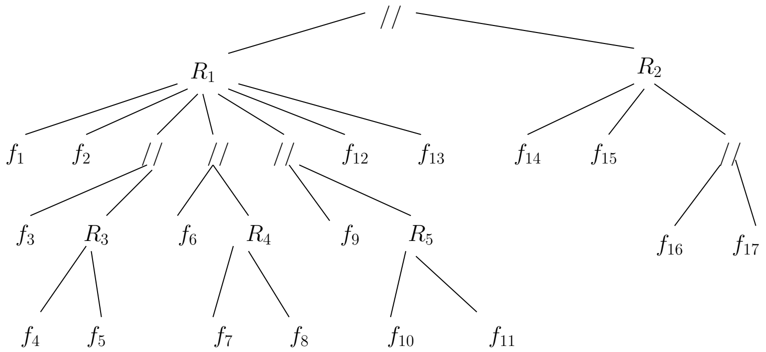

Fig. 7. The decomposition tree of the graph of Figure 6

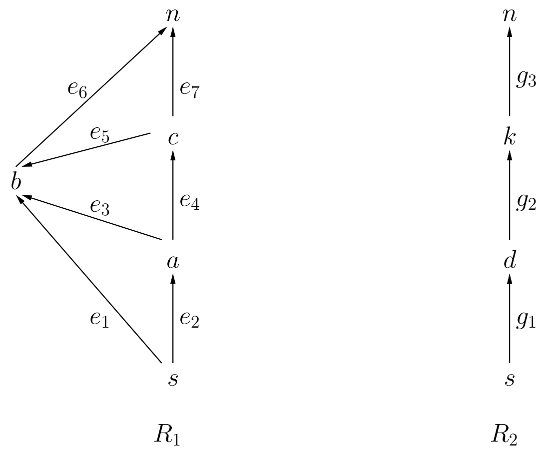

Fig. 8. The graphs R_1 and R_2 (cf Figure 7 and Example 6.8)

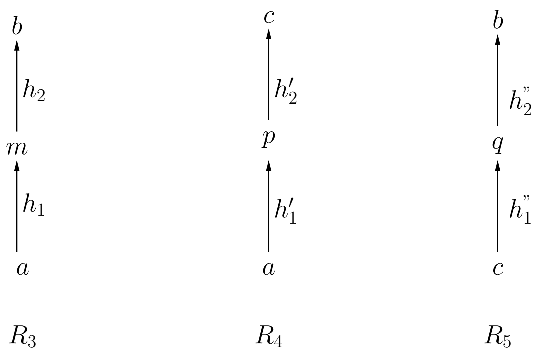

Fig. 9. The graphs R_3, R_4 and R_5 (cf Figure 7 and Example 6.8)

composition each $//$-atom R is expressed in a unique way in terms of series-composition and edge-substitutions in $//$-atoms U such that $U//e$ is 3-connected. The decomposition of [7] can be constructed in linear time. From it one can construct also in linear time the decomposition defined above.

6.3 Polar Pairs

We need some more definitions to discuss the structure of decomposition trees. For a rooted tree T and a node w of T, we denote by $m(w)$ the out-degree of w.

Definition 6.9 (Parallel Nodes and Non-Parallel Nodes) *We let G be a bipolar plane graph with decomposition tree T. For each node w of T, the subtree issued from w, denoted by T/w, defines a subgraph of G denoted by $G(w)$. If w is labeled by $//$, then we call w a* $//$-node *of T, and $G(w)$ is a $//$-factor of G. We denote $s(G(w))$ by $s(w)$ and $n(G(w))$ by $n(w)$.*

If w is a leaf then $G(w)$ is an edge. The set $E(G)$ is in bijection with the set of leaves of T (see Figures 6 and 7). A node w that is neither a leaf nor a $//$-node is called a non-$//$-node. *In this case $G(w)$ is a $//$-atom. If w is a $//$-node with sons $w_1, \ldots, w_{m(w)}$ in this order, then we have $G(w) = G(w_1)// \cdots //G(w_{m(w)})$. The graph $G(w)$ has internal vertices if and only if there is below w a non-$//$-node in the decomposition tree.*

Every non-$//$-node w represents the use of a substitution to the edges of a simple $//$-atom R. Hence w has sons $w_1, \ldots, w_m$ corresponding to the set $E(R)$ enumerated as $e_1, \ldots, e_m$. The nodes $w_1, \ldots, w_m$ are leaves or $//$-nodes.

For a $//$-node w of T, we let $F_j(w)$ for $j = 1, \ldots, m(w) - 1$, be the face whose border cycle consists of the right border of $G(w_j)$ and the left border of $G(w_{j+1})$. These faces are the internal faces of the graph $P = e_1// \cdots //e_{m(w)}$ such that $G_1// \cdots //G_{m(w)} = P(G_1, \ldots, G_{m(w)})$.

Lemma 6.10 *Let $R_1, \ldots, R_p$ be the $//$-atoms associated with the non-$//$-nodes of T enumerated as $w_1, \ldots, w_p$. Then $V_{Int}(G) = \bigcup_{1 \leq i \leq p} V_{Int}(R_i)$. The sets $V_{Int}(R_i)$ are all nonempty.*

Definition 6.11 (Polar Pairs) *Let G be a bipolar plane graph with decomposition T. A* polar pair *is a pair of vertices of the form $(s(w), n(w))$ for some node w of T. It is* $//$-polar *if w is a $//$-node. We say that a polar pair* (x, y) separates u and v *if $\{u, v\} \cap \{x, y\} = \emptyset$ and $(x, y) = (s(w), n(w))$ for some node w such that $u \in V_{Int}(G(w))$ and $v \notin V_{Int}(G(w))$ or vice-versa by exchanging u and v.*

A polar pair $(s(w), n(w))$ is not $//$-polar in the following few cases: w is a leaf and the corresponding edge is simple (it has no parallel edge) or it is $(s(G), n(G))$ and G is a $//$-atom. It follows that if a polar pair separates u and v it is necessarily a $//$-polar pair.

It is clear that if u and v are separated by a polar pair (x, y) then, they are separated by the set $\{x, y\}$. In the example of Figure 6 the pairs $(s, b), (a, c), (c, b), (c, n)$ are polar, the pairs $(c, b), (a, c)$ are //-polar and the pairs $(s, k), (d, n)$ are not polar.

Lemma 6.12 *If in a bipolar plane graph with adjacent poles two vertices are incident with 3 faces, they form a //-polar pair.*

Proof. Let G be a bipolar plane graph with decomposition tree T. Let x, y be two vertices incident with 3 faces f, g and h (and possibly others).

Claim 6.13 *The vertices x and y are on a same border of each face f, g, h.*

Proof of Claim 6.13. Assume that x and y are not on a same border of f. None of them is a pole of f.

Case 1. f is the external face. Consider the cycle $C := x - f - y - g - x$ of G^+ and the cycle C' of G, whence also of G^+, consisting of the border path of f going from $s(f) = s(G)$ to $n(f) = n(G)$ that goes through x and an edge between $s(G)$ and $n(G)$ which cannot be the other border of f since the other border must contain y. They have only x in common and they *cross* at x, that is, in the circular sequence of edges incident with x in G^+, $x - f$ and $x - g$ are separated by edges of C'. This contradicts the planarity of G^+(see e.g. Courcelle [2]). Hence this case cannot happen.

Case 2. f is not the external face. At least one of g and h, say g, is not the external face of G. We consider the cycle C as in Case 1 and the cycle C' of G consisting of the border path of f going from $s(f)$ to $n(f)$ that goes through x, a path from $n(f)$ to $n(G)$, the edge linking $s(G)$ and $n(G)$, and a path from $s(G)$ to $s(f)$. Since y cannot belong to C', this cycle crosses C at x. As in Case 1 we get an impossibility.

Hence x and y are on a same border of each face f, g, h. □

By this claim and without loss of generality we can assume that $y \xrightarrow{*} x$ in G

Claim 6.14 *At least one of f, g or h has (y, x) as pair of poles.*

Proof of Claim 6.14. In the plane graph G^+ we have 3 paths $y - f - x$, $y - g - x$ and $y - h - x$, and without loss of generality we have around x the circular order $x - f, x - g, x - h$. Because of planarity (see [2]) we have necessarily around y the circular order $y - f, y - h, y - g$. Without loss of generality we can assume that g is inside the cycle C'' of G^+ defined as $x - f - y - h - x$.

We will prove that $x = n(g)$. If this is not the case we let x' be the vertex following x on the border of g that contains x. The right-border of f (resp. the left-border of h) contains x. Let z (resp. u) be the vertex that precedes x on this border. Figure 10 shows a part of G^+ around x:

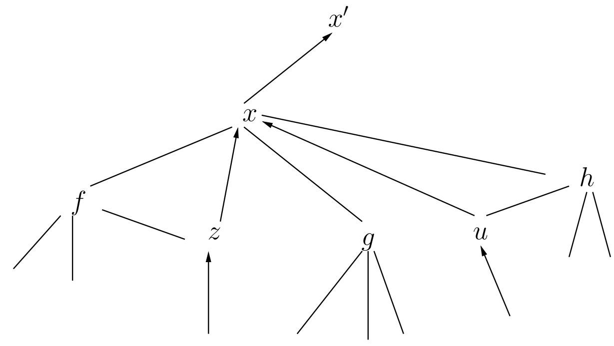

Fig. 10.

We must have around x the following cyclic order of edges: $z \to x$, $u \to x$ and $x \to x'$ by Lemma 6.4 (1). We have $g - x$ between $z \to x$ and $u \to x$. But we also have $x' - g$ in G^+. Hence we have in G^+ two crossing cycles: the cyle $x - g - x' \leftarrow x$ and the cycle of G going through z, x, u and edges from the right-border of f and the left-border of h. We get a contradiction. Hence $x = n(g)$ and similarly $y = s(g)$. □

For completing the proof, we consider the induced subgraph $G[U]$ where U consists of x, y and all vertices that lie inside the cycle $x - f - y - g - x$. It is a factor of G with poles $s(g)$ and $n(g)$. The subgraph $G[U']$ with U' defined similarly from the cycle $x - g - y - h - x$ is also a factor with the same poles. Hence $G[U \cup U'] = G[U]//G[U']$ and is a //-factor of G. Hence (y, x) is a //-polar pair of G. □

We can now state the following.

Lemma 6.15 *Let G be a bipolar plane graph with adjacent poles and let $m \geq 3$. Two vertices x and y are incident with exactly m faces if and only if they are the poles of $G(w)$ for some //-node w such that:*

(1) either w is the root and w has m sons,
(2) or w is not the root and it has $m - 1$ sons.

Therefore we can prove the following key result.

Proposition 6.16 *Every bipolar plane graph with adjacent poles has an* $O(\log(n))$*-labeling supporting the query: "Is* (x, y) *a polar pair separating* u *and* v *?" for all* 4*-tuples of vertices* (x, y, u, v)

The idea is to apply Theorem 6.2 to a tree that encodes enough information about G [5]. We define from the decomposition tree T a tree T^* some nodes of which are (or correspond bijectively to) the vertices of G. Letting $w_1, \ldots, w_p$ be the non-//-nodes of T with associated graphs $R_1, \ldots, R_p$ respectively (cf. Lemma 6.10) we let a vertex x of G belonging to $V_{Int}(R_i)$ be a son of w_i. (The poles of G are represented in a special way as sons of the root.) The major problem is to identify polar pairs. We will use auxiliary unary functions in addition to the information encoded in T^*.

Consider a polar pair (x, y) with $\{x, y\} = \{s(w), n(w)\} \neq \{s(G), n(G)\}$. There are two cases (up to exchanging x and y):

Case 1. $x, y \in V_{Int}(R_i)$ and there is an edge $x \to y$ or $y \to x$ in R_i. Since R_i is planar, we can use Lemma 3.3 and represent such edges (and their directions) by 6 unary functions (g_i for $i = 3, \ldots, 8$). Hence such an edge is represented "at x" or "at y". More precisely if $y \in \{g_4(x), g_6(x), g_8(x)\}$ then there is an edge $x \to y$ represented "at x"; if $y \in \{g_3(x), g_5(x), g_7(x)\}$ there is an edge $y \to x$ also represented "at x". At most 3 such edges are represented at each vertex x or y.

An edge $x \to y$ of R_i is actually a place where a bipolar graph $G(w)$ is substituted (cf. Proposition 6.6 (2)) so that $x = s(w)$ and $y = n(w)$. If this edge is represented by $y = g_i(x)$ for $i \in \{4, 6, 8\}$ then we let w be a son of x in T^* with edge $x \to w$ labeled by i. if it is represented by $x = g_i(y)$ for some $i \in \{3, 5, 7\}$, we let w be a son of y and we label the edge $y \to w$ by i. It follows that for a node w, son of a node x representing a vertex of G, such that the edge $x \to w$ is labeled by $i \in \{3, 4, \ldots, 8\}$ we have that x and $g_i(x)$ are the poles of $G(w)$. Furthermore x is the South pole if i is even and the North pole if i is odd.

Case 2. $x \in V_{Int}(R_i)$, y is a pole of R_i. In this case we let $g_1(x) = y$ if y is a the South pole and $g_2(x) = y$ if y is the North pole. These values of g_1 and g_2 represent respectively edges from $y = s(R_i)$ to x and x to $y = n(R_i)$ of R_i, to which some $G(w)$ is substituted. Similarly as in the previous case we

[5] If we add to the tree T, for an example to the tree on Figure 7, binary relations encoding incidences, for example that edges f_3 and f_4 have same tail, then we get a relational structure T' from which the considered graph can be obtained by a monadic second order (MS) transduction. These 'enriched' trees T' are not images of trees under any MS transduction because otherwise all planar 3-connected graphs would have bounded clique-width, which is not the case. It follows that the results of [4] are not applicable to such a relational structure T' (see [4] for definitions).

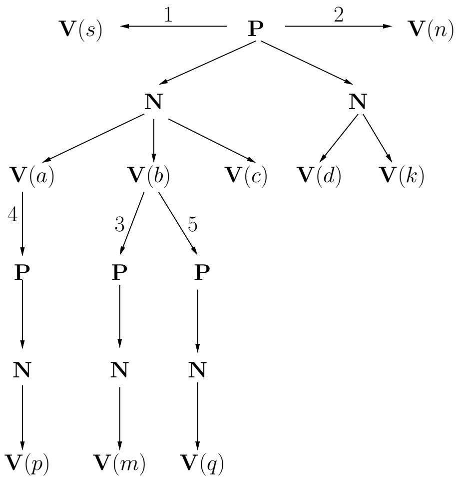

Fig. 11. The tree T^* of the graph of Examples 6.8 and 6.17

let in T^* the node w be a son of x (with edge $x \to w$ labeled by 1 or 2). If $x \to w$ is labeled by 1 or 2 then x is the North pole or the South pole of $G(w)$ respectively.

To conclude this informal presentation, we state that the tree T^* (to be defined formally below) belongs to $\mathcal{T}(A)$ where A is the set of labels $\{\mathbf{P}, \mathbf{N}, \mathbf{V}, 1, \ldots, 8\}$. The nodes labeled $\mathbf{V}$ correspond bijectively to the vertices of G; those labeled by $\mathbf{N}$ are the non-//-nodes of T (the decomposition tree of the considered graph); those labeled by $\mathbf{P}$ are some of the //-nodes of T. The integers $1, \ldots, 8$ are edge labels used as explained above to encode, together with functions $g_1, \ldots, g_8$, the edges of the graphs R_i and, consequently the polar pairs of G.

Example 6.17 *We use the graph of Example 6.8. The table below shows mappings* $g_1, \ldots, g_5$*. The mappings* g_6, g_7, g_8 *are everywhere undefined. The graphs* $R_1, \ldots, R_5$ *are shown on Figures 8 and 9.*

	R_1			R_2		R_3	R_4	R_5
	a	b	c	d	k	m	p	q
g_1	s	s		s		a	a	c
g_2		n	n		n	b	c	b
g_3		a						
g_4	c			k				
g_5		c						

The tree T^* *is shown in Figure 11. For each node labeled by* $\mathbf{V}$*, we indicate between parentheses the corresponding vertex of* G *for helping to understand the construction.*

We now give the precise definition of T^*.

Definition 6.18 (The Labeled Tree T^*) *The* labeled tree T^* *is defined from a bipolar plane graph with adjacent poles G by the following steps.*

Step *1. Construction of the decomposition tree T, by using Corollary 6.7. We let $w_1, \ldots, w_p$ be its non-//-nodes with associated graphs $R_1, \ldots, R_p$. Since G has adjacent poles, the root is a //-node.*

Step *2. Construction of unary partial functions $g_1, \ldots, g_8 : V_{Int}(G) \to V(G)$ such that for each x in $V_{Int}(R_i)$ (we recall that by Lemma 6.10, $(V_{Int}(R_i))_{1 \le i \le p}$ is a partition of $V(G)$):*

$$
\begin{array}{lr}
g_1(x) = s(R_i) & \text{if } s(R_i) \to x \\
g_2(x) = n(R_i) & \text{if } x \to n(R_i) \\
g_j(x) = y & \text{if } y \to x,\ y \in V_{Int}(R_i),\ j \in \{3,5,7\} \\
g_j(x) = y & \text{if } x \to y,\ y \in V_{Int}(R_i),\ j \in \{4,6,8\}
\end{array}
$$

Every edge of R_i is represented by one and only one of these conditions. This construction is possible by Lemma 3.3. We make $g_1, \ldots, g_8$ total by means of Convention 3.2.

Step *3. We construct T^* from T and the functions $g_1, \ldots, g_8$ as follows.*

(T1) Its set of nodes is $N(T^*) = V(G) \cup \{u \in N(T) \mid w \le_T u$ for some non-//-node $w\}$.

(T2) A node of T^* is labeled by **V** *if it belongs to $V(G)$, by* **P** *if it is a //-node of T and by* **N** *if it is a non-//-node.*

(T3) The edges of T^* are defined as follows:

(T3.1) Edges $u \to w$ of T where u is a //-node and w is a non-//-node; they are unlabeled.

(T3.2) If $w = w_i$ is a non-//-node corresponding to R_i, for $1 \le i \le p$, and $w \to w'$ is an edge of T corresponding (cf. Definition 6.9) to an edge $x \to y$ of R_i, we may have the following two cases:

(T3.2.a) either $x = g_j(y)$ for j odd, which implies that $y \in V_{Int}(R_i)$, $x \in V_{Int}(R_i) \cup \{s(R_i)\}$, and we define an unlabeled edge $w \to y$ and an edge $y \to w'$ labeled by j if $w' \in N(T^*)$;

(T3.2.b) or $y = g_j(x)$ for j even, $x \in V_{Int}(R_i)$, $y \in V_{Int}(R_i) \cup \{n(R_i)\}$ and we define an unlabeled edge $w \to x$ and an edge $x \to w'$ labeled by j if $w' \in N(T^*)$.

(T3.3) We also define two edges from the root to nodes $s(G)$ and $n(G)$ respectively labeled by 1 and 2.

Remark 6.19 *From the tree T^* and the associated functions $g_1, \ldots, g_8$, one can "almost reconstruct" G, but not always exactly. For an example, if in the* *graph G on Figure 6, one deletes the edge f_{17}, the tree T^* and the functions g_i do not change. The decomposition tree on Figure 7 is modified. For another example without parallel edges, let $f_1, \dots, f_5$ be edge graphs such that the expression $E = (f_1 \bullet f_2)//(f_3 \bullet f_4)//f_5$ is well-defined. Then the trees T^* associated with E and $(f_1 \bullet f_2)//(f_3 \bullet f_4)$ are the same. Apart from edges between the vertices of a polar pair, the graph G can be reconstructed from T^* and $g_1, \dots, g_8$. The edges which are not encoded by T^* play no role in the determination of the separation of vertices by polar pairs.*

Proof of proposition 6.16. Let G be a bipolar plane graph for which the decomposition tree T, the functions $g_1, \dots, g_8$ and the tree $T^* \in \mathcal{T}(A)$ of Definition 6.18 have been constructed.

For every $x, y, u, w \in N(T^*)$ let $P(u, w, x, y)$ mean:

> x, y, u are labeled by $\mathbf{V}$ (hence are vertices of G), w is labeled by $\mathbf{N}$ or $\mathbf{P}$, $u <_{T^*} w$, $(x, y) = (s(w), n(w))$.

Claim 6.20 *There exists a formula ψ in $MS(A, \{u, v, x, x_1, \dots, x_8, y, y_1, \dots, y_8, z_s, z_n\})$ such that for every $u, w, x, y \in V(G)$ the property $P(u, w, x, y)$ holds if and only if:*

$$S(T^*) \models \psi\Big(u, w, x, g_1(x)/x_1, \dots, g_8(x)/x_8, y, g_1(y)/y_1, \dots, g_8(y)/y_8, s(G)/z_s, n(G)/z_n\Big).$$

The notation $g_i(x)/x_i$ means that the term $g_i(x)$ is substituted to x_i (and similarly for $g_i(y)/y_i$), and $s(G)/z_s$ means that z_s is given the value $s(G)$ (and similarly for $n(G)/z_n$).

Proof of Claim 6.20. The only difficulty is to express the condition $(x, y) = (s(w), n(w))$. We distinguish several cases.

Case 1. w is the root or w is a son of the root which is labeled by $\mathbf{P}$ (hence w is labeled by $\mathbf{N}$). In this case the condition $(x, y) = (s(w), n(w)) = (s(G), n(G))$ where $P(u, w, x, y)$ is expressed by the formula $x = z_s \wedge y = z_n$.

Case 2. w is not the root and is labeled by $\mathbf{P}$; hence it is not a son of the root (by the way T^* is constructed). Its father w' is labeled by $\mathbf{V}$, hence is a vertex of G and w' is one of the two poles of $G(w)$. Let $j \in [8]$ be the label of the edge $w' \to w$. Then the other pole of $G(w)$ is $g_j(w')$. It follows that the condition $(x, y) = (s(w), n(w))$ is equivalent to $\theta[g_1(x)/x_1, \dots, g_8(x)/x_8, g_1(y)/y_1, \dots, g_8(y)/y_8]$

where $\theta(w, x, x_1, \ldots, x_8, y, y_1, \ldots, y_8)$ expresses:

$$\left(\text{“}x \text{ is the father of } w\text{”} \wedge \bigvee_{j=2,4,6,8} y = x_j\right) \vee \left(\text{“}y \text{ is the father of } w\text{”} \wedge \bigvee_{j=1,3,5,7} x = y_j\right).$$

Case 3. w is not the root, is labeled by **N** and its father w'' is labeled by **P** and is not the root otherwise Case 1 applies. The father w' of w'' is labeled by **V**. We have $(s(w), n(w)) = (s(w''), n(w''))$ and $w' \in \{s(w), n(w)\}$ as in Case 2. The construction is the same as in Case 2 with θ' instead of θ, obtained by replacing “x is the father of w” by “x is the grand-father of w” and similarly for y.

Then the desired formula ψ can be written as $\psi_1 \vee \psi_2 \vee \psi_3$, where ψ_1, ψ_2 and ψ_3 express Cases 1,2 and 3 respectively.

ψ_1 is $\Big($“w is the root” $\vee$ “the father of w is the root labeled by **P**” $\wedge\, (x = z_s \wedge y = z_n)\Big)$.

ψ_2 is $\Big($“w is not the root” $\wedge$ “w is labeled by **P**” $\wedge\, \theta(w, x, x_1, \ldots, y_8)\Big)$.

ψ_3 is $\Big($“w is not the root”$\wedge$“w is labeled by **N**”$\wedge$“the father of w is not the root”$\wedge$ $\theta'(w, x, x_1, \ldots, y_8)\Big)$.

This finishes the proof of the claim. □

We now complete the proof of Proposition 6.16. The condition $Q(u, v, x, y)$ defined as “(x, y) is a polar pair separating u and v” can be expressed as follows from Definition 6.11:

> There exists w such that either $P(u, w, x, y)$ holds and v is labeled by **V** and $v \not<_T w$ or $P(v, w, x, y)$ holds and u is labeled by **V** and $u \not<_T w$.

It follows from Claim 6.20 that one can build a formula φ in $MS(A, \{u, v, x, x_1, \ldots, x_8, y, y_1, \ldots, y_8, z_s, z_n\})$ such that $Q(u, v, x, y)$ holds if and only if

$$S(T^*) \models \varphi\Big(u, v, x, g_1(x)/x_1, \ldots, g_8(x)/x_8, y, g_1(y)/y_1, \ldots, g_8(y)/y_8, s(G)/z_s, n(G)/z_n\Big). \quad (5)$$

We now apply Theorem 6.2 to T^* and φ. This theorem gives an $O(\log(n))$-labeling $L(w)$ of the nodes w of T^*, hence in particular of the vertices of G.

The desired labeling $K(x)$ of the vertices of G is then defined as

$$K(x) = (L(x), L(g_1(x)), \ldots, L(g_8(x)), L(s(G)), L(n(G)))\,.$$

We have $|K(x)| = O(\log(n))$ and by Equivalence (5), we can determine if (x, y) is a polar pair separating u and v by using $K(u), K(v), K(x)$ and $K(y)$. □

6.4 Reduced Barriers

Let G be a bipolar plane graph with decomposition tree T. For every //-polar pair (x, y) we let $\mathit{Select}(x, y)$ be some face incident with x and y. We can make this definition deterministic by letting $\mathit{Select}(x, y) = F_1(w)$ (cf. the end of Definition 6.9) where w is the //-node such that $(x, y) = (s(w), n(w))$, but any other face, say $F_j(w)$ for any j with $j \leq m(w) - 1$ would work.

Definition 6.21 (Reduced Barriers for Bipolar Plane Graphs) *Let G be a bipolar plane graph with adjacent poles and augmented graph G^+. For $x, y \in V(G)$, $x \neq y$ we define $RBar(\{x, y\})$ as the following set of edges of G^+:*

(R1) if x and y are incident with at most 2 faces then $RBar(\{x, y\}) = Bar(\{x, y\})$;
(R2) otherwise by Lemma 6.12, x and y form a //-polar pair, say (x, y), and we let $RBar(\{x, y\})$ consist of the two edges $x - f$ and $y - f$ where $f = \mathit{Select}(x, y)$.

For $X \subseteq V(G)$ we let $RBar(X) := \bigcup\{RBar(\{x, y\}) \mid x, y \in X\}$ and we call it the reduced barrier of X.

If $\mathcal{E}^+$ is an embedding of G^+, then $RBar(X, \mathcal{E}^+)$ denotes the union of the segments representing the edges in $RBar(X)$. The use of reduced barriers is based on the following proposition which extends Proposition 2.6.

Proposition 6.22 *Let G be a bipolar plane graph with adjacent poles and let $\mathcal{E}^+$ be an embedding of G^+. Let $X \subseteq V(G)$ and $u, v \in V(G) - X$. Then u and v are separated by X if and only if either:*

(a) u and v are separated by a polar pair belonging to $X \times X$ or:
(b) u and v are separated in the plane by $RBar(X, \mathcal{E}^+)$.

Proof. Let G, X, u, v be as in the statement. If (a) or (b) holds then u and v are separated by X (for the second case, we observe that $RBar(X, \mathcal{E}^+) \subseteq Bar(X, \mathcal{E}^+)$ and we use Proposition 2.6).

Let us conversely assume that u and v are separated by X, but (a) does not hold. By Proposition 2.6, they are separated in the plane by $Bar(X, \mathcal{E}^+)$. As in the proof of Proposition 2.6 we need only prove the result for a minimal separator $Y \subseteq X$ of u and v, because if u and v are separated by $RBar(Y, \mathcal{E}^+)$ they are also by $RBar(X, \mathcal{E}^+)$. Hence we assume that $X = \{x_1, \ldots, x_m\}$ is a minimal separator of u and v in G. We first assume that $m \geq 3$. Then, $Bar(X)$ has the structure shown on Figure 12 where, for each $i \in [m]$, $\{f_{i,1}, \ldots, f_{i,p_i}\}$ is the set of faces incident with x_i and x_{i+1} (letting x_{m+1} denote also x_1).

Then $RBar(X)$ is obtained from $Bar(X)$ by removing for each i such that $p_i \geq 3$ all vertices $f_{i,j}$ (and the incident edges) but one, so that $RBar(X)$ contains a cycle going through $x_1, \ldots, x_m$. If u, v are separated by $Bar(X, \mathcal{E}^+)$ and not by $RBar(X, \mathcal{E}^+)$ this means that one and only one of them is inside a cycle $x_i - f_{i,j} - x_{i+1} - f_{i,j+1} - x_i$ of $Bar(X)$ such that $f_{i,j}$ or $f_{i,j+1}$ (or both) has been removed. This implies that $p_i \geq 3$ hence that x_i and x_{i+1} form a //-polar pair (by Lemma 6.12). Furthermore the set of vertices that are inside this cycle are the internal vertices of $G(w_j)$ where w_j is the *j-th* son of the //-node w with poles x_i and x_{i+1}. Hence u and v are separated by a polar pair with components x_i and x_{i+1} in X, hence (a) holds, but we assumed the contrary. Hence (b) must hold. This completes the proof for the case $m \geq 3$.

If $m = 2$ and $p_1 = p_2 = 1$ then $Bar(X) = RBar(X)$ hence (b) holds. If $p_1 + p_2 \geq 3$ then, by Lemma 6.12, x_1 and x_2 form a polar pair. As for the case $m \geq 3$ we get that u and v are separated by $RBar(X, \mathcal{E}^+)$ otherwise (a) holds.

We cannot have $m = 1$ because the graph is assumed 2-connected. □

Example 6.23 *For clarity on Figure 13 we number faces from* 1 *to* 8 *but we do not show the edges of* G^+ *incident with the face-vertices* $1, \ldots, 8$. *The set* $Bar(\{x, y\})$ *contains the* 4 *paths* $x - i - y$ *for* $i = 2, 6, 7, 8$. *Note that* (x, y) *is a //-polar pair. The reduced barrier* $RBar(\{x, y\})$ *contains only one of them, say* $x - 2 - y$. *However for any two vertices* u *and* v *separated by* $\{x, y\}$, *Condition (R1) is applicable. The set* $RBar(\{x, y, c\})$ *contains then* $x - 2 - y$, $x - 3 - c$, $x - 4 - c$, $c - 5 - y$. *This reduced barrier separates* b *and* d. *The edges* $x - 2$ *and* $2 - y$ *are useful for that: without them* b *and* d *are not separated.* $RBar(\{a, x\}) = Bar(\{a, x\}) = \{x - 2, 2 - a\}$ *and the graph* $G \backslash \{a, x\}$ *is connected. Note that* a *and* x *do not form a polar pair.*

6.5 The Main Theorem for 2-Connected Planar Graphs

After proving a last technical lemma, we will establish the following theorem.

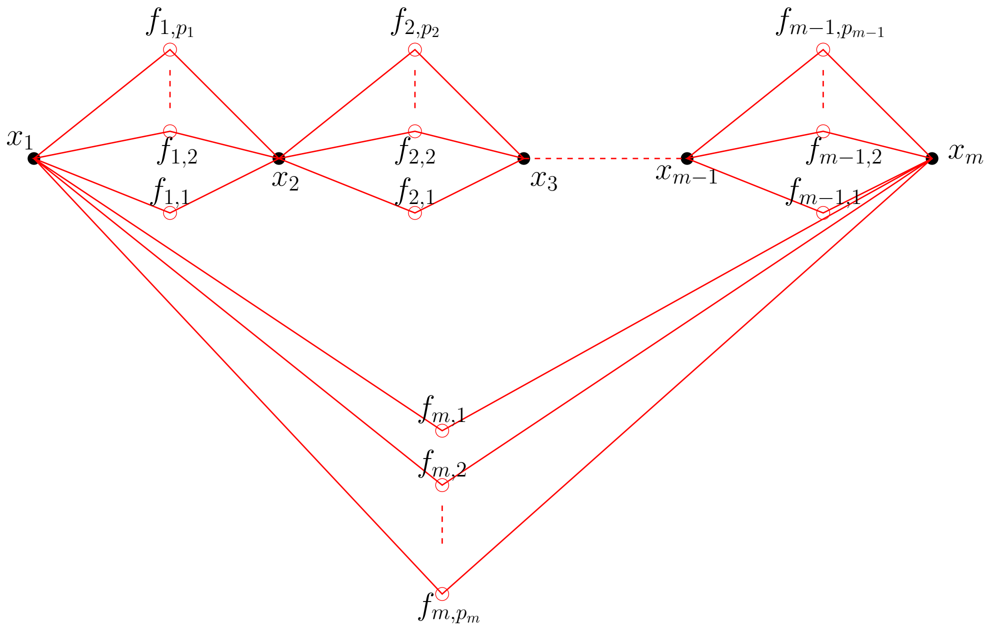

Fig. 12. A barrier (cf the proof of Proposition 6.22)

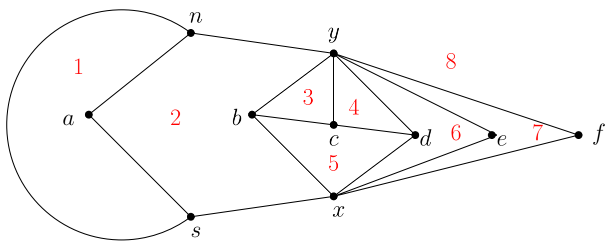

Fig. 13. The graph of Example 6.23

Theorem 6.24 *For every* 2*-connected planar graph we can construct an* $O(\log(n))$*-labeling supporting extended connectivity queries with forbidden vertices* X*. The labels can be constructed in time* $O(n)$ *and queries answered in time* $O(|X|^2)$*.*

We first state and prove a lemma, akin to that in Section 3.

Lemma 6.25 *In every bipolar plane graph* G *one can represent with* 12 *functions on* $V(G) \cup F(G)$ *the property pp, defined as:*

$$pp(x,y) \iff (x,y) \text{ is a } //\text{-polar pair}$$

and any fixed Select function as defined at the beginning of Section 6.4.

Proof. The proof is a variant of that of Proposition 3.5. We let H be the simple directed graph with $V(H) = V(G)$ and an edge $x \to y$ if and only if (x,y) is a $//$-polar pair. It is planar because these edges can be inserted without crossings in a planar embedding of G. With 6 functions, one can represent adjacency and edge directions of a planar graph by Lemma 3.3. Hence there exist functions $g_i^+, g_i^- : V(H) \to V(H)$ for $i \in [3]$ such that:

$$\begin{aligned} g_i^+(x) = y \quad & \text{implies} \quad x \to y, \\ g_i^-(x) = y \quad & \text{implies} \quad y \to x. \end{aligned}$$

Each edge is represented by a unique such clause. Hence with 6 partial functions, we can represent the property *pp*.

We now define 6 partial functions h_i^α for $i \in [3], \alpha \in \{+, -\}$ as follows:

$$\begin{aligned} h_i^+(x) &= \mathit{Select}(x, g_i^+(x)), \\ h_i^-(x) &= \mathit{Select}(g_i^-(x), x). \end{aligned}$$

By using also the 6 functions g_i^+, g_i^- we can represent the *Select* function with 12 functions. □

We can now prove Theorem 6.24.

Proof of Theorem 6.24. We are given a 2-connected planar graph with n vertices. In time $O(n)$ we can make it into a bipolar plane graph with adjacent poles, we can construct its decomposition tree T, the functions $g_1, \ldots, g_8$ and the labeling $(K(x))_{x \in V(G)}$ of Proposition 6.16, such that $|K(x)| = O(\log(n))$ relative to the tree T^* of Definition 6.18. We can also construct a straight-line embedding of the plane graph G^+ with coordinates in $[3n-6]^2$ by Proposition 5.2. We let $C(x)$ be the pair of coordinates of $x \in V(G^+)$. In order to be able to build $RBar(X)$ from $O(\log(n))$ we attach bounded information to the elements of X. We will use:

- 21 functions for representing the property "x and y are incident with at most 2 faces" and for specifying these faces (Proposition 3.5)
- 12 functions for representing the property $pp(x, y)$ and defining $\mathit{Select}(x, y)$ by Lemma 6.25

Hence we will use 33 functions $f_i : V(G) \to V(G) \cup F(G), i \in [33]$. We let then

$$D(x) = (C(x), C(f_1(x)), \ldots, C(f_{33}(x)), C(s(G)), C(n(G)))$$

for each $x \in V(G)$, and $J(x) = (K(x), D(x))$. It is clear that $|J(x)| = O(\log(n))$ (in particular $|D(x)| \leq 72(\log(n) + \log(3))$ and we claim that J supports connectivity queries in subgraphs defined by excluded vertices. The checking procedure is the following for given $u, v \in V(G)$ and $X \subseteq V(G)$.

Step 1. By using the K-parts of the labels attached to u, v and to the vertices in X, one can test by trying every two vertices in X whether u and v are separated by a polar pair in $X \times X$. If this is the case one can report that u and v are separated by X and stop. Otherwise one performs Step 2.

Step 2. By using the D-parts of the labels, one can determine, for every two vertices x, y in X the coordinates of the end vertices of the edges forming $RBar(\{x,y\}, \mathcal{E}^+)$, which are straight-line segments. One can test if u and v are separated by $RBar(X, \mathcal{E}^+)$ (cf. Section 4) and by Proposition 6.22, this gives the final answer.

The time taken to decompose G and to construct T^* is $O(n)$. The time taken to build the labels $D(x)$ is $O(n)$. The time taken to build the labels $K(x)$ is $O(n \cdot \log(n))$. This bound depends on the results of [4] and may perhaps be improved to $O(n)$. Hence the labeling $J(x)$ can be constructed in time $O(n \log(n))$.

The answers to Step 1 can be obtained in time $O\left(|X|^2\right)$. The answers to Step 2 can be obtained in time $O\left(|X| \cdot \log(|X|)\right)$. □

7 Connectivity Queries on 2-Connected Components

We prove the Main Theorem by using as in Section 6 some results of [4] applied to the classical decomposition of a graph into a tree of biconnected components.

Let G be a connected graph. We denote by $Bcc(G)$ the set of its biconnected components. We denote by $B(G)$ the bipartite tree with set of nodes $V(G) \cup W(G)$ where $W(G) \cap V(G) = \emptyset$ and $W(G)$ is in bijection with $Bcc(G)$ by $bcc : W(G) \to Bcc(G)$, and with edges $v - w$ whenever $w \in W(G)$ and $v \in V(bcc(w))$. A vertex of G has degree at least 2 in $B(G)$ if and only if it is separating in G.

The biconnected components containing at least 2 vertices of X are therefore the ones we must deal with.

Definition 7.1 (Problematic Biconnected Components) *Let $X \subseteq V(G)$ and $u, v \in V(G) - X$. We say that a biconnected component of G is* problematic *for (u, v, X) if it (or rather the node of $W(G)$ representing it) is on the unique path $p(u, v)$ in $B(G)$ from u to v and contains at least 2 vertices of X.*

Let us assume that no vertex on $p(u, v)$ belongs to X. Let $C_1, \ldots, C_m$ be the sequence of problematic components enumerated in their order of occurrence on $p(u, v)$. Let $x_1, x_2, \ldots, x_{m-1}$ be vertices such that x_i is between C_i and C_{i+1} on $p(u, v)$. Let $x_0 = u$ and $x_m = v$. The following is clear from the definition:

Fact 7.2 *The vertices u and v are separated by X if and only if either:*

(a) the path $p(u, v)$ goes through a vertex in X,
(b) or (a) does not hold and for some $i = 0, \ldots, m-1$, the vertices x_i and x_{i+1} are separated by $X \cap V(C_i)$ in G.

We will use Theorem 6.2 in order to build an $O(\log(n))$-labeling with which one can check the conditions of Fact 7.2.

We choose a vertex r of G to be the root of $B(G)$ that belongs to a unique biconnected component. From this choice, $B(G)$ is directed, rooted with partial order $\leq_{B(G)}$ and r is the greatest element (see Introduction). For each $C \in Bcc(G)$ the set $V(C)$ has a $\leq_{B(G)}$-greatest element called the *leader of* C. Each vertex v belongs to a unique $\leq_{B(G)}$-maximal biconnected component. We call it its *mother* if $v \neq r$. The root has no mother.

Our next aim is to prove the following proposition, stated with the notation of Definition 7.1 and Fact 7.2.

Proposition 7.3 *Let G be a connected graph with n vertices. One can build an $O(\log(n))$-labeling $(M(x))_{x \in V(G)}$ such that:*

(1) one can determine from the labels of any $u, v \in V(G)$ and of the vertices in any set $X \subseteq V(G) - \{u, v\}$ whether $p(u, v)$ goes through X and, if it does not,
(2) one can determine the sets $X \cap V(C_i)$ for $i = 1, \ldots, m$ and vertices $x_1, \ldots, x_{m-1}$ that are leaders of some of the problematic components $C_1, \ldots, C_m$ and such that:

$$Conn(u, v, X) \Longleftrightarrow \bigwedge_{0 \leq i \leq m-1} Conn(x_i, x_{i+1}, X \cap V(C_{i+1})) \qquad (6)$$

Proof. The tree $B(G)$ is handled as the logical structure $\langle V(G) \cup W(G), member, root \rangle$ where $member(v, w)$ holds if and only if $v \in V(bcc(w))$, and $root(v)$ holds if and only if v is the root.

Among the elements of $V(G) \cup W(G)$ the vertices of G are those, say x, satisfying $\exists w.member(x, w)$. The order $\leq_{B(G)}$ is definable by a monadic second order (MS) formula [4].

For $x \in V(G) \cup W(G)$, $x \neq r$ the unique smallest element y such that $x <_{B(G)} y$ represents the mother of x if $x \in V(G)$ and is denoted by $mother(x)$; it is the leader of $bcc(x)$ if $x \in W(G)$, and is denoted by $leader(x)$. (The root is the leader of a unique biconnected component.) These two functions are thus definable by MS formulas. We consider the following properties of the nodes of $B(G)$:

$$P_1(u,v,x) \iff \text{u,v,x are pairwise distinct vertices and x is on the path $p(u,v)$ linking u to v.}$$

$$P_2'(u,v,w,x,y) \iff \text{u,v,x,y are pairwise distinct vertices, w belongs to $W(G)$ and lies on the path $p(u,v)$, and furthermore $x,y \in V(bcc(w))$.}$$

$$P_2(u,v,x,y) \iff P_2'(u,v,w,x,y) \text{ holds for some } w.$$

We use Theorem 6.2 to construct an $O(\log(n))$-labeling M_0 for checking the properties $x \leq y$, $member(x,y)$, P_1 and P_2. This labeling defines a label $M_0(x)$ for each $x \in V(G) \cup W(G)$. For $x \in V(G)$ we define:

$$M(x) = \Big(M_0(x), M_0(mother(x)), leader(mother(x))\Big). \tag{7}$$

(If x is the root we mark the last two components as "undefined").

By using $M_0(u), M_0(v)$, and $M_0(x)$ for each $x \in X$ in turn, we can check if $P_1(u,v,x)$ holds for some $x \in X$, hence whether $p(u,v)$ goes through some vertex in X. If this is the case we can report that u and v are separated by X. This test takes time $O(|X|)$.

Otherwise we consider the path $p(u,v)$. It can be of 3 possible types depending on how its nodes are related under $\leq_{B(G)}$ where $C_1,\dots,C_m$ are the problematic biconnected components relative to u,v and X; we denote $<_{B(G)}$ by $<$.

Case 1. $u < C_1 < C_2 < \cdots < C_m < v$ or the same by changing $<$ into $>$,

Case 2. $u < C_1 < C_2 < \cdots < C_{p-1} < C_p > C_{p+1} \cdots > C_m > v$,

Case 3. $u < C_1 < C_2 < \cdots < C_p < w > C_{p+1} \cdots > v$ where w is either a vertex or a biconnected component that is not problematic. In all cases we let $x_0 = u, x_m = v$.

In the first case we let x_i be the leader of C_i for $i = 1,\dots,m-1$. In the variant of the first case where $u > v$, we let x_i be the leader of C_{i+1} for $i = 1,\dots,m-1$. In the second case, we do the same for $i = 1,\dots,p-1$ and we let x_i be the leader of C_{i+1} for $i = p,\dots,m-1$. In the third case we do as in the first for $i = 1,\dots,p$ and we let x_i be the leader of C_{i+1} for $i = p+1,\dots,m-1$.

By using $M_0(u), M_0(v)$ and $M_0(x)$ for all $x \in X$, we can determine those pairs of elements (x,y) in X^2 such that $P_2(u,v,x,y)$ holds, hence such that x and y belong to a problematic component $bcc(w)$, determined as follows (we let r

be the root of $B(G)$):

$$\texttt{if } y = r \texttt{ or if } mother(x) \leq mother(y) \texttt{ then } w = mother(x),$$
$$\texttt{if } x = r \texttt{ or if } mother(y) \leq mother(x) \texttt{ then } w = mother(y).$$

as one checks easily. (We may have $mother(x) < mother(y)$ if y is the leader of $mother(x)$ and $p(u, v)$ goes through $mother(x)$ but not through y. We recall that $mother(r)$ is undefined.) Since the label $M(x)$ contains $M_0(mother(x))$ we can obtain the set:

$$P = \{M_0(w) \mid bcc(w) \text{ is a problematic component}\}.$$

Since M_0 makes it possible to know from $M_0(w)$ and $M_0(w')$ if $w < w'$, one can order P as $\{M_0(bcc^{-1}(C_1)), \ldots, M_0(bcc^{-1}(C_p))\}$, and one can determine which of the Cases 1,2 or 3 holds. Note that in Case 3, we cannot determine (and we need not) determine the "central element" w.

Since each component C_i is problematic we know at least one x in $X \cap V(C_i)$ such that $C_i = bcc(mother(x))$. Since $M(x)$ contains $leader(mother(x))$ for each $x \in X$ we get the leaders of the problematic components, whence the desired list $x_1, \ldots, x_{m-1}$ (we also have $u = x_0$ and $v = x_m$).

If $C_i = bcc(mother(x))$ then $X \cap V(C_i)$ is the set of elements y of X such that $member(y, mother(x))$. From $M_0(y)$ and $M_0(mother(x))$ which are available from $M(y)$ and $M(x)$ for all $x, y \in X$, we can determine when $member(y, mother(x))$ does hold. Hence we have for each i, the indices of the vertices in $X \cap V(C_i)$. □

This proposition shows that the connectivity query in a connected, non necessarily planar, graph reduces to connectivity queries in this graph that are of the form $Conn(u, v, Y)$ where Y is contained in a biconnected component. Hence, we can prove the following. We first need a definition.

The third part of each label $M(x)$ is the *index* of a vertex, and not as the others, a label constructed by Theorem 6.2. Assume $J : V(G) \to L$ is another injective labeling where $|J(x)| = O(f(n))$ for some function f ($f(n) \geq \ulcorner \log(n) \urcorner$). We denote by $M[J]$ the new labeling N defined as follows:

$$N(x) = \Big(J(x), M_0(x), M_0(mother(x)), M_0(leader(mother(x))), J(leader(mother(x)))\Big)$$

We have clearly $|N(x)| = O(\log(n) + f(n))$.

Proposition 7.4 *Assume we have an injective $f(n)$-labeling scheme J for the graphs G of a class $\mathcal{C}$ giving the right answers to queries $Conn(u, v, Y)$ such that $Y \subseteq V(C)$ for a biconnected component C of G. Then there exists an $O(\log(n)+f(n))$-labeling scheme supporting connectivity queries $Conn(u, v, X)$ for all sets X.*

Proof. J is injective implies that $f(n) \geq \lceil \log(n) \rceil$. We take the labeling $M[J]$ where M is defined in Proposition 7.3. Note that the labeling M gives the indices of the vertices $x_1, \ldots, x_{m-1}$ and those in the sets $X \cap V(C_i)$. However, only their J-labels together with $J(u)$ and $J(v)$ are needed to obtain the truth values of $Conn(u, v, X)$ (by using Equivalence (6)). This is why we can use $M[J]$. Since J is injective the equality tests made when using M are correct if they are made with $M[J]$. □

8 The General Case

Before getting into technical details we give an overview of the proof. Extending the proof of Section 5 to the general case of planar connected graphs G presents two difficulties.

First the plane graph G^+ may have multiple edges which forbids a straight-line embedding. This situation occurs only if G is not 2-connected. A second difficulty occurs for 2-connected graphs because there is no upper bound to the number of faces to which two vertices may be incident. This situation does not occur if G is a subdivision of a 3-connected graph.

We overcome these difficulties as follows. First we replace G^+ by a simple subgraph of itself with same adjacencies, obtained by removing parallel edges. We denote this graph by G^-. The associated notion of barrier may "miss some cases of separation" because it is a subset of the original one associated with G^+. In other words if u and v are separated by the barrier associated with X in G^- they are also by the corresponding barrier in G^+, but the converse does not always hold. To handle this case, we query the tree of biconnected components as explained in Section 7. The result of this query is either that u and v are separated (case (a) of Lemma 7.2) or a "call" to several queries of the form $Conn(x, y, Y)$ where Y is included in a biconnected component. In this case, the barrier relative to G^- (in Definition 2.5 we replace G^+ by G^-) gives the correct result, because it is the same as the one relative to G^+.

The second difficulty concerns biconnected components and we use the method of Section 6. Because barriers may be unbounded, we replace them by reduced barriers to be constructed from sets Y as above. Reduced barriers can miss some cases of separation, but these cases will be detected by queries in the decomposition trees defined in Corollary 6.7. This is proved in Propositions 6.16 and 6.22. In order to obtain the general proof, we will combine the constructions of Sections 5, 6 and 7. In particular we will merge the trees $T^*(C)$ associated with biconnected components C of G and the tree $B(G)$ into a single tree $BT^*(G)$ to which we will apply simultaneously Theorem 6.2 and Propositions 6.16 and 7.3. We first explain the global structure of the proof.

Step 1. Given a connected planar graph G, we construct a straight-line planar embedding of the graph G^- defined above. We obtain thus for each vertex x of G and each face-vertex x of G^+ a pair of integer coordinates denoted by $C_0(x)$. For each vertex x of G we let $C(x)$ consist of $C_0(x)$ and of $C_0(f_1(x)), \dots, C_0(f_{24}(x))$ where $f_1, \dots, f_{24}$ are the functions of Proposition 3.5 for $m = 2$ and $f_1(x), \dots, f_{24}(x)$ are vertices of G^+ at distance at 1 or 2 of x.

Step 2. We construct a tree $BT^*(G)$ (according to Definition 8.1) and, by using Theorem 6.2, a labeling R_0 of this tree for checking 5 monadic second-order queries.

Step 3. The label $J(x)$ of a vertex x of G is then defined as

$$\Big(C(x), R_0(x), R_0(mother(x)), R_0(leader(mother(x))), C(leader(mother(x)))\Big)$$

where *mother* and *leader* are relative to the rooted tree $B(G)$ of biconnected components of G.

Connectivity Checking with labels J. Assume we are given $J(u), J(v)$ and $J(X)$ for $u, v \in V(G)$ and $X \subseteq V(G) - \{u, v\}$. We now explain how to obtain the answer to the query $Conn(u, v, X)$ in G.

Step 1. By using $R_0(u), R_0(v)$ and $R_0(X)$ we can query $BT^*(G)$ to check if some vertex of X is a separating vertex of G that separates u and v (this is possible because the tree $B(G)$ is definable in $BT^*(G)$ by monadic second-order formulas). If this is the case, we can stop and return the answer that $Conn(u, v, X)$ is false. Otherwise, we continue as follows.

Step 2. We let $C_1, \dots, C_p$ be the problematic biconnected components for (u, v, X) and let $x_1, \dots, x_{m-1}$ be leaders of some of them as in Proposition 7.3. We can determine from $R_0(u), R_0(v), R_0(X)$ the following objects: $R_0(x_1), \dots, R_0(x_{m-1})$ and $R_0(bcc^{-1}(C_1), \dots, R_0(bcc^{-1}(C_m))$ and, for each $i = 1, \dots, m$ the set $\{R_0(y) \mid y \in X \cap V(C_i)\}$. Since $x_0 = u$ and $x_m = v$, we also have $R_0(x_0)$ and $R_0(x_m)$ from $J(u)$ and $J(v)$.

Step 3. For each $i = 1, \dots, m$ we can check if there is a pair $(x, y) \in (X \cap V(C_i))^2$ that is a polar pair in C_i and separates x_{i-1} and x_i. This can be done by means of $R_0(x_{i-1}), R_0(x_i)$ and the set of labels $R_0(X \cap V(C_i))$. If one such i is found then, we can stop and report that $Conn(u, v, X)$ is false.

Step 4. For each $i = 1, \dots, m$ by using $C(x_{i-1}), C(x_i)$ and $C(X \cap V(C_i))$ which we can get from $J(u), J(v)$ and $J(X)$ when performing Step 2, we can construct the reduced barrier of $X \cap V(C_i)$ and check from it and by means of the algorithm of Section 4 whether $Conn(x_{i-1}, x_i, X \cap V(C_i))$ is true or not. By Proposition 6.22 reduced barriers suffice for this. We obtain that

$Conn(u, v, X)$ holds if and only if all conditions $Conn(x_{i_1}, x_i, X \cap V(C_i))$ are true.

To achieve this goal, we need some definitions and preliminary results.

Definition 8.1 (The Tree $BT^*(G)$ of a Connected Planar Graph) *Let G be a connected planar graph. Let us choose a vertex r that belongs to a single biconnected component as root of $B(G)$. Each biconnected component C has thus a leader, that we denote by $n(C)$. For each such component we choose a vertex adjacent to $n(C)$, we denote it by $s(C)$ and we define a bipolar orientation of C with South pole $s(C)$ and North pole $n(C)$. We make C into a plane bipolar graph by choosing an appropriate circular incidence sequence around each vertex. We combine the plane biconnected components and we make in this way G into a plane graph that we still denote by G.*

For each $C \in Bcc(G)$ we let $T^*(C)$ be the corresponding tree as defined in Section 6. If C is reduced to a single edge: $s(G) \to n(G)$ we let $T^*(C)$ be the tree $s(G) \stackrel{1}{\leftarrow} r(C) \stackrel{2}{\rightarrow} n(G)$ where $r(C)$ is its root with the convention used in Figure 11. We recall that the set of nodes of $T^*(C)$ is the union of $V(C)$ and a set of nodes labeled by $\mathbf{P}$ or $\mathbf{N}$ that represent the decomposition of C with the help of auxiliary partial functions $g_1, \ldots, g_8$.

We define $BT^*(G)$ as the union of the trees $T^*(C)$ for all $C \in Bcc(G)$. These trees have in common the nodes that are vertices of G. We let $Root(C)$ be the root of $T^*(C)$. It is not in $V(G)$, and will be taken as a node representing C, like $\mathrm{bcc}^{-1}(C)$ in $B(G)$ (cf. Section 7 for notation about $B(G)$).

The following facts are clear from the definitions.

Fact 8.2 *The graph $BT^*(G)$ is a directed tree. Its nodes labeled by $\mathbf{V}$ are the vertices of G. Its nodes of indegree 0 are in bijection by a function, that we will denote by Root, with $Bcc(G)$ and thus with the set $W(G)$ of $B(G)$. For each $C \in Bcc(G)$ its leader and North pole $n(C)$ is the unique vertex x such that $Root(C) \stackrel{2}{\rightarrow} x$ in $BT^*(G)$. The nodes of $T^*(C)$ are the nodes of $BT^*(G)$ accessible from $Root(C)$ by a directed path, and $T^*(C)$ is the sub-tree of $BT^*(G)$ induced on this set.*

Example 8.3 *Let W be the directed plane graph on Figure 14. Its biconnected components are bipolar. Letting g_3 map 4 to 5 (no other value of g_3 and no other function $g_4, \ldots, g_8$ are needed), its tree $BT^*(W)$ is shown on Figure 15.*

Fact 8.2 shows that the trees $T^*(C)$, for $C \in Bcc(G)$, are induced sub-trees of $BT^*(G)$, and that their sets of nodes are definable by MS formulas. The tree $B(G)$, the tree of biconnected components of G, is also definable in $BT^*(G)$ by MS formulas. If N is the set of nodes of $BT^*(G)$, we let:

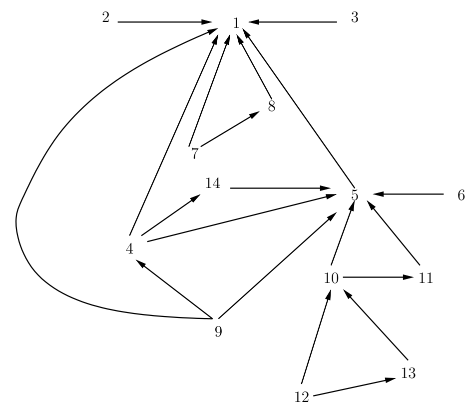

Fig. 14. A directed plane graph W

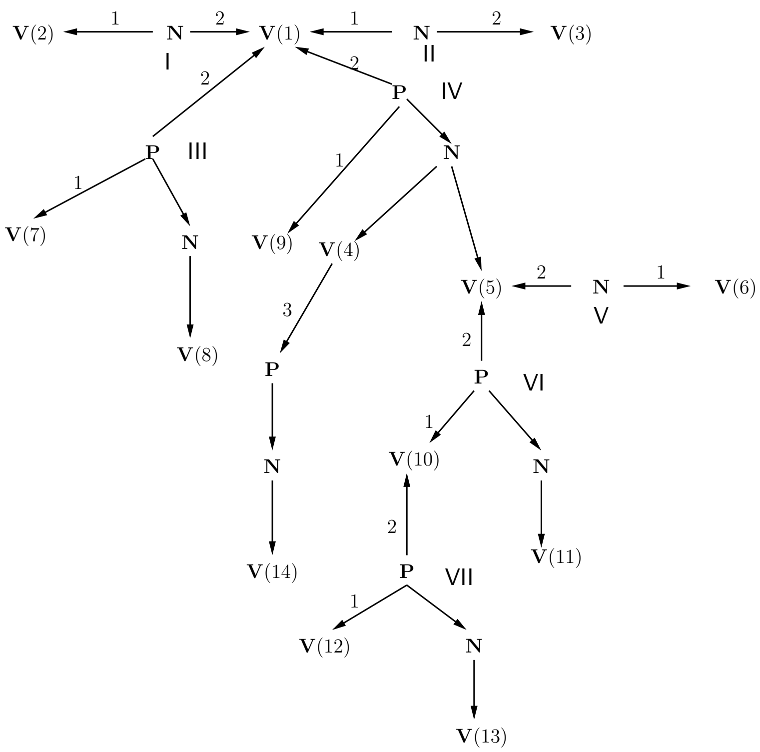

Fig. 15. The tree $BT^*(W)$ of the graph W

(1) V be the set of nodes labeled by $\mathbf{V}$.
(2) W be the set of nodes in N of in-degree 0.
(3) $member'$ be the binary relation such that $member'(v, w)$ holds if and only if $v \in V$, $w \in W$ and $w \xrightarrow{*} v$ in $BT^*(G)$.
(4) $\leq'$ be the reflexive and transitive closure of the relation $<_0$ defined as follows:

$u <_0 u'$ if and only if either $u \in W$, $u' \in V$ and $u \xrightarrow{2} u'$, or $member'(u, u')$ and we do not have $u' \xrightarrow{2} u$.

We have the following fact.

Fact 8.4 *The sets V, W, the relations $member'$ and $\leq'$ are definable in $BT^*(G)$ by MS formulas. The structure $\langle V \cup W, member', \leq' \rangle$ is isomorphic to $B(G)$ with $V = V(G)$ and W in bijection by Root with $Bcc(G)$.*

The queries $\leq, member, P_1, P_2$ for which we constructed in Proposition 7.3 an $O(\log(n))$-labeling can be translated into MS queries over $BT^*(G)$, denoted by $\leq', member', P_1', P_2'$.

We consider next the construction done for proving Proposition 6.16. Let us first introduce some notations and a lemma. Let C be a biconnected component of a connected graph G. For every vertex u of G we let:

$$Att(u, C) := \begin{cases} u & \text{if } u \in C, \\ u' & \text{if } u \notin C \text{ and } u' \text{ is the unique vertex of } C \\ & \text{on the path in } B(G) \text{ that links } u \text{ and } bcc^{-1}(C) \end{cases}$$

In other words, u' is the first vertex of C on any path in G from u to some vertex of C. (We write $Att_G(u, C)$ if G must be specified.)

Lemma 8.5 *There exists a monadic second-order formula $\alpha(u, u', w)$ such that for every connected planar graph G, $BT^*(G) \models \alpha(u, u', w)$ if and only if $u, u' \in V(G), w = Root(C)$ for some biconnected component C of G and $u' = Att_G(u, C)$.*

Proof. We let $\alpha(u, u', w)$ express the following: u and u' are labeled by $\mathbf{V}$, w is of in-degree 0, there is a directed path from w to u' and, either $u = u'$ (which implies $u = bcc^{-1}(w)$) or there is an undirected path between u and u' containing an edge $y \to u'$ that does not belong to the path from w to u'. It follows from the definitions that these conditions are equivalent to $u' = Att_G(u, C)$. □

Example 8.6 (Continuation of Example 8.3) *Consider the tree on Figure 15. The nodes marked I,II,...,VII (in Roman numbers) are those of the form $Root(C)$. We have in particular $10 = Att(2, C) = Att(6, C) = Att(5, C) = Att(10, C)$ where VII=$Root(C)$. The validity of the definition of α can be checked on these examples.*

We have used in Proposition 6.16 the query $Q(u, v, x, y)$ relative to a bipolar plane graph meaning "(x, y) is a polar pair separating u and v". We let $Q_1(u, v, x, y, w)$ mean for nodes u, v, x, y, w of $BT^*(G)$:

> "$w = Root(C)$ for some biconnected component C and (x, y) is a polar pair of C separating u and v in C".

We will rather use the property $Q'(u,v,x,y,w)$ meaning:

" $w = Root(C)$ for some biconnected component C, (x,y) is a polar pair of C that separates $Att(u,C)$ and $Att(v,C)$"

that is equivalent to

$$\exists u', v'[\alpha(u,u',w) \wedge \alpha(v,v',w) \wedge Q_1(u',v',x,y,w)]$$

Since $T^*(C)$ is the union of the directed paths in $BT^*(G)$ originating from $Root(C)$ so that its set of nodes is MS-definable in $BT^*(G)$, the queries Q_1 and Q' can be expressed in $BT^*(G)$ by monadic second-order formulas.

Proposition 8.7 *For every connected planar graph with associated tree $BT^*(G)$ constructed as in Definition 8.1, we can build in time $O(n \cdot \log(n))$ an $O(\log(n))$-labeling R_0 of the associated tree $BT^*(G)$ that supports the queries $\leq'$, $member'$, P_1', P_2' and Q'.*

Proof. Immediate consequence of Theorem 6.2 and the previous remarks. □

The construction time of $O(n \cdot \log(n))$ can be reduced to $O(n)$ if a similar improvement is possible for Theorem 6.2. We let then for each $x \in V_G$:

$$R(x) = \Big(R_0(x), R_0(mother(x)), leader(mother(x))\Big). \tag{8}$$

It is constructed like M in Proposition 7.3, and refines the labeling K of Proposition 6.16. It makes it possible to query, not only the global structure of G defined by $B(G)$, but also the internal structure of each biconnected component.

Next we adapt the notion of reduced barrier, and we generalize Propositions 2.6 and 6.22.

Definition 8.8 (Augmented Graphs of Biconnected Components) *For every graph H we let $Spl(H)$ be a simple graph obtained from H by removing edges and such that H and $Spl(H)$ have same adjacency relation. Let G be a connected plane graph, and G^+ be its augmented graph. If G^+ has parallel edges linking two vertices x and y then one of them is a face and the other is a separating vertex. We let G^- be $Spl(G^+)$. Hence G^- is a simple connected plane graph.*

Let G be a plane graph and let C be a biconnected component of G. We denote by $F_G(C)$ the set of faces $f \in F(G)$ that are incident with an edge of C

equivalently such that there exist two adjacent vertices x, y of C such that $f - x$ and $f - y$ are edges of G^+.

Lemma 8.9 *Let G be a simple connected plane graph and $\mathcal{E}$ be an embedding of G^+. We let $\mathcal{E}^-(C)$ be the restriction of $\mathcal{E}$ to $G^-[V(C) \cup F_G(C)]$ for some biconnected component C of G. Then $\mathcal{E}^-(C)$ is an embedding of C^+.*

Before proving the lemma we show an example.

Example 8.10 *Consider the graph G^+ of Figure 1. It is not simple. Let G^- be obtained by deleting a and c, and let $\mathcal{E}^-$ be the corresponding planar embedding. Let C be the biconnected component with $V(C) = \{x, t, v\}$. Then the restriction of $\mathcal{E}^-$ to $G^-[V(C) \cup F_G(C)]$ is shown in Figure 16. It is an embedding of C^+.*

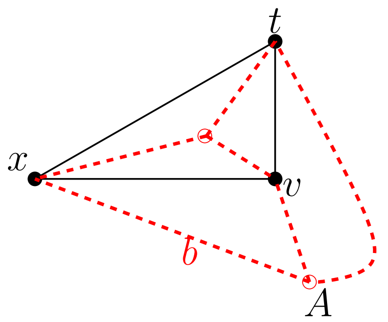

Fig. 16. Illustration of Example 8.10.

Proof. It is clear that the restrictions of $\mathcal{E}$ and $\mathcal{E}^-$ to C coincide and form an embedding $\mathcal{E}''$ of C. Each face $f \in F_G(C)$ defines a unique face f'' of $\mathcal{E}''$. We first prove that $\overline{f} \neq \overline{f'}$ if $f \neq f'$.

Assume this is not the case. The border cycle Γ of f (considered as a face of G) contains at least one edge of C and at least one edge not in C because it separates f and f' in $\mathcal{E}$ and does not in $\mathcal{E}^-$ (since we assume $f \neq f'$ and $\overline{f} = \overline{f'}$). Hence Γ contains a nonempty path with no edge in C that links two distinct vertices of C. This is not possible since we assumed that C is a biconnected component of G. It follows that the mapping $f \mapsto \overline{f}$ that maps $F_G(C)$ into $F(C)$ is injective.

Conversely, let $g \in F(C)$ with the corresponding open subset of the plane $\mathcal{E}''(g)$ associated with the embedding $\mathcal{E}''$. Each biconnected component of G is either embedded by $\mathcal{E}$ in $\mathbb{R}^2 - \mathcal{E}''(g)$ or in $\mathcal{E}''(g) \cup \mathcal{E}''(\Gamma)$. It is clear that $\mathcal{E}''(g) - \bigcup\{\mathcal{E}(D) \mid D \text{ is a biconnected component of } G,\ D \neq C\}$ is $\mathcal{E}(f)$ for some face $f \in F_G(C)$ and that $g = \overline{f}$. Hence we have a bijection $f \mapsto \overline{f}$ of $F_G(C)$ onto $F(C)$.

In $G^+[V(C) \cup F_G(C)]$ there are several edges between f (such that $g = \overline{f}$ as above) and a vertex x of G if some biconnected component D of G is embedded by $\mathcal{E}$ in $\mathcal{E}''(g) \cup \mathcal{E}''(\Gamma)$ and is such that $V(D) \cap V(C) = \{x\}$. In $G^-[V(C) \cup F_G(C)]$ only one remains in such a case between f and x. It is follows that the restriction of $\mathcal{E}$ to $G^-[V(C) \cup F_G(C)]$ is an embedding of C^+. $\square$

Definition 8.11 (Reduced Barriers for Connected Graphs) *Let G, G^+, G^- be as in Definition 8.8. For $X \subseteq V(G)$ we define its reduced barrier $RBar(X)$ as a set of edges from G^-, defined as follows:*

$$RBar(X) = \bigcup_{\substack{x,y \in X \\ x \neq y}} RBar(\{x,y\}) \tag{9}$$

where $RBar(\{x,y\})$ is the set $Bar(\{x,y\}) \cap E(G^-)$ if x and y are incident with at most 2 faces, otherwise $RBar(\{x,y\})$ consists of the edges $x - f$ and $y - f$ of G^- where $f = Select(x,y)$, and, as in Section 6, Select associates with every two vertices that are incident with at least 3 faces one of these faces.

We recall that since G^- is plane without multiple edges, it has a straight-line embedding $\mathcal{E}_0$.

Lemma 8.12 *Let C be a biconnected component of G with a bipolar orientation and adjacent poles (according to Definition 8.1). Let $X \subset V(C)$, let $u, v \in V(G) - X$ that are either in $V(C)$, or are connected to $V(C)$ by paths that do not go through X and be such that $Att_G(u,C)$ and $Att_G(v,C)$ are not separated in C by a polar pair in $X \times X$. Then u and v are separated in G by X if and only if they are separated by $RBar(X, \mathcal{E}_0)$.*

Proof. Let us extend $\mathcal{E}_0$ into an embedding $\mathcal{E}$ of G^+ with edges in $E(G^+) - E(G^-)$ represented by curve segments so that $\mathcal{E}^- = \mathcal{E}_0$. If u and v are separated in the plane by $RBar(X, \mathcal{E}_0)$, they are separated by $RBar(X, \mathcal{E}^-)$, hence they are also separated by X in G.

For the other direction let u, v be separated by X in G. Then $u' = Att_G(u,C)$ belongs to $V(C) - X$ and is linked to u by a path avoiding X. Let $v' = Att_G(v,C)$ be similarly linked to v. Clearly, u' and v' are separated in C by X. By the hypothesis, Case (b) of Proposition 6.22 applies and u' and v' are separated in the plane by $RBar(X, \mathcal{E}^-(C))$ where $\mathcal{E}^-(C)$ is the embedding of C^+ from Lemma 8.9 defined as a restriction of $\mathcal{E}_0 = \mathcal{E}^-$. Hence u' and v' are separated by $RBar(X, \mathcal{E}_0)$ in the plane. Each of the two paths linking u to u' and v to v' avoids X, hence is in a connected component of $\mathbb{R}^2 - RBar(X, \mathcal{E}_0)$. Hence u and v are also separated in the plane by $RBar(X, \mathcal{E}_0)$, as was to be proved. $\square$

Example 8.13 *We use W of Example 8.3. Figure 17 shows the graph W^-. We have $F(W) = \{A, B, C, \ldots, F, G, H\}$. We do not show in full all edges in-*

cident with A. Let Z be the biconnected component with $V(Z) = \{1, 4, 5, 9, 14\}$, $s(Z) = 9$, $n(Z) = 1$. Then Z^+ consists of Z augmented with the following edges: $A-1, A-5, A-9,\ , C-1, C-5, C-14, C-4, D-4, D-14, D-5, E-4, E-5, E-9, H-1, H-4, H-9$. It is clear that $Z^+ = W^-[\{1, 4, 5, 9, 14, A, C, D, E, H\}]$.

Let $X = \{1, 4, 5\}$. Condition (a) of Lemma 7.2 shows that 2 and 3, and 9 and 14 are separated by X. Note that 4 and 5 form a //-polar pair. They are incident with 3 faces; 1 and 4 form also a polar pair but not a //-polar pair.

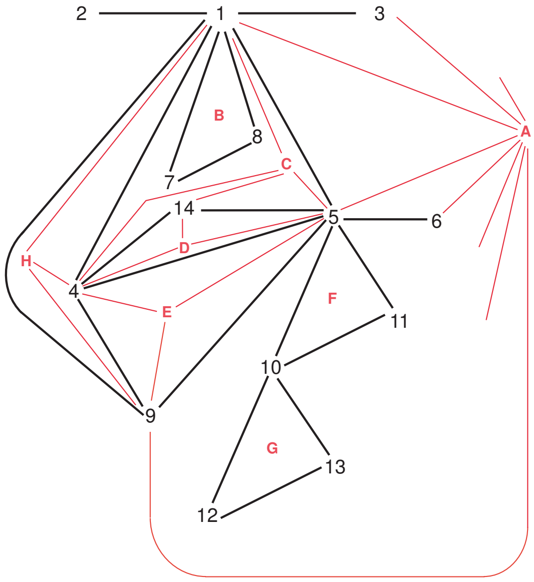

Fig. 17. The graph W^- of Example 8.13.

Proof of Theorem 1.1(Main Theorem). We first consider connectivity queries in induced subgraphs defined by excluded vertices, as we did in Theorems 5.1 and 6.24.

Let be given a connected planar graph G and let its associated tree $BT^*(G)$ be as explained in Definition 8.1. This can be done in time $O(n)$, using classical depth-first algorithms.

Then we define $G^- = Spl(G^+)$ by eliminating edges from G^+ and we define a straight-line embedding $\mathcal{E}_0$ of G^- in $\mathbb{R}^2$ with integer coordinates of absolute value in $[3n - 6]$. We can use here Schnyder's algorithm [15]. Each vertex of G^-, i.e, each element x of $V(G) \cup F(G)$ has a pair of integer coordinates $C_0(x)$ of size at most $2 \cdot (\lceil \log(n) \rceil + \log(3))$.

We let $m = 2$ and we will use 24 unary functions $f_i, i \in [24]$ (cf Section 3) in order to construct the necessary reduced barriers. We let thus for every $x \in V(G)$

$$C(x) = (C_0(x), C_0(f_1(x)), \ldots, C_0(f_{24}(x)))$$

We also determine the labels $R(x)$ for $x \in V(G)$ by Proposition 8.7. They make possible to query the tree $BT^*(G)$. The final labeling is $J(x) = R[C](x)$ clearly of size $O(\log(n))$. This labeling can be constructed in time $O(n{\cdot}\log(n))$.

We explained at the beginning of the section how J can be used to answer queries $Conn(u, v, X)$. We add a few remarks:

About Step 1 and 2. Since the tree $B(G)$ is definable in $BT^*(G)$ by monadic second-order formulas (Fact 8.4) the 4 queries over it used in Section 7 can be expressed as MS queries over $BT^*(G)$, and the labeling R_0 makes it possible to answer them.

About Step 3. We must answer for each i an extended connectivity query $Conn(x_{i-1}, x_i, X \cap V(C_i))$ where x_{i-1} and x_i may be outside of C_i. Hence, it is not sufficient to use a translation of Q (used in Section 6) into a query over $BT^*(G)$. However, $Conn(x_{i-1}, x_i, X \cap V(C_i))$ on G is equivalent to the query

$$Conn(Att_G(x_{i-1}, C_i), Att_G(x_i, C_i), X \cap V(C_i))$$

in C_i. The definition of Q' is based on this observation.

About Step 4. The correctness of the final answer is ensured by Lemma 8.12.

In order to handle forbidden-edge queries $Conn(u, v, X, Y)$ (where X is a set of vertices, Y a set of edges), we transform G by subdividing each edge (or only each "unsafe" edge, for which deletions may have to be handled), i.e., by inserting a new vertex w_e on each edge e. We obtain a graph G' which is simple, connected and planar. It is clear that u and v are connected in $(G - Y) \setminus X$ if and only if they are connected in $G' \setminus X'$ where $X' = X \cup \{w_e \mid e \in Y\}$. Hence we can apply to G' the above described construction, and we obtain an $O(\log(n))$-labeling J' of vertices G', whereas we wish an $O(\log(n))$-labeling J of the edges and vertices of G, since edges to delete are specified as pairs of adjacent vertices.

We use again unary functions to specify edges from pairs of vertices. We let $g_1, g_2, g_3 : V(G) \to V(G)$ be 3 functions as in Lemma 3.3. We let g_4, g_5, g_6 be the 3 functions : $V(G) \to E(G)$ defined as follows:

$$g_{i+3}(x) = e \text{ if } e \text{ is the edge } x - g_i(x).$$

These 6 functions represent adjacency and the binary function $Edg : V(G) \times V(G) \to E(G)$ that associates with (x, y) the edge $x - y$ if it exists. We let thus $J(x)$ be defined as:

$$(J'(x), J'(g_1(x)), \ldots, J'(g_6(x)))$$

for every $x \in V(G)$.

For a family $(x_y, z_y)_{y \in Y}$ of pairs of adjacent vectices defining a set Y of edges to be deleted, we get from the labels $J(x_y), J(z_y)$ for $y \in Y$ the labelings $J'(w_y)$. We can thus decide from $J(u), J(v), (J(x_y), J(z_y))_{y \in Y}$, and $(J(x))_{x \in X}$ whether $Conn_{G'}(u, v, X \cup \{w_y \mid y \in Y\})$ holds, i.e., whether $Conn_G(u, v, X, Y)$ holds. It is clear that $|J(x)| = O(\log(n))$ and that the computation times for constructing J' and J and answering queries is as in the initial case. □

We can also consider extended connectivity queries that include additional edges. The idea is simple: for a set X of vertices, and H of edges connecting vertices in $G \setminus X$, we check for each endpoint in H the connected component of $G \setminus X$ to which it belongs. The following makes this precise.

Corollary 8.14 *Theorem 1.1 extends to edge additions.*

Proof. Let X and F be respectively the set of vertices and the set of edges to delete and let H be a set of new links, defined as a set of pairs of (x, y) for $x, y \in A \subseteq V(G) - X,\ x \neq y$ is added. We use the previous constructions as follows in order to answer the query $Conn(u, v, X, F, H)$ (cf. Introduction):

- We build the reduced barrier associated with (X, F).
- For any two vertices u', v' in $A \cup \{u, v\}$ we can determine whether $Conn(u', v', X, F)$ holds; we let C be the set of all such pairs $\{u', v'\}$ that are connected in $(G - F) \backslash X)$.
- We build the graph G' with vertex set $A \cup \{u, v\}$ and set of edges $H \cup C$.

Then $Conn(u, v, X, F, H)$ holds if and only if u and v are connected in the graph G'. □

The labeling of Corollary 8.14 can be applied to single crossing graphs or in general to classes of graphs of bounded crossing number (see [18]).

Remark 8.15 *For planar graphs of degree at most d, we need not use the tools of Section 6 because their 2-connected components are d-face bounded. However, the tree of biconnected components (Section 7) remains necessary. For them we use Proposition 7.4.*

9 Related Work

There has been a lot of work on answering connectivity queries after a single update to the network, by studying bridges and articulation points in graphs.

Handling the case of multiple updates, such as multiple failed vertices or edges, is significantly more difficult. Obviously, on every batch of updates, one can recompute a connectivity oracle (such as the Thorup-Zwick scheme [19], which answers standard connectivity queries in $O(1)$ time and space $\tilde{O}(n^{1/2})$) and then make queries to it. But this is inefficient if the network changes often, or even worse, in an emergency planning situation where queries need to be made without the time to recompute labels or new oracles. In this situation, it is also important to have algorithms with good worst-case bounds on the query time, rather than amortized bounds. It is this setup that our work lies in.

For general graphs, Pătraşcu and Thorup [13] give a centralized construction that answers extended connectivity queries of the form "are vertices u, v in the same connected component in $G - F$, where F is a set of d of deleted edges. Their oracle answers queries in time $O(d\text{polylog}n)$, after preprocessing the graph. It is not clear if their construction extends to handle vertex deletions with similar time and space bounds.

We will now explain how our construction can be modified to give 'oracle-like' bounds when the set X is the same for several queries. One can decompose the algorithm into the following general steps:

Step 1. For a graph G, construct a global data structure $S(G)$ or a labeling.

Step 2. For given sets X and F of deleted vertices and edges, and by using $S(G)$ or the labels of vertices in X and those of the ends of the edges in F, construct an intermediate data structure $T(G, X, F)$.

Step 3. For any two vertices u, v, quickly answer $Conn(u, v, X)$.

For a connected planar graph with n vertices, we can perform Step 1 in time $O(n)$ for constructing $S(G)$ and in time $O(n \log(n))$ for a labeling. Then given X of size m, we can construct $T(G, X)$, i.e., the data structure of Theorem 4.1 associated with the reduced barrier in expected time $O(m \log(m))$ (the reduced barrier is constructed in $O(m^2)$). After this, each query $Conn(u, v, X)$ with $u, v \notin X$ can be answered in time $O(\log(m))$.

It is open whether we can also efficiently answer queries of the form $Conn(u, v, Y)$ with $Y \subseteq X$, and $u, v \notin X$ in time $O(\log(m))$ by considering the subset of the reduced barrier associated with Y.

10 Conclusion

We conjecture that the main theorem extends to graphs embedded in any fixed surface, in particular graphs of bounded genus or those excluding a fixed minor.

An interesting problem is to investigate constructions of graph classes and combining their labeling schemes: if $\mathcal{C}, \mathcal{D}$ are two graph classes supporting extended connectivity queries (with small labels) and $\mathcal{F}$ is defined as a class of combinations of graphs in $\mathcal{C}$ and $\mathcal{D}$ (by operations like substitutions or clique-sums), then we would like to be able to combine the labeling schemes of $\mathcal{C}$ and $\mathcal{D}$ into one supporting extended connectivity queries on graphs in $\mathcal{F}$, still using short labels.

For example, Kanté [11] has considered graphs that are obtained by "gluing" graphs of small clique-width such that their intersection graph [6] is planar and has bounded degree.

References

[1] M. de Berg, M. Van Kreveld, M. Overmars and O. Schwarzkopf. *Computational Geometry : Algorithms and Applications*. Springer, 1991.

[2] B. Courcelle. The Monadic Second-Order Logic of Graphs XII: Planar Graphs and Planar Maps. *Theoretical Computer Science*, 237(1-2):1-32, 2000.

[3] B. Courcelle, A. Twigg Constrained-path labellings on graphs of bounded clique-width, To appear in *Theory of Computing Systems*, 2009.

[4] B. Courcelle and R. Vanicat. Query Efficient Implementation of Graphs of Bounded Clique-width. *Discrete Applied Mathematics*, 131(1):129-150, 2003.

[5] B. Courcelle, C. Gavoille and M.M. Kanté. Efficient First-Order Model Checking Using Short Labels. In F.P. Preparata, X. Wu and J. Yin editors, *Frontiers in Algorithmics Workshop*, volume 5059 of LNCS, pages 159-170. Springer, 2008.

[6] B. Courcelle, C. Gavoille, M. Kanté, and A. Twigg. Connectivity check in 3-connected planar graphs with obstacles. In *Electronic Notes in Discrete Mathematics*, volume 31 (2008), 151–155.

[7] G. Di Battista and R. Tamassia. On-Line Planarity Testing. *SIAM Journal on Computing*, 25(5):956-997, 1996.

[6] The intersection graph of $G_1, \ldots, G_m$ is the graph with set of vertices $x_1, \ldots, x_m$ and there is an edge $x_i x_j$ whenever G_i and G_j intersects.

[8] R. Diestel. *Graph Theory.* Springer, 3^{rd} edition, 2005.

[9] H. de Fraysseix, P. Ossona de Mendez and P. Rosenstiehl. Bipolar Orientations Revisited. *Discrete Applied Mathematics*, 56(2-3):157-179, 1995.

[10] H. de Fraysseix, J. Pach and R. Pollack. How to Draw a Planar Graph on a Grid. *Combinatorica*, 10(1):41-51, 1990.

[11] M. M. Kanté. *Graph Structurings: Some Algorithmic Applications.* PhD Thesis, Université Bordeaux 1, 2008.

[12] B. Mohar and C. Thomassen. *Graphs on Surfaces.* The Johns Hopkins University Press, Baltimore and London, 2001.

[13] M. Pătraşcu and M. Thorup. Planning Fast Connectivity Updates. In 48^{th} *Symposium on Foundations of Computer Science*, pages 263-271. IEEE Computer Society, 2007.

[14] N. Sarnak and R.E. Tarjan. Planar Point Location Using Persistent Search Trees. *Communications of the ACM*, 29(7):669-679, 1986.

[15] W. Schnyder. Embedding Planar Graphs on the Grid. *First ACM-SCAM Symposium on Discrete Algorithms*, pages 138-148. SIAM, 1990.

[16] J. Snoeyink. Point Location. Chapitre 34, pp. 767-785 in *Handbook of Discrete and Computational Geometry (Second Edition)*, edited by J.E. Goodman and J. O'Rourke. Chapman and Hall/CRC (2004).

[17] R. Tamassia and I. G. Tollis. A Unified Approach and Visibility Representation of Planar Graphs. *Discrete & Computational Geometry*, 1:321-341, 1986.

[18] J.A. Telle and D.R. Wood. Planar Decompositions and the Crossing Number of Graphs with an Excluded Minor. *New York Journal of Mathematics*, 13:117-146, 2007.

[19] M. Thorup and U. Zwick. Approximate distance oracles. *J. ACM*, 52(1):1–24, 2005, (ACM)