2.4 Triangulating a plane graph

A large and important class of 3-connected graphs is formed by the maximal planar graphs. A graph is *maximal planar* if no edge can be added so that the resulting graph is still planar.

Lemma 2.23 *A maximal planar graph on* $n \geqslant 3$ *vertices is biconnected.*

Proof. Consider a maximal planar graph $G = (V, E)$. If G is not biconnected, then it has a cut-vertex v. Take a plane drawing Γ of G. As $G \setminus v$ is disconnected, removal of v also splits $N_G(v)$ into at least two components. Therefore, there are two vertices $a, b \in N_G(v)$ that are adjacent in the circular order of vertices around v in Γ and are in different components of $G \setminus v$. In particular, $\{a, b\} \notin E$ and we can add this edge to G (routing it very close to the path (a, v, b) in Γ) without violating planarity. This is in contradiction to G being maximal planar and so G is biconnected. □

Lemma 2.24 *In a maximal planar graph on* $n \geqslant 3$ *vertices, all faces are topological triangles, that is, each is bounded by exactly three edges.*

Proof. Consider a maximal planar graph $G = (V, E)$ and a plane drawing Γ of G. By Lemma 2.23 we know that G is biconnected and so by Lemma 2.17 every face of Γ is bounded by a cycle. Suppose that there is a face f in Γ that is bounded by a cycle $v_0, \ldots, v_{k-1}$ of $k \geqslant 4$ vertices. We claim that at least one of the edges $\{v_0, v_2\}$ or $\{v_1, v_3\}$ is not present in G.

Suppose to the contrary that $\{\{v_0, v_2\}, \{v_1, v_3\}\} \subseteq E$. Then we can add a new vertex v' in the interior of f and connect v' inside f to all of v_0, v_1, v_2, v_3 by an edge (curve) without introducing a crossing. In other words, given that G is planar, also the graph $G' = (V \cup \{v'\}, E \cup \{\{v_i, v'\} \mid i \in \{0, 1, 2, 3\}\})$ is planar. However, v_0, v_1, v_2, v_3, v' are branch vertices of a K_5 subdivision in G': v' is connected to all other vertices within f, along the boundary ∂f of f each vertex v_i is connected to both $v_{(i-1) \bmod 4}$ and $v_{(i+1) \bmod 4}$ and the missing two connections are provided by the edges $\{v_0, v_2\}$ and $\{v_1, v_3\}$ (Figure 2.16a). By Kuratowski's Theorem this is in contradiction to G' being planar. Therefore, one of the edges $\{v_0, v_2\}$ or $\{v_1, v_3\}$ is not present in G, as claimed.

So suppose without loss of generality that $\{v_1, v_3\} \notin E$. But then we can add this edge (curve) within f to Γ without introducing a crossing (Figure 2.16b). It follows that the edge $\{v_1, v_3\}$ can be added to G without sacrificing planarity, which is in contradiction to G being maximal planar. Therefore, there is no such face f bounded by four or more vertices. □

The proof of Lemma 2.24 also contains the idea for an algorithm to *topologically triangulate* a plane graph.

Theorem 2.25 *For a given connected plane graph* $G = (V, E)$ *on* n *vertices one can compute in* $O(n)$ *time and space a maximal plane graph* $G' = (V, E')$ *with* $E \subseteq E'$.

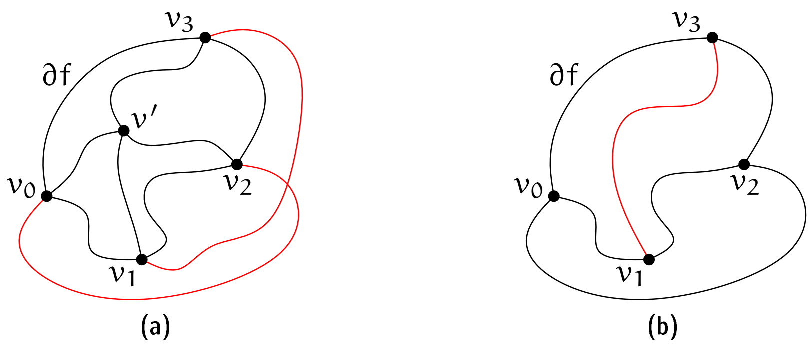


Figure 2.16: *Every face of a maximal planar graph is a topological triangle.*

Proof. Suppose, for instance, that G is represented as a DCEL[2], from which one can easily extract the face boundaries. If some vertex v appears several times along the boundary of a single face, it is a cut-vertex. We fix this by adding an edge between the two neighbors of all but the first occurrence of v. This can easily be done in linear time by maintaining a counter for each vertex on the face boundary. The total number of edges and vertices along the boundary of all faces is proportional to the number of edges in G, which by Corollary 2.5 is linear. Hence we may suppose that all faces of G are bounded by a cycle.

For each face f that is bounded by more than three vertices, select a vertex v_f on its boundary and store with each vertex all faces that select it. Then process each vertex v as follows: First mark all neighbors of v in G. Then process all faces that selected v. For each such face f with $v_f = v$ iterate over the boundary $\partial f = (v, v_1, \ldots, v_k)$, where $k \geqslant 3$, of f to test whether there is any marked vertex other than the two neighbors v_1 and v_k of v along ∂f.

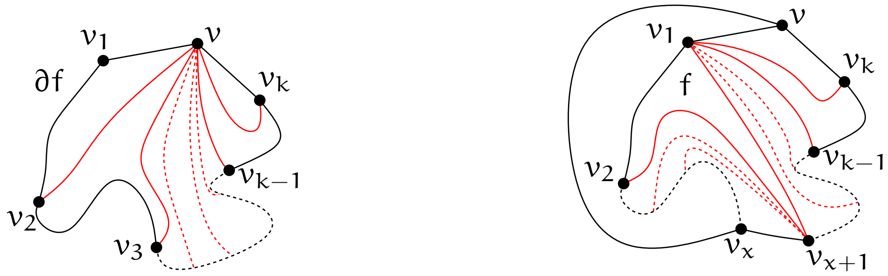
If there is no such vertex, we can safely triangulate f using a star from v, that is, by adding the edges $\{v, v_i\}$, for $i \in \{2, \ldots, k-1\}$ (Figure 2.17a).

Otherwise, let v_x be the first marked vertex in the sequence $v_2, \ldots, v_{k-1}$. The edge $\{v, v_x\}$ that is embedded as a curve in the exterior of f prevents any vertex from $v_1, \ldots, v_{x-1}$ from being connected by an edge in G to any vertex from $v_{x+1}, \ldots, v_k$. (This is exactly the argument that we made in the proof of Lemma 2.24 above for the edges $\{v_0, v_2\}$ and $\{v_1, v_3\}$, see Figure 2.16a.) In particular, we can safely triangulate f using a bi-star from v_1 and v_{x+1}, that is, by adding the edges $\{v_1, v_i\}$, for $i \in \{x+1, \ldots, k\}$, and $\{v_j, v_{x+1}\}$, for $j \in \{2, \ldots, x-1\}$ (Figure 2.17b).

Finally, conclude the processing of v by removing all marks on its neighbors.

Regarding the runtime bound, note that every face is traversed a constant number of times. In this way, each edge is touched a constant number of times, which by Corollary 2.5 uses linear time overall. Similarly, the vertex marking is done at most twice (mark und unmark) per vertex. Therefore, the overall time needed can be bounded by

[2] If you wonder how the—possibly complicated—curves that correspond to edges are represented: they do not need to be, because here we need a representation of the combinatorial embedding only.



(a) **Case 1:** v does not have any neighbor on ∂f other than v_1 and v_k.

(b) **Case 2:** v has a neighbor v_x on ∂f other than v_1 and v_k.

Figure 2.17: *Topologically triangulating a plane graph.*

$\sum_{v \in V} \deg_G(v) = 2|E| = O(n)$ by the Handshaking Lemma and Corollary 2.5. □

Theorem 2.26 *A maximal planar graph on $n \geqslant 4$ vertices is 3-connected.*

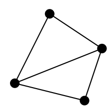
Exercise 2.27 *Prove Theorem 2.26.*

Using any of the standard planarity testing algorithms we can obtain a combinatorial embedding of a planar graph in linear time. Together with Theorem 2.25 this yields the following

Corollary 2.28 *For a given planar graph $G = (V, E)$ on n vertices one can compute in $O(n)$ time and space a maximal planar graph $G' = (V, E')$ with $E \subseteq E'$.* □

The results discussed in this section can serve as a tool to fix the combinatorial embedding for a given graph G: augment G using Theorem 2.25 to a maximal planar graph G', whose combinatorial embedding is unique by Theorem 2.22.

Being maximal planar is a property of an abstract graph. In contrast, a geometric graph to which no straight-line edge can be added without introducing a crossing is called a *triangulation*. Not every triangulation is maximal planar, as the example depicted to the right shows.



It is also possible to triangulate a geometric graph in linear time. But this problem is much more involved. Triangulating a single face of a geometric graph amounts to what is called "triangulating a simple polygon". This can be done in near-linear[3] time using standard techniques, and in linear time using Chazelle's famous algorithm, whose description spans a fourty pages paper [6].

Exercise 2.29 *We discussed the DCEL structure to represent plane graphs in Section 2.2.1. An alternative way to represent an embedding of a maximal planar graph is the following: For each triangle, store references to its three vertices and*

[3] $O(n \log n)$ or—using more elaborate tools—$O(n \log^* n)$ time

to its three neighboring triangles. Compare both approaches. Discuss different scenarios where you would prefer one over the other. In particular, analyze the space requirements of both.

Connectivity serves as an important indicator for properties of planar graphs. Another example is the following famous theorem of Tutte that provides a sufficient condition for Hamiltonicity. Its proof is beyond the scope of our lecture.

Theorem 2.30 (Tutte [24]) *Every 4-connected planar graph is Hamiltonian.*

Moreover, for a given 4-connected planar graph a Hamiltonian cycle can also be computed in linear time [7].

2.5 Compact straight-line drawings

As a next step we consider plane embeddings in the geometric setting, where every edge is drawn as a straight-line segment. A classical theorem of Wagner and Fáry states that this is not a restriction in terms of plane embeddability.

Theorem 2.31 (Fáry [9], Wagner [25]) *Every planar graph has a plane straight-line embedding (that is, it is isomorphic to a plane straight-line graph).*

This statement is quite surprising, considering how much more freedom arbitrarily complex Jordan arcs allow compared to line segments, which are completely determined by their endpoints. In order to further increase the level of appreciation, let us note that a similar "straightening" is not possible, when fixing the point set on which the vertices are to be embedded: For a given planar graph $G = (V, E)$ on n vertices and a given set $P \subset \mathbb{R}^2$ of n points, one can always find a plane embedding of G that maps V to P [20]. However, this is not possible in general with a plane *straight-line* embedding. For instance, K_4 does not admit a plane straight-line embedding onto a set of points that form a convex quadrilateral, such as a rectangle. In fact, it is NP-hard to decide whether a given planar graph admits a plane straight-line embedding onto a given point set [5].

Although Fáry-Wagner's theorem has a nice inductive proof, we will not discuss it here. Instead we will prove a stronger statement that implies Theorem 2.31.

A very nice property of straight-line embeddings is that they are easy to represent: We need to store points/coordinates for the vertices only. From an algorithmic and complexity point of view the space needed by such a representation is important, because it appears in the input and output size of algorithms that work on embedded graphs. While the Fáry-Wagner Theorem guarantees the existence of a plane straight-line embedding for every planar graph, it does not provide bounds on the size of the coordinates used in the representation. But the following strengthening provides such bounds, by describing an algorithm that embeds (without crossings) a given planar graph on a linear size integer grid.