Math 3322: Graph Theory[1] Mikhail Lavrov

Lecture 16: Directed graphs and multigraphs

October 10, 2024 *Kennesaw State University*

1 Introduction to multigraphs

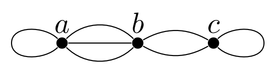
Today, we will generalize our notion of graphs in two ways: to multigraphs and to directed graphs.

First, a note on terminology. I will continue to assume that the word "graph" refers to the same notion we've discussed in all previous lectures. (If necessary for emphasis, I will say that a graph is **simple** if it is not a multigraph, and **undirected** if it is not a directed graph.) If we are considering either of these new notions, I will warn you!

With that said, on to multigraphs. A multigraph is a graph that allows the following two things:

- Loops: edges that join a vertex to itself.
- Parallel edges: two or more edges with the same endpoints.

For example, the following graph is a multigraph with loops at a and c, three parallel edges between a and b, and two parallel edges between b and c.



Formally, in a multigraph, edges are no longer simply 2-element subsets of V. We can think of a multigraph G as a triple of three pieces of information: its vertex set $V(G)$, its edge set $E(G)$, and a relationship between these, in which each edge $e \in E(G)$ has two vertices v, w (possibly equal) called its endpoints.

Of course, this formulation is rich enough to encompass simple graphs as well, but it's too cumbersome to introduce right away when multigraphs are not needed.

Many notions we've studied for graphs still make sense for multigraphs, with some modifications. Here is a summary that covers everything we've done:

- A walk in a graph is no longer just a sequence of vertices: it matters *how* we get from one edge to another. We say that a $v - w$ walk of length k is a sequence

$$(v_0, e_1, v_1, e_2, v_2, \dots, v_{k-1}, e_k, v_k)$$

  where $v = v_0$, $w = v_k$, and for each i, the endpoints of e_i are v_{i-1} and v_i.

- Paths and cycles remain the same, but it will be convenient to allow cycles of length 1 and 2. We have a cycle (v, e, v) of length 1 if e is a loop with both endpoints at v. We have a cycle (v, e, w, f, v) of length 2 if e and f are two *different* edges with endpoints at v and w.

[1]This document comes from an archive of the Math 3322 course webpage: http://misha.fish/archive/3322-fall-2024

- We need to be more specific with the degree of a vertex; $\deg(v)$ is now defined as the number of times an edge has v as an endpoint. A loop at v contributes two to the degree: we think of the loop as having two endpoints, both of which are v. This is done so that the handshake lemma still holds.
- The graphic sequence problem becomes much easier for multigraphs; see the exercises at the end of these lecture notes.
- The easiest way to define a multigraph isomorphism is to say that it is still a bijection f between the vertex sets, but modify the condition on f: for any vertices v, w (possibly equal), the number of edges between v and w should be the same as the number of edges between $f(v)$ and $f(w)$.
- Trees are always simple graphs: loops and parallel edges are cycles of length 1 and length 2, respectively. Even if you start with a multigraph, its spanning trees will be simple graphs.
- Our results about matchings apply unchanged to multigraphs. (Note that a bipartite multigraph may have parallel edges, but it cannot have loops.) In particular, it is useful to know that a regular bipartite multigraph always has a perfect matching.

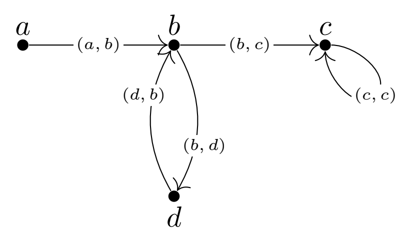
In upcoming lectures, results about Eulerian graphs and about connectivity are more general (and not any harder) if we apply them to multigraphs, so we will do this. Occasionally, multigraphs will be useful when we discuss planar graphs, but other times, we will want to consider only simple planar graphs.

2 Introduction to directed graphs

A **directed graph** or **digraph** is a generalization of a graph, meant to model asymmetric relationships. (Friendships on Facebook give us a graph; follows on Twitter give us a digraph.)

Formally, a directed graph D has a set of vertices $V(D)$ and a set of **directed edges** or **arcs** $E(D)$, which is a subset of $V(D) \times V(D)$. An arc is an ordered pair (v, w), which we might sometimes write "$v \to w$", and represents a relationship with a direction: from v to w.

We will allow loops: arcs of the form (v, v), which go from v back to v. We also allow the arcs (v, w) and (w, v) to exist in the same digraph. In diagrams, we represent arcs by arrows:



(Usually, we will not label the arcs; they are labeled here to show you how we interpret an arrow as a pair (v, w).)

We can also consider directed multigraphs, which allow multiple arcs from v to w to exist. As with undirected multigraphs, these require three objects to define: a set of vertices $V(D)$, a set of arcs $E(D)$, and a relationship which assigns every arc a start vertex and an end vertex.

2.1 Indegrees and outdegrees

In a directed graph, a vertex doesn't just have a degree. Instead, we count:

- The **indegree** $\deg^-(v)$ is the number of arcs oriented toward v. In the graph above, we have

$$\deg^-(a) = 0 \qquad \deg^-(b) = 2 \qquad \deg^-(c) = 2 \qquad \deg^-(d) = 1$$

- The **outdegree** $\deg^+(v)$ is the number of arcs oriented away from v. In the graph above, we have

$$\deg^+(a) = 1 \qquad \deg^+(b) = 2 \qquad \deg^+(c) = 1 \qquad \deg^+(d) = 1$$

A version of the degree sum formula[2] holds for directed graphs: for any digraph D,

$$\sum_{v \in V(D)} \deg^-(v) = \sum_{v \in V(D)} \deg^+(v).$$

The argument is the same: both sums are equal in a digraph with no arcs, and every arc we add will increase both sums by 1. Specifically, arc (v, w) (where v may equal w) contributes 1 to $\deg^+(v)$ and 1 to $\deg^-(w)$.

2.2 Walks, paths, and cycles

We can define a $u - v$ walk in a directed graph D in almost the same way as we did for graphs. It's a sequence of vertices $(v_0, v_1, \dots, v_\ell)$ where $v_0 = u$, $v_\ell = v$, and for every i, (v_i, v_{i+1}) is an arc; the difference is that the arc must be oriented from v_i to v_{i+1}. In other words, a walk in a directed graph must follow the arrows.

Paths and cycles are going to be very similar. However, we allow the cycle (v, v) of length 1 if there is a loop at v, and the cycle (v, w, v) of length 2 if there are arcs in both directions between v and w. In the undirected case, we did not like (v, w, v) because it used the same edge twice; here, the cycle has to use two different arcs.

We can still ask about the distance between u and v: the length of a shortest $u - v$ path. However, this is asymmetric: $d(u, v)$ might not be equal to $d(v, u)$, and one of those might not even be defined!

3 New properties of digraphs

3.1 Weak and strong connectedness

The first problem we encounter is with defining a "connected" digraph. What are we supposed to do in situations where a path exists from v to w, but not from w back to v?

There are several solutions:

[2] If the degree sum formula is called the "handshake lemma", then this formula should be called the "handshake dilemma". The dilemma is this: how can you have a directed handshake?

1. A **weakly connected** digraph is one that becomes a connected graph if we ignore the directions of the arcs. This is sometimes a useful notion, but if that's all we do, why even bother having a digraph in the first place?

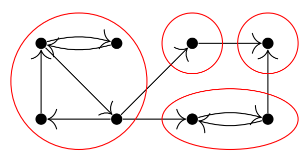
2. A **rooted** digraph has a "root vertex" u such that for all v, there is a directed $u - v$ path. This is not going to be very useful for us this semester, but comes up occasionally.

3. A **strongly connected** digraph has a directed $v-w$ path for any v and w (in either direction). This is asking for a lot, but it's very useful.

We defined connected components of graphs using equivalence relations. The relationship "there is a $v - w$ path" is no longer an equivalence relation: it is not symmetric. This complicates things.

However, the relationship "there is both a $v - w$ path and a $w - v$ path" *is* an equivalence relation. It lets us break up a directed graph into **strongly connected components** or just **strong components** for short. Within each strongly connected component, every vertex can reach every other vertex. However, if v and w are in different connected components, either there is no $v - w$ path, or no $w - v$ path, or both.

Here is an example of a directed graph being separated into strongly connected components:



Note that this picture is more complicated than splitting up a graph into connected components, since some arcs go between different strongly connected components. In particular, the strongly connected components don't tell the whole story of which vertices have a path to which other vertices...

3.2 Directed acyclic graphs

Directed acyclic graphs, sometimes abbreviated **dag**s,[3] are exactly what they sound like: directed graphs that contain no cycles. In the directed case, there cannot even be closed walks of positive length, because any such closed walk contains a cycle.

These are a sort of polar opposite of strongly connected digraphs. In a directed acyclic graph, there cannot be a $v - w$ path and a $w - v$ path at the same time, because that would form a closed walk. So there cannot be a strongly connected component containing more than one vertex. Instead, each vertex is its own strongly connected component.

If you have any directed graph that represents prerequisites (for classes you take, or for software libraries you install, or for steps in a recipe), it had better be acyclic. If there were a cycle of prerequisites between classes, then you wouldn't be able to take any of them, because each one of them would have another one you have to take first! This is one common way directed acyclic graphs show up in applications.

[3] It would probably make more sense to call them "acyclic directed graphs", but "adg" is harder to pronounce.

If you have a recipe, the steps will probably be listed in an order so that the prerequites for each step all come before it. (Sometimes there is flexibility, but you're going to have to follow the recipe in some order, anyway.) This is a general feature of directed acyclic graphs:

Theorem 3.1. *If D is a directed acyclic graph, then it has a* ***topological ordering****: an ordering $v_1, v_2, \dots, v_n$ of the vertices of D such that every arc (v_i, v_j) has $i < j$. (In a diagram where vertices $v_1, v_2, \dots, v_n$ are placed from left to right in that order, every arc will also point from left to right.)*

Proof. We induct on n, the number of vertices. If $n = 1$, then there is nothing to check.

Assume that the theorem holds for $n-1$ vertices, and let D be an n-vertex directed acyclic graph. Remember: to induct, we have to find a vertex of D to delete. We will try to delete a vertex that can be placed last in the ordering.

Start at any vertex and follow arbitrary arcs in D. If this process ever returns to a vertex we've seen before, we've found a cycle, and we assumed that there are no cycles. So in at most n steps, we get to a vertex v with $\deg^+(v) = 0$.

Now, we use the induction hypothesis. Take a topological ordering $v_1, \dots, v_{n-1}$ of $D - v$, and then add vertex v last, as v_n. There are no arcs out of v, and all arcs into v are (v_i, v_n) with $i < n$, so they obey this ordering. □

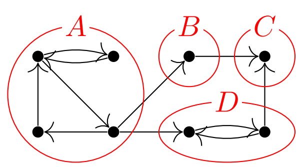
3.3 Putting them together

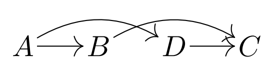
Here is one final application of directed acyclic graphs. Suppose we want to have a summary that says, for any two vertices v, w, if there is a $v - w$ path. How should we give it?

Part of the story is, of course, the strongly connected components. If v, w are in the same strongly connected component, then there are paths between them in either direction. But what if they are in different strongly connected components?

For this, we can take every strongly connected component and condense it into a single vertex. In this new directed graph, for every two distinct strongly connected components A and B, we create an arc (A, B) whenever the original directed graph has an arc (a, b) with $a \in A$ and $b \in B$: whenever it's possible to get from the first strongly connected component to the second.

Here is an example:





This graph is called the **condensation** of the original graph D.

Theorem 3.2. *The condensation of any directed graph D is a directed acyclic graph.*

Proof. Of course the condensation is a directed graph, based on the definition. How is it acyclic?

Suppose, for the sake of contradiction, that it has a cycle: strong components $S_1, S_2, \dots, S_k$ which have arcs $(S_1, S_2), \dots, (S_{k-1}, S_k), (S_k, S_1)$ in the condensation. (Note that $k \geq 2$: our definition does not include arcs from any strong component to itself.) What does that tell us about D? It tells us that we can choose vertices $x_i, y_i \in S_i$ for every i such that there is an arc (x_i, y_{i+1}) for $i = 1, \ldots, k-1$ and an arc (x_k, y_1). (These are the arcs in D that give us the arcs in the condensation that form the cycle.)

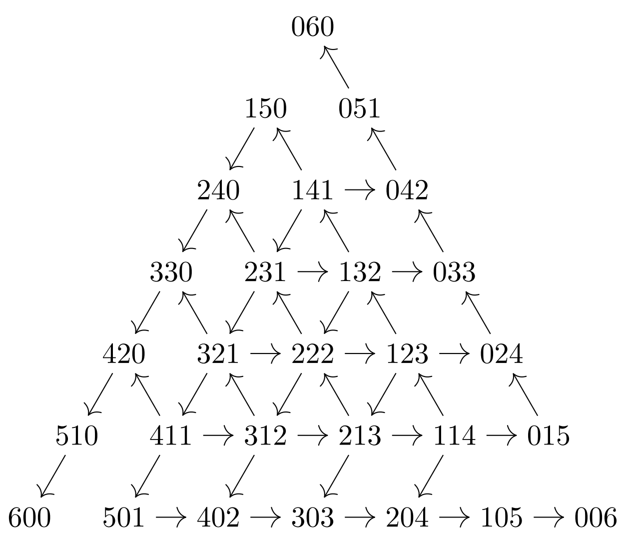
Because each S_i is strongly connected, there is also a directed path $(y_i, \ldots, x_i)$ inside S_i. Joining these paths together with the arcs we found earlier, we get a cycle that passes through all of $x_1, \ldots, x_k$ and $y_1, \ldots, y_k$.

But now, we observe that by following this cycle, we can find directed paths from each of the vertices $x_1, \ldots, x_k$ and $y_1, \ldots, y_k$ to each of the others. So all $2k$ vertices should actually be in the same strongly connected component. This is a contradiction, because we assumed there were at least two different strongly connected components involved. □

We can use the condensation as a reference to see which vertices in the original digraph can reach which other vertices. To see if there is a $v - w$ path in the original digraph, look up the strongly connected components of v and w, and see if there is a path between them in the condensation. For instance, there is an $A - C$ path (A, D, C) in the example above; therefore any vertex in strongly connected component A has a path to the vertex in strongly connected component C.

4 Example: a game with tokens

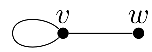
The following directed graph models a game between three players. The three players sit in a circle. There are six tokens, split up between them somehow. A player with at least one token can take a token from the player on their right. (A player with no tokens is out of the game.)



What are the strongly connected components? There's one big one: the center of the triangle. If all three players are active, then we can redistribute the tokens however we like between them. Once one player is out of the game, the outcome is predetermined: only one of the active players can take from the other, and the only possible moves involve that player taking all of the tokens, one by one.

5 Practice problems

1. Consider the multigraph below. How many closed walks of length 10 begin and end at v?



2. Draw the condensation of the graph in section 4.

3. Here is why we consider the graphic sequence problem for simple graphs, and not for multigraphs.

   (a) Let $d_1 \geq d_2 \geq \cdots \geq d_n$ be a sequence of nonnegative integers. Show that it is the degree sequence of a multigraph if and only if $d_1 + d_2 + \cdots + d_n$ is even.

   (b) Let $d_1 \geq d_2 \geq \cdots \geq d_n$ be a sequence of nonnegative integers. Show that it is the degree sequence of a multigraph *without loops* if and only if $d_1 + d_2 + \cdots + d_n$ is even and $d_1 \leq d_2 + d_3 + \cdots + d_n$.

4. Let G be a simple graph with n vertices and maximum degree r. We assume that rn is even, so that an r-regular graph on n vertices exists. (But G might not be r-regular; some of its vertices can have degree less than r.)

   (a) Prove that there is an r-regular multigraph H, with $V(H) = V(G)$, such that G is a subgraph of H. (In other words: we can add edges to G to make it regular.)

   (b) Give an example of a graph G for which the graph H in part (a) cannot possibly be a simple graph.

5. A game is played with two piles of stones. On a turn, a player picks a pile which is not empty, and takes one or more stones from it. The game ends once both piles are empty.

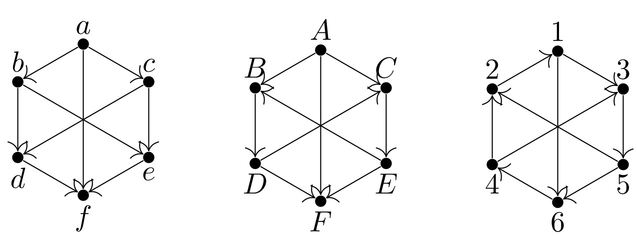
   We can represent this game by a digraph whose vertices are the states, with arcs representing the valid moves. Let $D_{n,m}$ be the digraph we get when the game is played with a pile of size n and a pile of size m.

   (a) Draw a diagram of $D_{3,2}$. (It should have 12 vertices.)

   (b) Why is $D_{n,m}$ always acyclic?

   (c) Find a topological ordering of $D_{3,2}$. (There are many.)

6. Here are three directed graphs which would be isomorphic if we forgot about the directions of the arcs. (We say that their *underlying graphs* are isomorphic.)



   Prove that as directed graphs, none of these are isomorphic.

(An isomorphism between directed graphs D and D' is a bijection $f : V(D) \to V(D')$ such that $(v, w) \in E(D)$ if and only if $(f(v), f(w)) \in E(D')$. The idea of isomorphism is the same for directed graphs: all properties of directed graphs which don't depend on how the vertices are named or how the graph is drawn should be the same for two isomorphic digraphs.)