Math 3322: Graph Theory[1] Mikhail Lavrov

Lecture 2: Connected components

August 19, 2021 *Kennesaw State University*

1 Equivalence relations

Let G be a graph. We define a relation $\sim$ on two vertices $v, w \in V(G)$: $v \sim w$ if there is a $v - w$ walk in G. Our first plan for today is to prove:

Proposition 1.1. *The relation $\sim$ is an equivalence relation on $V(G)$.*

Before we do that, though, let's talk about equivalence relations.

You may have already seen the definition in other classes (or not). A relation R is an **equivalence relation** on a set S if it has three properties:

1. It is **reflexive**: for all $x \in S$, $x \mathrel{R} x$.
2. It is **symmetric**: for all $x, y \in S$, if $x \mathrel{R} y$, then $y \mathrel{R} x$.
3. It is **transitive**: for all $x, y, z \in S$, if $x \mathrel{R} y$ and $y \mathrel{R} z$, then $x \mathrel{R} z$.

But these are not just nice properties we want to prove because we like how they sound! There is a purpose to showing that something is an equivalence relation.

These three properties are exactly the things we need to check in order to know that we can split up (or **partition**) S into equivalence classes: disjoint sets $S_1, \dots S_k$ such that $S = S_1 \cup \dots \cup S_k$ and we have $x \mathrel{R} y$ exactly when x and y are in the same class. In the case of our relation $\sim$, the equivalence classes will give us the connected components of a graph.

Proof of Proposition 1.1. Let's check all three properties of an equivalence relation.

We check that $\sim$ is reflexive: for any vertex v, there is a $v - v$ walk in G. Well, one such walk that's guaranteed to exist is the walk of length 0 that's just the sequence (v). (There may or may not be others.)

We check that $\sim$ is symmetric: if there is a $v - w$ walk in G, there is also a $w - v$ walk. Well, suppose that $(v_0, v_1, v_2, \dots, v_\ell)$ is a $v - w$ walk (with $v_0 = v$ and $v_\ell = w$). Then reverse it: $(v_\ell, v_{\ell-1}, v_{\ell-2}, \dots, v_0)$ is a $w - v$ walk. It starts at w, ends at v, and consecutive vertices in the sequence are adjacent, because they were also consecutive in the $v - w$ walk.

Finally, we check that $\sim$ is transitive. Suppose u, v, w are vertices in G such that there is a $u - v$ walk $(x_0, x_1, \dots, x_\ell)$ and a $v - w$ walk $(y_0, y_1, \dots, y_m)$. We know $u = x_0$, $v = x_\ell = y_0$, and $w = y_m$. Then there is also a $u - w$ walk: the sequence

$$(x_0, x_1, \dots, x_\ell, y_1, y_2, \dots, y_m).$$

[1] This document comes from the Math 3322 course webpage: http://facultyweb.kennesaw.edu/mlavrov/courses/3322-fall-2021.php

This definitely starts at $u = x_0$ and ends at $w = y_m$. All consecutive vertices are adjacent, because they were already consecutive in the two walks we started with, with the exception of one pair we need to look at more closely: x_ℓ and y_1. These are adjacent because $x_\ell = v = y_0$, and $y_0 y_1$ is an edge because $(y_0, y_1, \dots, y_m)$ is a walk.

Having checked all three properties, we know that $\sim$ is an equivalence relation. □

2 Connected components

There's two ways to think of connected components of a graph: as sets of vertices, and as subgraphs. We don't know what subgraphs are yet, so let's table that one for now; define a **connected component** of G to be a set $C \subseteq V(G)$ such that

- Whenever $v, w \in C$, there is a $v - w$ walk in G;
- If $v \in C$ but $w \notin C$, there is no $v - w$ walk in G.

The relation $\sim$ is an equivalence relation on $V(G)$, so we can partition $V(G)$ into equivalence classes: $V(G) = C_1, C_2, \dots, C_k$. Look at what it means to be an equivalence class of $\sim$, and at what it means to be a connected component. We conclude:

Theorem 2.1. *The vertices of any graph can be partitioned into connected components.*

In particular, if there is only one connected component—if there is a walk from any vertex to any other vertex—then we say the graph is **connected**.

Why is this interesting? There are two reasons:

- When we were asking the question "which vertices v, w have a walk between them?" the answer could have been, in principle, a disorganized $n \times n$ table of yeses and nos (for an n-vertex graph).

  Now we know that the answer can be given much more concisely: we can just say what the connected components are (in a list of length n) and be able to answer the question for any two vertices.

- Connected components of a graph basically don't interact with each other. A lot of the time, if we're asking a question about graphs, we can work with each connected component separately.

  For example, if we're looking for a circuit board layout, and our circuit has multiple connected components, then we can just find a layout of each component separately. Then, put those components in different places on the circuit board.

  Or suppose we're looking for the largest independent set: a set of vertices with no edges between them. (This came up in the tiling problem from the previous lecture.) We can just find the largest independent set inside each component, then take their union.

3 Paths, distances, and cycles

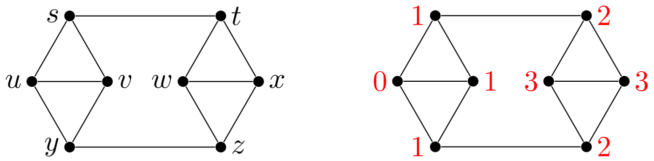
In the previous lecture, we already saw the definition of a $v - w$ path: it is a $v - w$ walk with no repeated vertices. When is there a path from one vertex to another? The answer is simple:

Theorem 3.1. *Let v, w be two vertices of a graph G. If there is a $v - w$ walk in G, then there is a $v - w$ path in G, as well. Moreover, the shortest $v - w$ walk is always a path.*

We will return to this theorem (how to prove it, and how to use it) in the next lecture.

Recall that the length of a walk $(v_0, v_1, \dots, v_\ell)$ is ℓ. The **distance** between vertices v and w (sometimes written $d(v, w)$) is the length of the shortest $v - w$ walk, which we now know is also the length of the shortest $v - w$ path. If v and w are in different connected components, then there is no such walk, and we sometimes say that $d(v, w) = \infty$ in that case.

For example, consider the following graph, in which every vertex has been labeled with its distance from u:



If you wanted to find all these distances, you could imagine the following procedure. Start by "exploring" all the edges out of u, and labeling their other endpoints with a 1. Then, explore all the edges out of those vertices: if they go somewhere new we haven't been before, label those vertices with a 2. Then, explore from the vertices labeled 2, and so on, until the entire graph has been visited.

The **diameter** of a graph is the longest distance between any two vertices. In the above graph, the diameter is 3, though that's not apparent just from the picture, which only includes distances involving u.

Another kind of walk that's important to mention is a **closed walk**: a walk that begins and ends at the same vertex. These can be very boring: for example, (u) by itself is a closed walk, and so is the walk $(s, t, s, t, s, t, s, t, s, t, s)$.

We define a **cycle** to be something more interesting, a closed walk that:

- Does not repeat any vertices, except at the start and end.
- Has length at least 3, to exclude examples like (u) and (s, t, s).

In the graph above, (s, u, v, s) is a cycle.

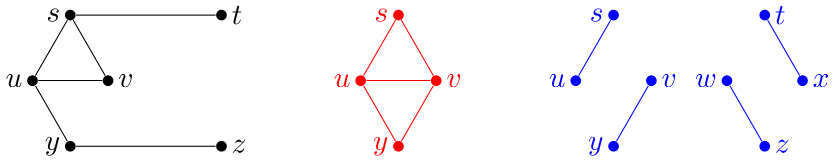
4 Subgraphs

When discussing connected components earlier, I mentioned subgraphs.

A subgraph H of a graph G is simply a graph such that:

- Its vertices $V(H)$ are a subset of the vertices $V(G)$. (Possibly all of them, and possibly just some.)
- Its edges $E(H)$ are a subset of the edges $E(G)$. (Possibly all of them, and possibly just some—but keep in mind that we can't include an edge of G unless we've included both of its vertices.)

Here are a few example subgraphs of the graph from the previous section:



We don't have to include all the vertices. If we include both endpoints of an edge of G, we don't have to include that edge.

However, if the vertices of H are a subset $S \subseteq V(G)$, and the edges of H are *all* the edges of G between vertices in S, then H is called an **induced subgraph**, or the subgraph of G **induced by** S. This is sometimes written $G[S]$. Essentially, the induced subgraph is the "most natural" subgraph to take with the vertices on S.

Among the examples above, the middle one is the only induced subgraph: it could be written as $G[\{s, u, v, y\}]$.

When we defined connected components earlier, we partitioned the vertices $V(G)$ as $C_1 \cup C_2 \cup \cdots \cup C_k$. Sometimes, we also define the connected components as the induced subgraphs

$$G[C_1], G[C_2], \dots, G[C_k].$$

These are connected subgraphs of G that are "as large as possible": you can't make larger connected subgraphs by including more vertices.

5 Drawing graphs on computers (optional)

There's lots of tools out there that will draw graphs for you. If you use a computer algebra system like Mathematica, Maple, or MATLAB, you might already know how to use it to draw graphs.

If you don't use any of those, I've looked around and found Graphviz, which has the following advantages:

- It is not too hard to use to do the basic things: I can get you started today.
- You can do fancier things if you look things up in the documentation (or ask me).
- You can use it through your browser, at http://www.webgraphviz.com/.

For the basics, here is code that you can paste into the online interface and generate the graph of the three cups puzle from the previous lecture:

```
graph {
  UUU -- UDD; UUU -- DDU; UDD -- DUD; DDU -- DUD;
  DDD -- DUU; DDD -- UUD; DUU -- UDU; UUD -- UDU;
}
```

Just use `vertex1 -- vertex2;` to draw an edge between two vertices. Graphviz automatically figures out what the vertices are based on the edges you put in; you can write `vertex;` all by itself if you want to add an isolated vertex with no edges out of it.

I don't mind you using this on homework to draw graphs, but you should make sure that you *can* do it yourself if you need to; you won't have this tool when you're taking exams.