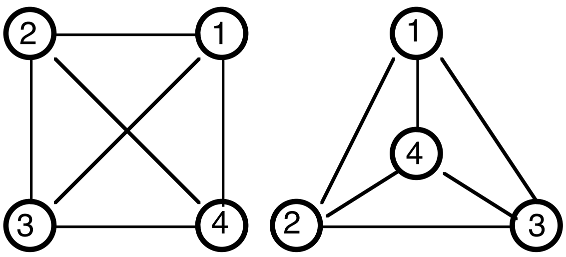
Ch4 Graph theory and algorithms

This chapter presents a few problems, results and algorithms from the vast discipline of Graph theory. All of these topics can be found in many text books on graphs.

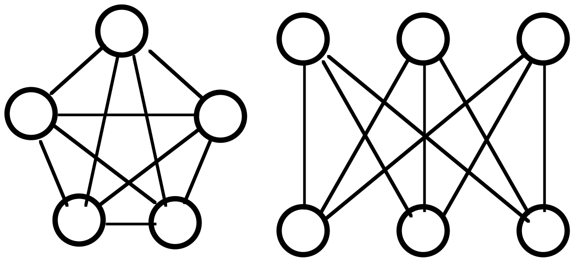
Notation: G = (V, E), V = vertices, E = edges, |V| = n, |E| = m. Edges can be symmetric of directed (arcs). Weighted graph G = (V, E, w), w: E -> Reals. We omit other variations. e.g. parallel edges or self-loops.

4.1 Planar and plane graphs

Df: A graph G = (V, E) is **planar** iff its vertices can be embedded in the Euclidean plane in such a way that there are no crossing edges. Any such embedding of a planar graph is called a **plane or Euclidean graph.**



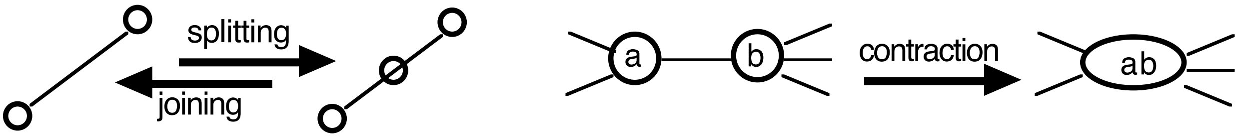
The complete graph K4 is planar



K5 and K3,3 are **not** planar

Thm: A planar graph can be drawn such a way that all edges are non-intersecting straight lines.

Df: graph editing operations: edge splitting, edge joining, vertex contraction:

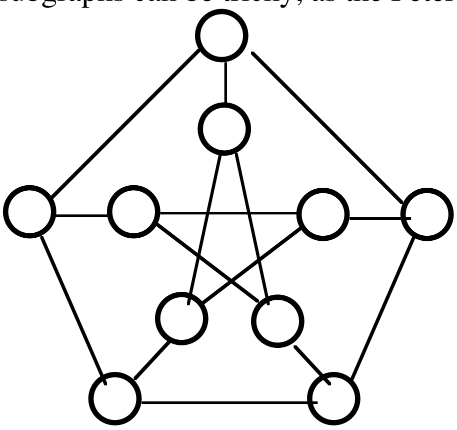


Df: G, G' are homeomorphic iff G can be transformed into G' by some sequence of edge splitting and edge joining operations.

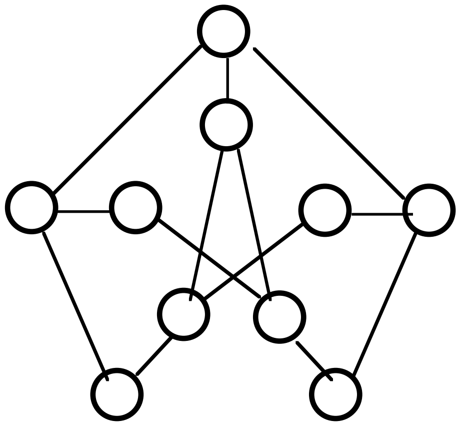
Thm (Kuratowski 1930): G is planar iff G contains no subgraph homeomorphic to K5 or K3,3.

Thm (Wagner 1937): G is planar iff G contains no subgraph contractable to K5 or K3,3.

Ex: Finding subgraphs can be tricky, as the Petersen graph shows:



Left: The Petersen graph is easily seen to be contractable to K5



Right: After removal of 2 edges followed by edge joining, the Petersen graph is seen to contain K3,3

4.2 Euler's formula for plane graphs

A **plane** graph (i.e. embedded in the plane) contains **faces**. A **face** is a connected region of the plane bounded by edges. If the graph is connected, it is said to contain a single **component**. If it is disconnected it has several components. Let |V|, |E|, |F|, |C| denote the number of vertices, edges, faces, components, respectively.

Thm (Leonhard Euler): $|V| - |E| + |F| = 2$ for a connected graph, or more generally: $|V| - |E| + |F| - |C| = 1$

Pf (of the general formula for graphs that may be disconnected) by induction on |E|.
Basis |E| = 0. Without any edges, a plane graph consists of n disconnected vertices each of which is a components, and a single face: $|V| - |E| + |F| - |C| = n - 0 + 1 - n = 1$.
Induction step: Assume Euler's formula is correct for all graphs with |E| = k, and consider an arbitrary graph G with k+1 edges. Choose any edge e in G, delete e to obtain a clipped graph G', and distinguish 2 cases:

a) e is on the boundary of 2 distinct faces of G, f1 and f2. By deleting e, we lose1 edge and 1 faces, since the former faces f1 and f2 are merged into a single face. **The quantity - |E| + |F| remains unchanged.**

b) e is on the boundary of a single face f of G. By deleting e, we lose1 edge and we gain 1 component, since the former component that contained e disconnects into 2 components. **The quantity - |E| - |C| remains unchanged.**

Since Euler's formula holds for the clipped graph G' by induction hypothesis, and the deletion of e keeps the quantity $|V| - |E| + |F| - |C|$ unchanged, Euler's formula holds also for G.

Thm (the number of edges in a planar graph grows at most linearly with the number of vertices):
G planar, $|V| \geq 3$ -> $|E| \leq 3|V| - 6$

Pf: Consider any embedding of G in the plane. If this embedding contains faces "with holes in them", add edges until **every face becomes a polygon bounded by at least 3 edges**. Proving an upper bound for this enlarged number |E| obviously proves it also for the smaller number of edges originally present. With respect to such an embedding, any **edge e bounds 2 distinct faces**.
Hence: # of incidences (edge e, face f) $= 2|E| \geq 3|F|$.
Together with Euler's formula (*3): $3|V| - 3|E| + 3|F| = 6$ we obtain $|E| \leq 3|V| - 6$.

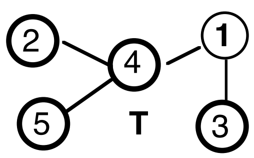
4.3 Enumerating all the spanning trees on the complete graph Kn

Cayley's Thm (1889): There are n^{n-2} distinct labeled trees on $n \geq 2$ vertices.

Ex n = 2 (serves as the basis of a proof by induction): 1---2 is the only tree with 2 vertices, $2^0 = 1$.

The most elegant proof of Cayley's Thm is based on Prüfer's coding scheme (1918): it establishes a 1-to-1 correspondence between the set of labeled trees on n vertices and the set of n^{n-2} vectors of length n-2, whose entries are labels chosen from { 1, 2, .. , n }.

Ex: The tree T at left is coded using the form shown in the middle, and filled out at right. T's code is 4 1 4.



steps	1	2	3	4 = n-1
hinges				5 = n
leaves				

#	1	2	3	4
H_i	4	1	4	5
L_i	2	3	1	4

code (Tn): for i <- 1 to n-1 do (L_i <- remove the currently least leaf; Hi <- the former neighbor of Li)
return [$H_1, H_2, .. , H_{n-2}$]

decode ([$H_1, H_2, .. , H_{n-2}$] :
H_{n-1} <- n
for i <- 1 to n-1 do L_i <- the least vertex NOT in $\{ L_1, .. , L_{i-1} \} \cup \{ H_i , .. , H_{n-1} \}$
return T <- { (L_1, H_1), (L_2, H_2), .. , (L_{n-1}, H_{n-1}) }

The proof that Prüfer's code establishes a 1-to-1 correspondence is by induction on n. Cayley's Thm follows.