CS 163 & CS 265: Graph Algorithms

Week 10: Planar graphs

Lecture 27: Planarity testing and planar graph drawing

David Eppstein
University of California, Irvine

Fall Quarter, 2022

Overview

Two stages of finding planar drawings

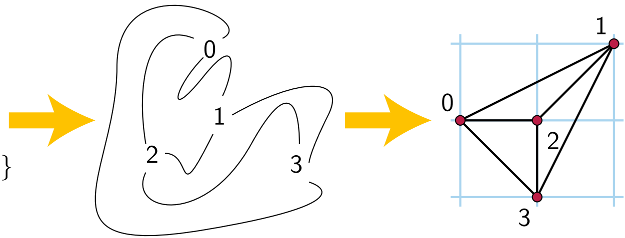
Stage 1: Find a topological drawing

- No coordinates for vertices
- Describe the ordering of the edges around each vertex
- Allow curved edges
- Today's lecture

Stage 2: Straighten the drawing

- Fáry's theorem: always possible [Wagner 1936; Fáry 1948; Stein 1951]
- Give coordinates for the vertices (small integers)
- Draw edges as straight line segments
- Next time

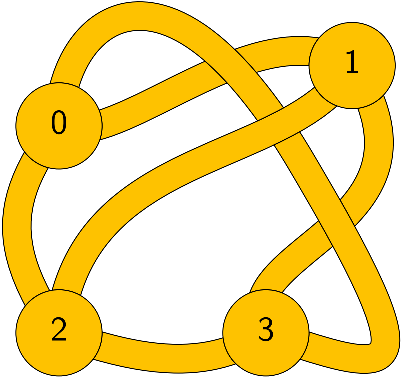
{ 0: [1,2,3],
1: [0,2,3],
2: [0,1,3],
3: [0,1,2] }



How to represent a topological drawing?

Think of the vertices as flat coins and the edges as (untwisted) ribbons linking them together

We will fill in the holes to make a surface

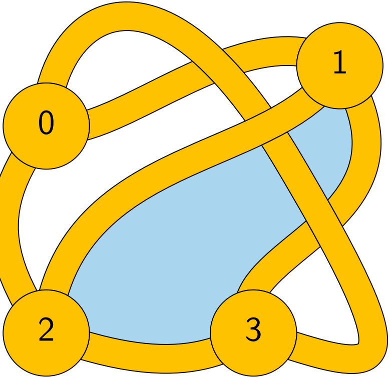


All we need to know is the cyclic ordering of the ribbons attached to each coin (a specially ordered adjacency list of the graph)

Filling in the holes

(Conceptually, not in the computer) construct disk-shaped patches of surface to connect to the boundaries of the ribbons and coins

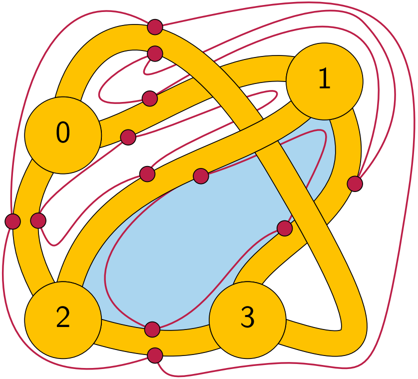
Glue them all together to make a surface



Don't worry about above/below relationships shown in this drawing (they're not part of the actual surface)

Finding the faces

Faces are connected components of an auxiliary graph with
vertices = boundaries of ribbons (two per ribbon),
edges = two ribbon boundaries next to each other on the same coin

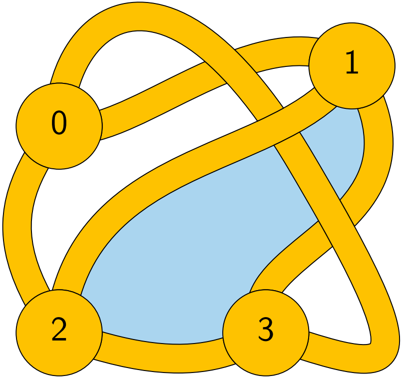


So if we already know cyclic ordering of ribbons around each coin,
we can build this graph and use it to find all faces in linear time

How to tell when it's planar?

Every cyclic ordering of edges around each vertex describes an embedding of the graph onto a topological surface, but that surface might not be the plane

Trace and count the faces (this one has four triangles)



It's planar if and only if $V - E + F = 2$

What we want an algorithm to do

Input: A graph that we think might be planar

Output: A cyclic ordering of neighbors around each vertex (represented as an ordered adjacency list structure)

Such that tracing faces and counting gives $V - E + F = 2$

This output will be the input to a different algorithm for finding vertex coordinates of a straight drawing

An error condition if the graph is not planar

Or maybe a subdivision of K_5 or $K_{3,3}$

Algorithms

Some history of planarity algorithms

First linear time algorithm by [Hopcroft and Tarjan 1974], winners of the 1986 Turing Award for their work on efficient algorithms

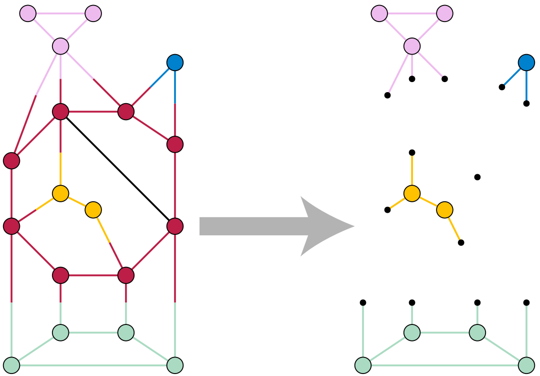
Other linear time methods include Booth and Lueker [1976] (Lueker is a retired UC Irvine professor), Boyer and Myrvold [2004], and de Fraysseix et al. [2006]

They are all quite complicated

Instead I'll describe a slower but simpler algorithm originally published by Auslander and Parter [1961]

The flaps of a cycle

Given a cycle in a graph (such as the red cycle below), split each edge that touches the cycle but is not part of it into a two-edge path, then remove the cycle vertices



Flaps are components of resulting graph (including single-vertex components for chords of the cycle)

Main idea of algorithm

Divide and conquer:

Find a cycle with more than one flap

Check that all flaps are compatible with each other

Recurse on each flap+cycle subgraph

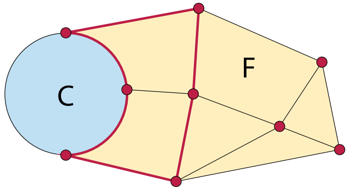
Glue the embeddings of the subgraphs into one big embedding

Finding a cycle with more than one flap

Auslander and Parter forgot this step, and just say to use any cycle
$\Rightarrow$ infinite recursion when only one flap

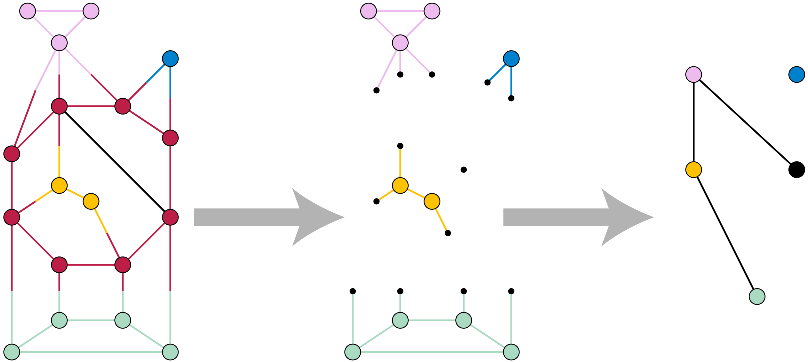
Some messy case analysis:

- Shrink edges at degree-1 and degree-2 vertices and merge multigraph edges until shrinking to nothing (in which case graph is planar) or remaining vertices have degree ≥ 3
- Find any cycle C, and if it has > 1 flaps return it
- Remaining case: one flap F touches all vertices of C. Replace an edge of C by path through $F \Rightarrow$ cycle with > 1 flaps



Compatibility of flaps

Draw a graph with a vertex for each flap, and an edge from flap X to flap Y when the cycle has four vertices p, q, r, and s (in cyclic order) with X touching p and r and Y touching q and s



This graph must be bipartite, with bipartition separating flaps drawn inside the cycle from flaps drawn outside

Analysis

Slowest part of algorithm: Building and testing bipartiteness of flap compatibility graphs

In a single recursive call, this can be done in time proportional to # of pairs of edges that belong to different flaps

There are $O(n^2)$ pairs of edges (whole graph has $\leq 3n - 6$ edges), each contributing to the time for a single recursive call

Other parts of the algorithm take $O(n)$ time/call, and $O(n)$ calls

$\Rightarrow$ Total time for whole algorithm is $O(n^2)$

Morals of the story

Separation of planar drawing algorithms into two stages, topological and geometric

Output of topological stage is a description of the ordering of neighbors around each vertex

Can be solved in linear time and has a simpler $O(n^2)$ algorithm

References I

L. Auslander and S. V. Parter. On imbedding graphs in the sphere. *Journal of Mathematics and Mechanics*, 10(3):517–523, 1961. URL `https://www.jstor.org/stable/24900736`.

Kellogg S. Booth and George S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using *PQ*-tree algorithms. *Journal of Computer and System Sciences*, 13(3):335–379, 1976. doi:`10.1016/S0022-0000(76)80045-1`.

John M. Boyer and Wendy J. Myrvold. On the cutting edge: simplified $O(n)$ planarity by edge addition. *Journal of Graph Algorithms and Applications*, 8(3):241–273, 2004. doi:`10.7155/jgaa.00091`.

Hubert de Fraysseix, Patrice Ossona de Mendez, and Pierre Rosenstiehl. Trémaux Trees and Planarity. *International Journal of Foundations of Computer Science*, 17(5):1017–1030, 2006. doi:`10.1142/S0129054106004248`.

References II

István Fáry. On straight-line representation of planar graphs. *Acta Sci. Math. (Szeged)*, 11:229–233, 1948.

John Hopcroft and Robert E. Tarjan. Efficient planarity testing. *Journal of the ACM*, 21(4):549–568, 1974. doi:10.1145/321850.321852.

S. K. Stein. Convex maps. *Proceedings of the American Mathematical Society*, 2(3):464–466, 1951. doi:10.2307/2031777. URL https://www.jstor.org/stable/2031777.

Klaus Wagner. Bemerkungen zum Vierfarbenproblem. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 46: 26–32, 1936. URL https://www.digizeitschriften.de/index.php?id=resolveppn{&}PPN=GDZPPN002131633.