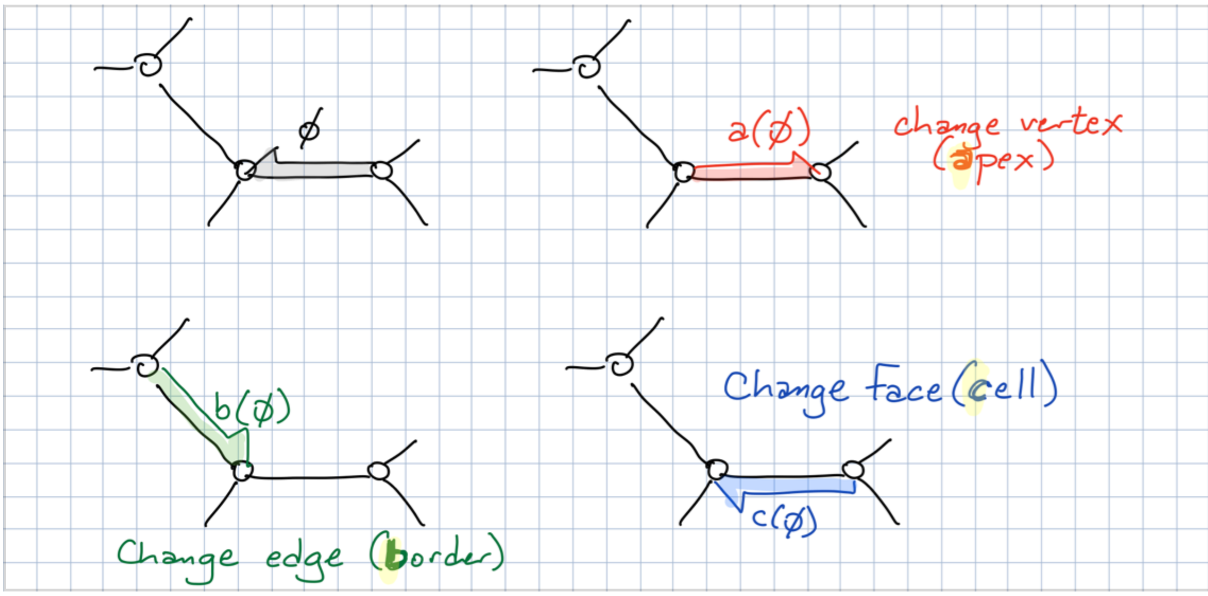
CS 7301.003.20F Lecture 14—October 5, 2020

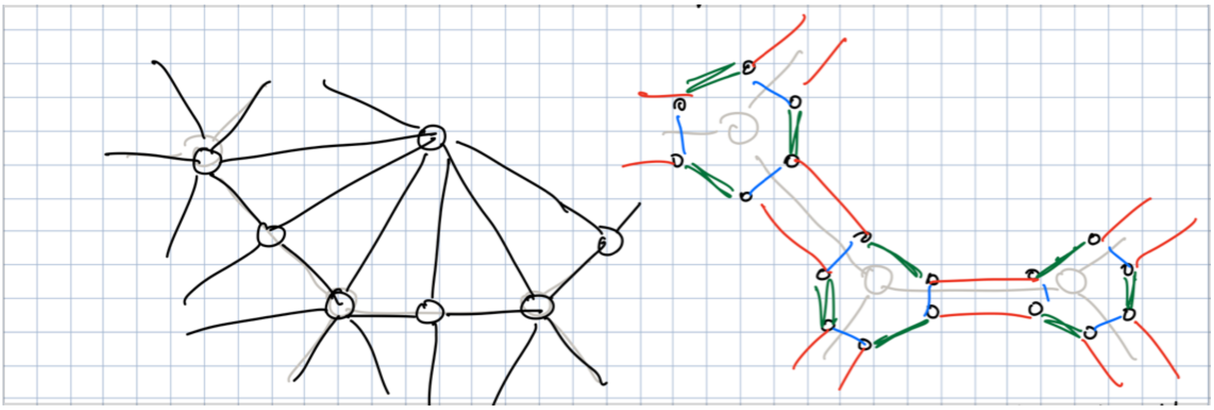
Main topics are #surface_maps.

Reflection Systems

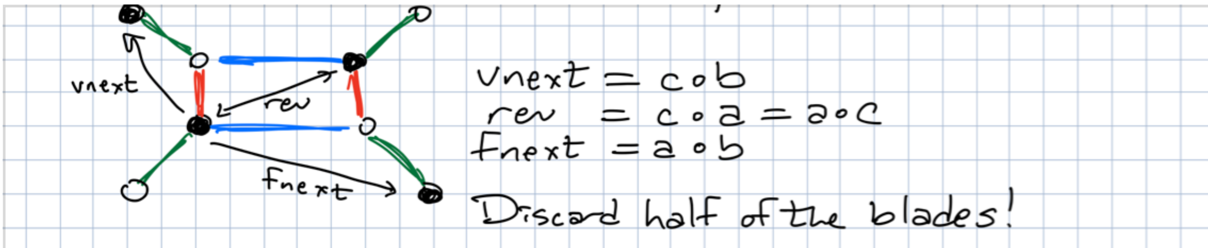
- When working with non-orientable surfaces, it's useful to describe an embedding using something called a *reflection system* (Phi, a, b, c).
- Each member of Phi is called a *blade* or *flag*.
- Each of a, b, and c is an involution on Phi such that a circ c = c circ a.



- Different orbits create the structures we're already used to:
    - Orbits of a form polygonal schema sides.
    - Orbits of b form polygonal schema corners.
    - Orbits of c form darts.
    - Orbits of permutation group <b, c> form vertices.
    - Orbits of <a, c> form edges.
    - Orbits of <a, b> form faces.
- There's two ways of thinking about the blade and a, b, c that may be a bit more intuitive.
- Blade can represent faces of the *barycentric subdivision* G^+. Here, we take G, add a node to each face and to the middle of each edge, and connect nodes of incident objects. In particular "edge nodes" all have degree 4.



- They can also represent vertices of the *band decomposition* G^box. Here, we replace each vertex of degree delta with a 2-delta-gon and replace edges with rectangles. Each involution a, b, c represent a single edge of the band decomposition.
- The dual of the reflection system can be defined as (Phi, c, b, a). Notice how G^box = (G^ *)^box, G^+ = (G^ *)^+, and (G^box)^ * = G^+.
- The reflection system can be used to describe a map on any compact surface, just like the signed polygonal schemas and rotation systems.
- I won't show the details but it turns out you can go between any one of the three to any other in O(m) time where m is the total complexity of the system.
- Finally, your surface is orientable if and only if the reflection system is bipartite, meaning Phi can be partitioned into Phi^+ and Phi^- where each involution goes between the two.
- In this case, we can even define a rotation system where
  - vnext = b circ c
  - rev = c circ a = a circ c
  - fnext = b circ a



(Yes, this image has cw and ccw mixed up so I fixed the equalities.)

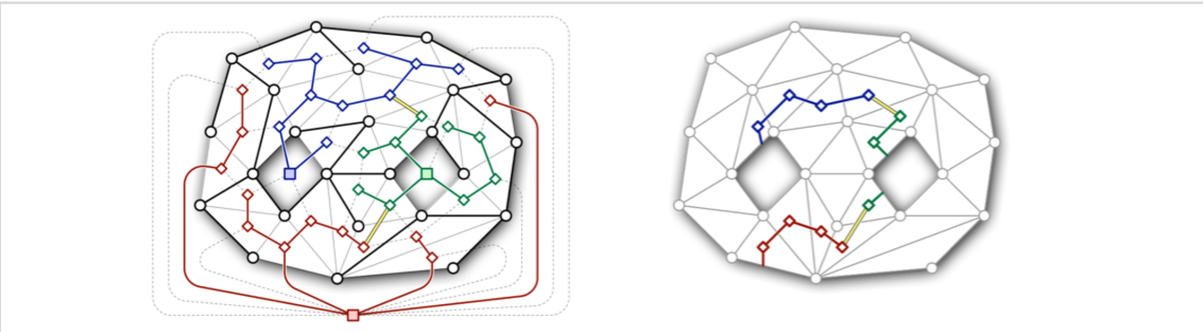
Classification

- I won't prove it during the semester, but there really aren't that many different surfaces (up to homeomorphism).
- Every compact surface is homeomorphic to some Sigma(g, h), which is the sphere with g handles + h twists, subspaces homeomorphic to the Möbius band.
- And it's orientable if and only if it is Sigma(g, 0).
- Euler's formula in full generality becomes V - E + F = 2 - 2g - h, and that implies any system of loops will have 2g + h loops.

- But actually, there are even fewer surfaces than implied by the above!
- Sigma(g, h) = Sigma(g - 1, h + 2) if h > 0.
- Implying every surface is either the orientable Sigma(g, 0) for some g ≥ 0 or the non-orientable Sigma(0, g) for some g > 0.
- Either way, g is the *genus* of the surface as defined earlier. We also say the 2g + h part of Euler's formula is the *Euler genus*.

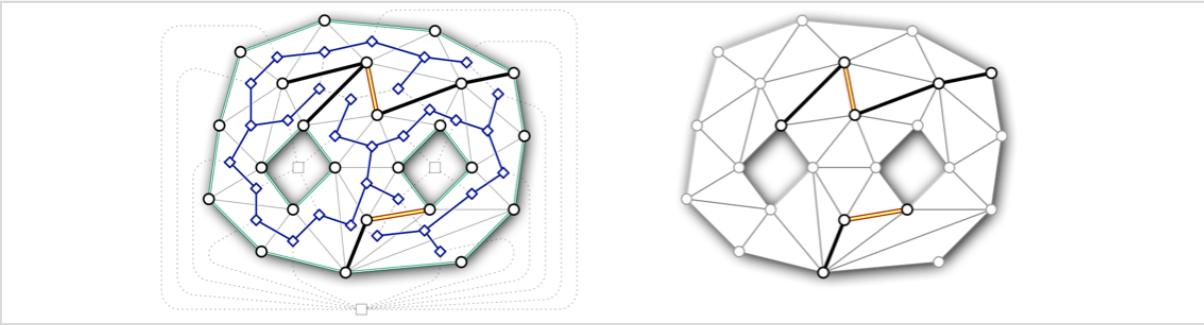
Surface with Boundary

- Beyond non-orientability, there's one more way to generalize compact 2-manifolds.
- Normally, the neighborhood around points look homeomorphic to the plane, but in a *surface with boundary*, there are some neighborhoods homeomorphic to the half plane. Imagine cutting open holes out of the surface.
- Combinatorially, we represent boundary components by just marking a disjoint subset of faces as "gone". In the dual graph, these "gone" dual vertices look more like punctures.
- Now recall how we can decompose the edges in a surface without boundary into three sets (T, C, L) where T and C are primal and dual spanning tree and |L| = 2g + h. There are actually two natural generalizations of this decomposition to surfaces with boundary, and they both have applications that we'll see next week.
- First is the *tree-coforest decomposition* (T, F, L).
  - T = spanning tree of G
  - F = spanning forest of G^* with one tree per puncture
  - L = E \ (T cup F)
  - So |L| = 2g + h + b - 1

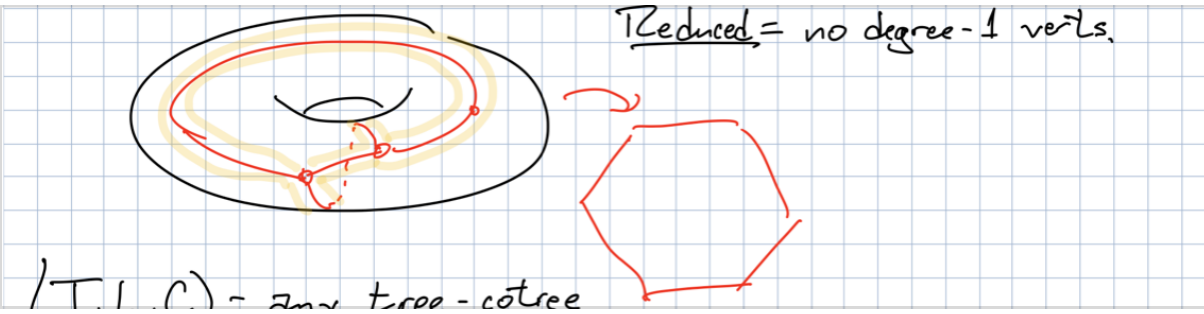


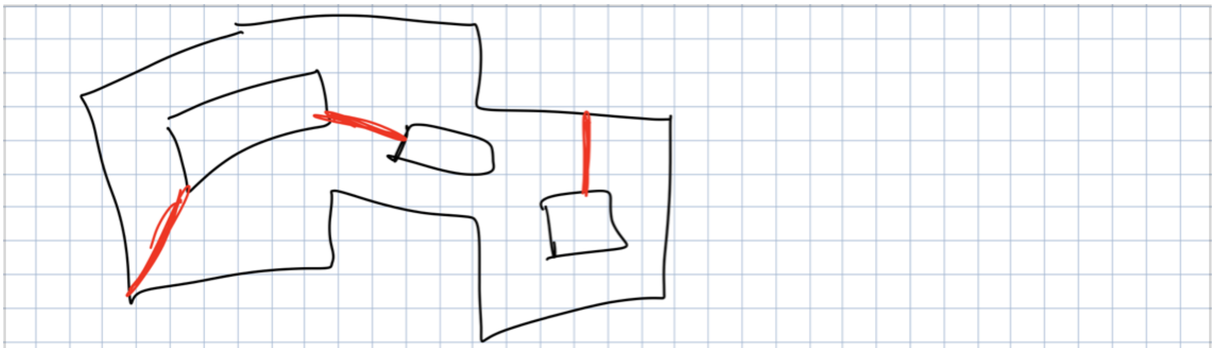
- This decomposition is useful for understanding how certain paths in the primal graph move around handles and twists. The fundamental cycle or puncture-to-puncture path created by adding any edge of L to F is analogous to n edge in a triangulated polygon with holes.
- There's also the *forest-cotree decomposition* (partial G, F, C, L).
  - partial G = boundary edges
  - C = spanning tree of G^* \ punctures
  - F = spanning forest of G with each tree containing one boundary vertex

- L = E \ (partial G cup C cup G)
- |L| = |E| - |C| - |F| - |partial G| = 2g + h + (# boundary vertices - 1) + b + (# boundary vertices) = 2g + h + b - 1



- This decomposition is useful because T cup L forms a *cutgraph*, a subgraph of G that cuts the surface into a disk (observe how the non-boundary faces are now simply connected, because C is a spanning tree of G^* \ punctures).





- Often, the most useful cutgraphs are the *reduced* ones, meaning a cutgraph with no degree-1 vertices. For example, if you want to map a texture to a surface, you can used a reduced cutgraph as your seam.
- Cutgraphs are also useful because "interesting" cycles on the surface are forced to cross them.
- You can reduce an arbitrary cutgraph (like T cup L) by iteratively removing degree-1 vertices. Think of it as trimming the "hair" off the cutgraph.