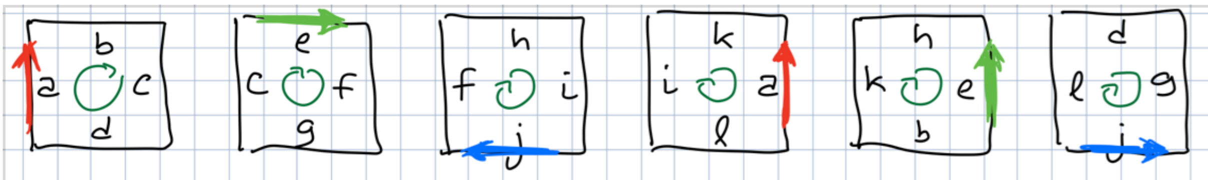
CS 7301.003.20F Lecture 13–September 30, 2020

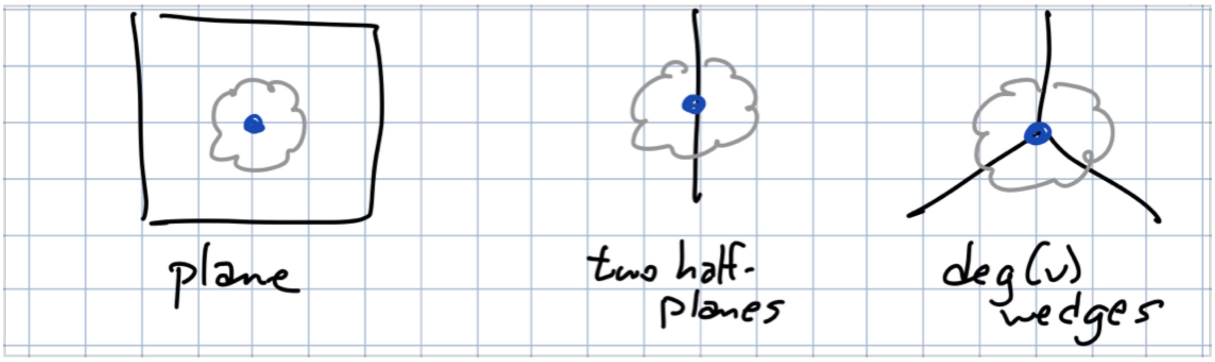
Main topics are #surface_maps.

Surface Maps

- It's finally time to leave the plane. Let's do that by going back to something we saw before.
- Recall a rotation system can be described as a triple of permutations succ, rev, and next from darts to darts (Erickson now calls these vnext, rev, and fnext to make it easier to remember which is which. Maybe we should start doing that too?)
  - rev is an involution w/o fixed points
  - fnext [i.e. next] = rev circ vnext
  - orbits of vnext go ccw around vertices at head
  - orbits of rev are edges
  - orbits of fnext go cw around faces to right
- V - E + F = 2 in connected planar graphs, but what if they equal something else?
- We still have an embedding… it's just not planar!
- Imagine every face (orbit of fnext) as a polygon. Label the sides of these polygons with the names of distinct edges. This construction is called a *polygonal schema*.
- Individual copies of vertices and edges within a polygon are called *corners* and *sides*.
- Each edge appears twice on the boundary of the polygons.
- Glue the polygons together at their corresponding edges by identifying each rev(d) with the reversal of dart d.



- If you look at a sufficiently small neighborhood around each point after gluing

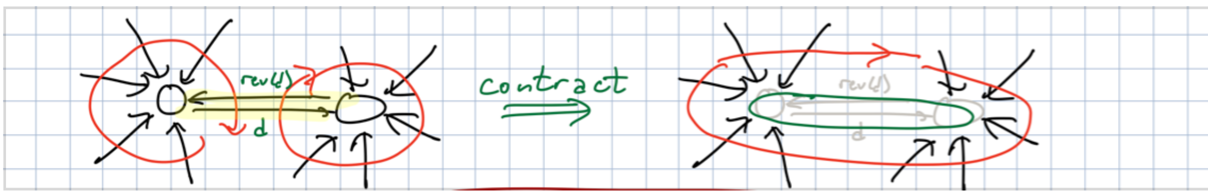


you'll see these neighborhoods are all homeomorphic to the plane.
- What we've created is an orientable compact 2-manifold. Otherwise known as a compact *surface*.

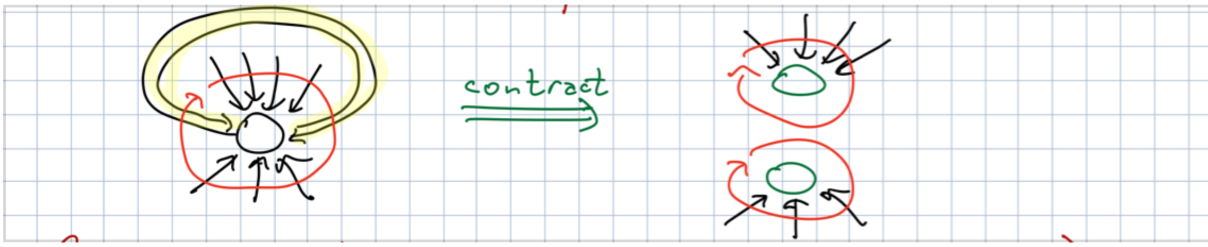
- There are infinitely many of these things, and they're distinguished (up to homeomorphism) by their genus.
- Intuitively, the genus is the number of handles you glue onto a sphere to get the surface, but sometimes it's hard to tell what the handles are.
- Formally, the *genus* is the maximum number of disjoint simple closed curves on the surface whose complement is still connected.

Trees, Co-trees, and Formulas

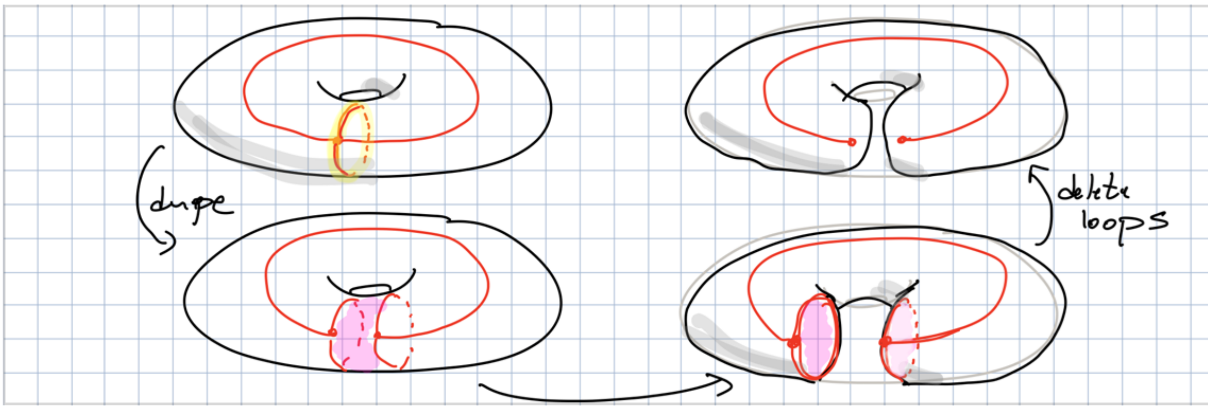
- Remember, if G is planar and T is a spanning tree, then (E \ T)^* is a spanning tree of G^*. But that's no longer true when genus > 0.
- An *isthmus* is any edge with the same face on both sides. We may have isthmuses now that are not bridges!
- We can contract any edge that's not a loop without messing with the faces. By duality, we can delete any edge separating distinct faces.
- Take any rotation system, and let g be the genus of the underlying surface.
- Suppose we contract non-loops until there's a single vertex. The V - 1 edges we contracted form a spanning tree T, but the faces remain intact.
- Now delete non-isthmuses until only one face remains. The F - 1 deleted edges form a dual spanning tree.
- But now there are L leftover edges that are all both loops *and* isthmuses. Counting the edges, we see V - E + F = 2 - L.
- These remaining edges form what is called a *system of loops*, a surface map with one vertex (the basepoint) and one face homeomorphic to an open disk.
- But how are L and g related? To find out, we're going to do something previously forbidden. We're going to contract a loop!
- Let's recall how to modify the rotation system to contract a non-loop edge.



- vnext(vprev(d)) ← vnext(rev(d)) and vprev(vnext(d)) ← vprev(rev(d))
- But if we do those operations to a loop, we split the vertex.



- And if we delete an isthmus, by duality we split a face.
- So what would this contraction look like on the actual surface?
- Well, you can't get from one side of the loop to the other anymore, so it's like we cut a handle. Here's another way to imagine what happened:



- The result is that we now have genus g - 1 but still only one face.
- The face uses both vertices, so there is at least one non-loop edge. Contract it as well.
- Now we have a new system of L - 2 loops on a surface of genus g - 1.
- Now we just need to figure out some base cases, and we'll be done.
- If L = 0, then we have the trivial map of a single vertex on a sphere.
- If L = 1, then we have two darts d and rev(d).
    - So vnext(d) = rev(d), implying fnext is the identity function. But that means we have two faces. There is no system of one loop (on an orientable surface).
- So using induction, we may conclude V - E + F = 2 - 2g.

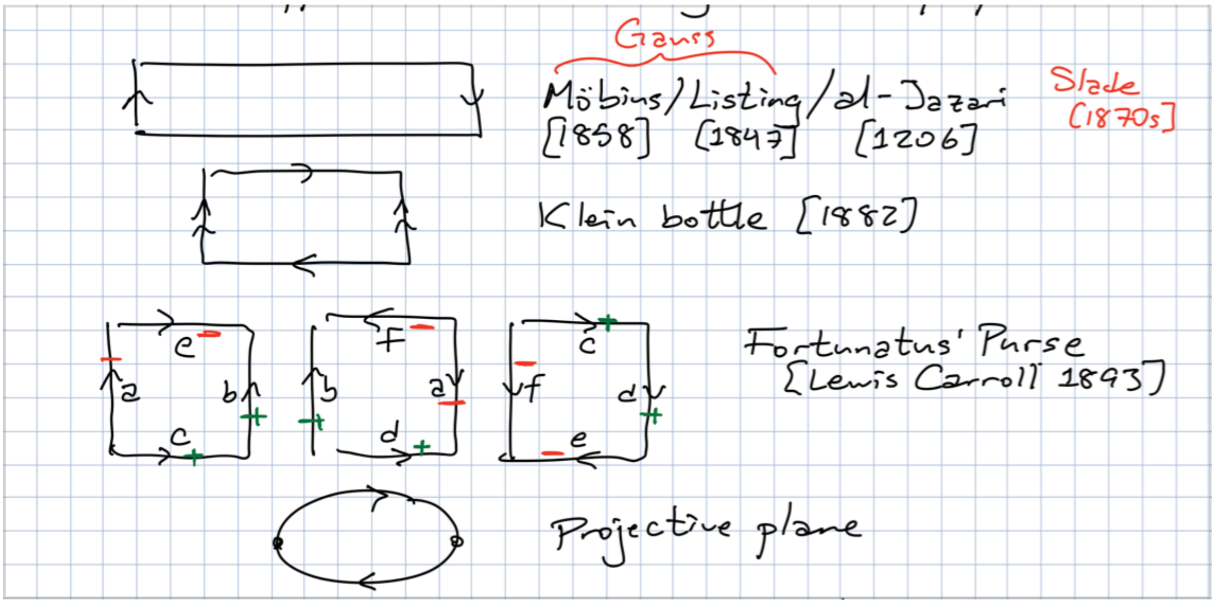
Surfaces as Systems of Loops

- It turns out you can learn almost all there is to know about your orientable surface just from polygonal schema or systems of loops.
- Kerékjártó-Radó: Any compact, connected 2-manifold can be described by a polygonal schema.
- Corollary: Any compact, connected 2-manifold can be described by a system of loops.
- Brahana: Two systems of loops with the same # loops describe homeomorphic orientable surfaces.

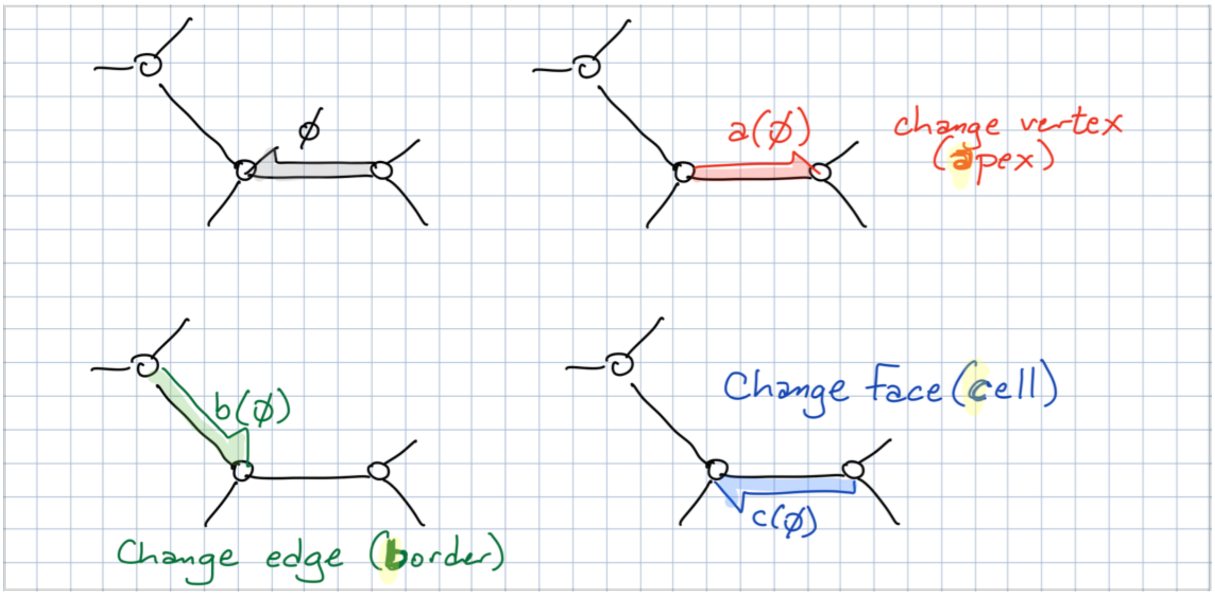
Non-orientable Surfaces

- Not every surface can be described using a mere rotation system, though.

- Suppose we take a polygonal schema but you're allowed to "turn over" the paper by flipping the edges before gluing.
- You get one of several non-orientable surfaces:



- We can represent such surfaces using *signed* polygonal schemas or rotation systems.
- For example, we could have a function fsign : E ➡ {-1, +1} where +1 means the two times you pass the edge you're going in opposite direction.
- Or you could have a vsign : E ➡ {-1, +1} where +1 means the "counterclockwise" ordering at the endpoints is consistent.
- Unfortunately, the relationship between these functions is more ugly than we'd like. In particular, there's no dual correspondence between direction (the head vs. tail of an edge) and orientation (which face lies to the left or right).
- In particular, it doesn't make sense to take the dual of a directed graph embedded on a non-orientable surface.
- In this case, it's more natural to use 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.