

Maximal planar graphs

Fact 1.

In a maximal planar graph, every face is a triangle. Otherwise, it is not maximal planar.

Fact 2.

Let G be a maximal planar graph and f a face of G . Extend G to G' by adding one vertex inside of G and connecting to every vertex defining f . Then, G' is maximal planar.

Proof. If f is an inner face, then it is a triangle with three vertices defining f . The vertex placed inside divides f into three triangles, since it is connected to every of the three vertices defining f . G' has now $v + 1$ vertices, three more edges and one face is substituted with three smaller faces. Therefore, the amount of faces is increased by two and Eulers Formula holds.

If f is the outerface, then there are three vertices on the outerface, as well. Otherwise, there would be edges addable to G , and G is not planar. The same holds for this case. \square

Fact 3.

Let f be a face with an area consumption of $\mathcal{O}(a(k))$, $a(k)$ monotone increasing function and k describing the granularity of the underlying grid. If the grid is refined by a factor of n for each dimension, then f consumes area of $\mathcal{O}(n^2 \cdot a(k))$.

Fact 4.

Let f be a face of a drawing Γ_G with a constant amount of vertices defining it, inheriting area $\Theta(f(n))$. If G is extended in a way that a new vertex, connected to all the vertices defining f , is placed inside f so that f is divided evenly, the resulting faces of the supergraph have still area $\Theta(f(n))$.

Proof. Let a be the area consumption of a face f , $a \in \Theta(f(n))$. Since the face is divided evenly by a constant amount c , every new face consumes at most $\frac{a}{c} + \epsilon$ area ($\epsilon > 0$).

$$a \in \Theta(f(n)) \Rightarrow \frac{a}{c} + \epsilon \in \Theta(f(n))$$

\square

Fact 5 (Zig-zag elongation).

Let A be a bounding box with area consumption $\mathcal{O}(a(k))$, where k describes a granularity of the underlying grid. The height and the width is described as $\mathcal{O}(h(k))$, $\mathcal{O}(w(k))$, respectively.

A poly-line p can be drawn inside A with length $\mathcal{O}(a(k))$, using $\mathcal{O}(\min\{h(k), w(k)\})$ bends.

Proof. Let w.l.o.g. the height be larger than the width. The other case is valid analogously. The bend points are placed at the top and bottom row along the width at the bounding box. Start at left corner of A and draw a line segment along the height. For the following line segments, alternate with bend points between the top

and bottom row until a right corner of A is reached. The amount of bends are at most two times the width length. The length of the poly-line p is therefore at most:

$$\text{len}(p) \leq (2 \cdot w(k) + 2) \cdot h(k) \in \mathcal{O}(w(k)h(k))$$

□

Fact 6.

Following Fact 5, $\mathcal{O}(\max\{h(k), w(k)\})$ bends can be used to increase the length of a poly-line p up to the area usage of a bounding box A .

Fact 7.

TODO: The same holds for a face of arbitrary shape with area usage $\mathcal{O}(a(k))$ in a drawing of a graph. I miss the idea of how to proof it.

Theorem 1.

Let G be a maximal planar graph, admitting a planar straight-line grid drawing Γ_G with area consumption $\mathcal{O}(n) \times \mathcal{O}(n)$. Let $f(n)$ be a monotone function which describes the amount of bends per edge allowed. If $f(n) \in \mathcal{O}(n)$, and the grid is refined by a factor of n , then Γ_G can be altered to a poly-line drawing Γ'_G such that the edge-length ratio lies in $\mathcal{O}(1)$ in area $\mathcal{O}(n^2) \times \mathcal{O}(n^2)$.

Proof. Let G be a maximal planar graph with n vertices. Obtain a straight-line planar drawing of G with help of an arbitrary drawing algorithm guaranteeing an area consumption of $\mathcal{O}(n) \times \mathcal{O}(n)$. Then, the edge-length ratio lies in $\mathcal{O}(n)$ since the shortest edge lies in $\mathcal{O}(1)$ and the longest edge lies in $\mathcal{O}(n)$.

Refine the grid by a factor of n in each dimension to obtain Γ_G in area $\mathcal{O}(n^2) \times \mathcal{O}(n^2)$. Then, by Fact 3, every inner face consumes at least area of $\mathcal{O}(n^2)$ since any inner face of the original straight-line drawing consumes at least $\mathcal{O}(1)$ area. By Fact 1, every face of G is a triangle.

For each inner face f , a vertex is inserted and connected to each of the three vertices defining f . Mark the new vertices and edges. These new vertices are placed in Γ_G so that the according face is evenly subdivided. This extension to the supergraph G' is by Fact 2 again a maximal planar graph with and since G has $2n - 5$ inner faces, G' has now $3n - 5$ vertices and $6n - 15$ faces. G has $3n - 6$ edges, there are 3 edges on the outerface and $3n - 9$ inner edges. Every edge of G is adjacent to two faces and enclosed by exactly four marked edges in G' . By Fact 4, the area enclosed by these marked edges is at least $\mathcal{O}(n^2)$ big and by construction, this area is free in Γ_G . By Fact 7, every edge drawn in Γ_G can be elongated “zig-zag wise“ by a factor of $f(n)$ bends used within the marked enclosing edges. For an edge of length $\mathcal{O}(n)$ in Γ'_G , $\mathcal{O}(n)$ bends are mandatory to increase the length sufficiently. Not every edge needs to be elongated. The point is to elongate the edges significantly shorter than the longest edge. By leveling the length of all edges to $\mathcal{O}(n^2)$, the result is a poly-line drawing with an edge-length ratio of $\mathcal{O}(1)$ in area $\mathcal{O}(n^2) \times \mathcal{O}(n^2)$ with $\mathcal{O}(n)$ bends per edge. □

This proof implies the drawing algorithm for any maximal planar graph.

Theorem 2.

Every planar graph G admits a poly-line drawing with an edge-length ratio of $\mathcal{O}(1)$ in area $\mathcal{O}(n^2) \times \mathcal{O}(n^2)$, allowing up to $\mathcal{O}(n)$ bends per edge.

Proof. At first, add edges to G so it is a maximal planar graph G_{\max} and mark them. Then, draw G_{\max} according to the proof of Theorem 1. The output is a poly-line grid drawing in area $\mathcal{O}(n^2) \times \mathcal{O}(n^2)$ with an edge-length ratio of $\mathcal{O}(1)$ and up to $\mathcal{O}(n)$ bends per edge. Afterwards, remove all marked edges from G_{\max} to obtain the drawing for G . \square