Math 530 Spring 2022, Lecture 8: multigraphs

website: `https://www.cip.ifi.lmu.de/~grinberg/t/22s`

1. Multigraphs (cont'd)

1.1. Generalizing from simple graphs to multigraphs (cont'd)

1.1.1. Lecture 3 (cont'd)

In Lecture 3, we defined induced subgraphs of a simple graph. The corresponding notion for multigraphs is defined as follows:

Definition 1.1.1. Let $G = (V, E, \varphi)$ be a multigraph.

(a) A **submultigraph** of G means a multigraph of the form $H = (W, F, \psi)$, where $W \subseteq V$ and $F \subseteq E$ and $\psi = \varphi \mid_F$. In other words, a submultigraph of G means a multigraph H whose vertices are vertices of G and whose edges are edges of G and whose edges have the same endpoints in H as they do in G.

We often abbreviate "submultigraph" as "**subgraph**".

(b) Let S be a subset of V. The **induced submultigraph of** G **on the set** S denotes the submultigraph

$$\left(S,\quad E',\quad \varphi \mid_{E'}\right)$$

of G, where

$$E' := \{e \in E \ \mid \ \text{all endpoints of } e \text{ belong to } S\}.$$

In other words, it denotes the submultigraph of G whose vertices are the elements of S, and whose edges are precisely those edges of G whose both endpoints belong to S. We denote this induced submultigraph by $G[S]$.

(c) An **induced submultigraph** of G means a submultigraph of G that is the induced submultigraph of G on S for some $S \subseteq V$.

The infix "multi" is often omitted. So we often speak of "subgraphs" instead of "submultigraphs".

In Lecture 3, we also defined the disjoint union of two or more simple graphs. The analogous definition for multigraphs is straightforward and left to the reader.

We already defined walks, paths, closed walks and cycles for multigraphs last time. The **length** of a walk is still defined to be its number of edges. Now, let's see which of their basic properties (seen in Lecture 3) still hold for multigraphs.

First of all, the edges of a path are still always distinct. This is just as easy to prove as for simple graphs.

Next, let us see how two walks can be "spliced" together:

Proposition 1.1.2. Let G be a multigraph. Let u, v and w be three vertices of G. Let $\mathbf{a} = (a_0, e_1, a_1, \ldots, e_k, a_k)$ be a walk from u to v. Let $\mathbf{b} = (b_0, f_1, b_1, \ldots, f_\ell, b_\ell)$ be a walk from v to w. Then,

$$\begin{aligned} & (a_0, e_1, a_1, \ldots, e_k, a_k, f_1, b_1, f_2, b_2, \ldots, f_\ell, b_\ell) \\ &= (a_0, e_1, a_1, \ldots, a_{k-1}, e_k, b_0, f_1, b_1, \ldots, f_\ell, b_\ell) \\ &= (a_0, e_1, a_1, \ldots, a_{k-1}, e_k, v, f_1, b_1, \ldots, f_\ell, b_\ell) \end{aligned}$$

is a walk from u to w. This walk shall be denoted $\mathbf{a} * \mathbf{b}$.

Walks can be reversed (i.e., walked in backwards direction):

Proposition 1.1.3. Let G be a multigraph. Let u and v be two vertices of G. Let $\mathbf{a} = (a_0, e_1, a_1, \ldots, e_k, a_k)$ be a walk from u to v. Then:

(a) The list $(a_k, e_k, a_{k-1}, e_{k-1}, \ldots, e_1, a_0)$ is a walk from v to u. We denote this walk by $\operatorname{rev} \mathbf{a}$ and call it the **reversal** of $\mathbf{a}$.

(b) If $\mathbf{a}$ is a path, then $\operatorname{rev} \mathbf{a}$ is a path again.

Walks that are not paths contain smaller walks between the same vertices:

Proposition 1.1.4. Let G be a multigraph. Let u and v be two vertices of G. Let $\mathbf{a} = (a_0, e_1, a_1, \ldots, e_k, a_k)$ be a walk from u to v. Assume that $\mathbf{a}$ is not a path. Then, there exists a walk from u to v whose length is smaller than k.

Corollary 1.1.5 (When there is a walk, there is a path)**.** Let G be a multigraph. Let u and v be two vertices of G. Assume that there is a walk from u to v of length k for some $k \in \mathbb{N}$. Then, there is a path from u to v of length $\leq k$.

All these results can be proved in the same way as their counterparts for simple graphs; the only change needed is to record the edges in the walk.

1.1.2. Lecture 4

The relation "path-connected" is defined for multigraphs just as it is for simple graphs (and is still denoted $\simeq_G$). It is still an equivalence relation (and the proof is the same as for simple graphs). The following also holds (with the same proof as for simple graphs):

Proposition 1.1.6. Let G be a multigraph. Let u and v be two vertices of G. Then, $u \simeq_G v$ if and only if there exists a path from u to v.

The definitions of "components" and "connected" for multigraphs are the same as for simple graphs. The following propositions can be proved in the same way as we proved their analogues for simple graphs:

Proposition 1.1.7. Let G be a multigraph. Let C be a component of G.

Then, the induced subgraph (= submultigraph) of G on the set C is connected.

Proposition 1.1.8. Let G be a multigraph. Let $C_1, C_2, \ldots, C_k$ be all components of G (listed without repetition).

Thus, G is isomorphic to the disjoint union $G[C_1] \sqcup G[C_2] \sqcup \cdots \sqcup G[C_k]$.

The following proposition was proved for simple graphs in Lecture 4:

Proposition 1.1.9. Let G be a multigraph. Let $\mathbf{w}$ be a walk of G such that no two adjacent edges of $\mathbf{w}$ are identical. (By "adjacent edges", we mean edges of the form e_{i-1} and e_i, where $e_1, e_2, \ldots, e_k$ are the edges of $\mathbf{w}$ from first to last.)

Then, $\mathbf{w}$ either is a path or contains a cycle (i.e., there exists a cycle of G whose edges are edges of $\mathbf{w}$).

Proof. The proof of this proposition for multigraphs is more or less the same as it was for simple graphs, with a mild difference in how we prove that the walk $(w_i, w_{i+1}, \ldots, w_j)$ is a cycle (of course, this walk is no longer $(w_i, w_{i+1}, \ldots, w_j)$ now, but rather $(w_i, e_{i+1}, w_{i+1}, \ldots, e_j, w_j)$, because the edges need to be included).[1] $\square$

Just as for simple graphs, we get the following corollary:

[1]Here are some details:

We assume that $\mathbf{w}$ is not a path, and we write the walk $\mathbf{w}$ as $(w_0, e_1, w_1, e_2, w_2, \ldots, e_k, w_k)$. Then, there exists a pair (i, j) of integers i and j with $i < j$ and $w_i = w_j$. Among all such pairs, we pick one with **minimum** difference $j - i$. Then, $(w_i, e_{i+1}, w_{i+1}, \ldots, e_j, w_j)$ is a closed walk. We claim that this closed walk is a cycle.

To do so, we need to show that

1. the vertices $w_i, w_{i+1}, \ldots, w_{j-1}$ are distinct;
2. the edges $e_{i+1}, e_{i+2}, \ldots, e_j$ are distinct;
3. we have $j - i \geq 1$.

The first of these claims follows from the minimality of $j - i$. The third follows from $i < j$. It remains to prove the second claim. In other words, it remains to prove that the edges $e_{i+1}, e_{i+2}, \ldots, e_j$ are distinct, i.e., that we have $e_a \neq e_b$ for any two integers a and b satisfying $i < a < b \leq j$. Let us do this. Let a and b be two integers satisfying $i < a < b \leq j$. We must show that $e_a \neq e_b$. We distinguish two cases: the case $a = b - 1$ and the case $a \neq b - 1$.

Corollary 1.1.10. Let G be a multigraph. Assume that G has a closed walk $\mathbf{w}$ of length > 0 such that no two adjacent edges of $\mathbf{w}$ are identical. Then, G has a cycle.

We ended Lecture 4 by proving the following theorem for simple graphs:

Theorem 1.1.11. Let G be a multigraph. Let u and v be two vertices in G. Assume that there are two distinct paths from u to v. Then, G has a cycle.

The same proof applies to multigraphs, once the obvious changes are made (e.g., instead of $p_{a-1}p_a$ and $q_{b-1}q_b$, we need to take the last edges of the two walks $\mathbf{p}$ and $\mathbf{q}$).

1.1.3. Lecture 5

Lecture 5 began with the following proposition:

Proposition 1.1.12. Let G be a simple graph with at least one vertex. Let $d > 1$ be an integer. Assume that each vertex of G has degree $\geq d$. Then, G has a cycle of length $\geq d + 1$.

Is this still true for multigraphs? No. We can take a multigraph with a single vertex and lots of loops around it. In that case, its degree can be very large, but it has no cycles of length > 1.

Next, we extend the definition of $G \setminus e$ to multigraphs:

Definition 1.1.13. Let $G = (V, E, \varphi)$ be a multigraph. Let e be an edge of G. Then, $G \setminus e$ will mean the graph obtained from G by removing this edge e. In other words,

$$G \setminus e := \left(V,\ E \setminus \{e\}\,,\ \varphi\mid_{E\setminus\{e\}}\right).$$

- If $a = b - 1$, then e_a and e_b are two adjacent edges of $\mathbf{w}$ and thus distinct (since we assumed that no two adjacent edges of $\mathbf{w}$ are identical). Thus, $e_a \neq e_b$ is proved in the case when $a = b - 1$.
- Now, consider the case when $a \neq b - 1$. In this case, we must have $a < b - 1$ (since $a < b$ entails $a \leq b - 1$). Also, $i \leq a - 1$ (since $i < a$). Hence, $i \leq a - 1 < a < b - 1 \leq j - 1$ (since $b \leq j$). Therefore, $b - 1$, $a - 1$ and a are three distinct elements of the set $\{i, i + 1, \ldots, j - 1\}$. Consequently, w_{b-1}, w_{a-1}, w_a are three distinct vertices (since the vertices $w_i, w_{i+1}, \ldots, w_{j-1}$ are distinct). Therefore, $w_{b-1} \notin \{w_{a-1}, w_a\} = \varphi(e_a)$ (since $\mathbf{w}$ is a walk, so that the edge e_a has endpoints w_{a-1} and w_a). However, $\varphi(e_b) = \{w_{b-1}, w_b\}$ (since $\mathbf{w}$ is a walk, so that the edge e_b has endpoints w_{b-1} and w_b). Now, comparing $w_{b-1} \in \{w_{b-1}, w_b\} = \varphi(e_b)$ with $w_{b-1} \notin \varphi(e_a)$, we see that the sets $\varphi(e_b)$ and $\varphi(e_a)$ must be distinct (since $\varphi(e_b)$ contains w_{b-1} but $\varphi(e_a)$ does not). In other words, $\varphi(e_b) \neq \varphi(e_a)$. Hence, $e_b \neq e_a$. In other words, $e_a \neq e_b$. Thus, $e_a \neq e_b$ is proved in the case when $a \neq b - 1$.

We have now proved $e_a \neq e_b$ in both cases, so we are done.

Some authors write $G - e$ for $G \setminus e$.

Just as we proved for simple graphs in Lecture 5, we can show the following for multigraphs:

Theorem 1.1.14. Let G be a multigraph. Let e be an edge of G. Then:

(a) If e is an edge of some cycle of G, then the components of $G \setminus e$ are precisely the components of G. (Keep in mind that the components are sets of vertices. It is these sets that we are talking about here, not the induced subgraphs on these sets.)

(b) If e appears in no cycle of G (in other words, there exists no cycle of G such that e is an edge of this cycle), then the graph $G \setminus e$ has one more component than G.

Note that an edge e that is a loop always is an edge of a cycle (indeed, it creates a cycle of length 1), and can never appear on any path; thus, removing such an edge e obviously does not change the path-connectedness relation.

Defining cut-edges and bridges just as we did for simple graphs, we equally recover the following corollary:

Corollary 1.1.15. Let e be an edge of a multigraph G. Then, e is a bridge if and only if e is a cut-edge.

We ended Lecture 5 by defining and studying dominating sets. We could define dominating sets for multigraphs in the same way as for simple graphs, but we would not get anything new this way. Indeed, if G is a multigraph, then the dominating sets of G are precisely the dominating sets of G^{simp}. So we can reduce any claims about dominating sets of multigraphs to analogous claims about simple graphs.

1.1.4. Lecture 6

As we said before, a multigraph G has a Hamiltonian path or Hamiltonian cycle if and only if the corresponding simple graph G^{simp} has one. This does not mean, however, that everything we proved in Lecture 6 still applies to multigraphs. For instance, take the following theorem:

Theorem 1.1.16 (Ore)**.** Let $G = (V, E)$ be a simple graph with n vertices, where $n \geq 3$.

Assume that $\deg x + \deg y \geq n$ for any two non-adjacent vertices x and y. Then, G has a hamc.

This does not hold for multigraphs, because we could duplicate edges to make degrees arbitrarily large, without necessarily creating a hamc.

Likewise, the following corollary also fails for multigraphs:

Corollary 1.1.17 (Dirac)**.** Let $G = (V, E)$ be a simple graph with n vertices, where $n \geq 3$.

Assume that $\deg x \geq \dfrac{n}{2}$ for each vertex $x \in V$.

Then, G has a hamc.

The necessary criterion for hamcs and hamps that we proved afterwards still holds for multigraphs, but that's clear because it can be derived from the corresponding property of G^{simp}.

1.2. Eulerian circuits and walks

Recall that a Hamiltonian path or cycle is a path or cycle that contains all vertices of the graph. Being a path or cycle, it has to contain each of them exactly once (except, in the case of a cycle, of its starting point).

What about a walk or closed walk that contains all **edges** exactly once instead? These are called "Eulerian" walks or circuits; here is the formal definition:

Definition 1.2.1. Let G be a multigraph.

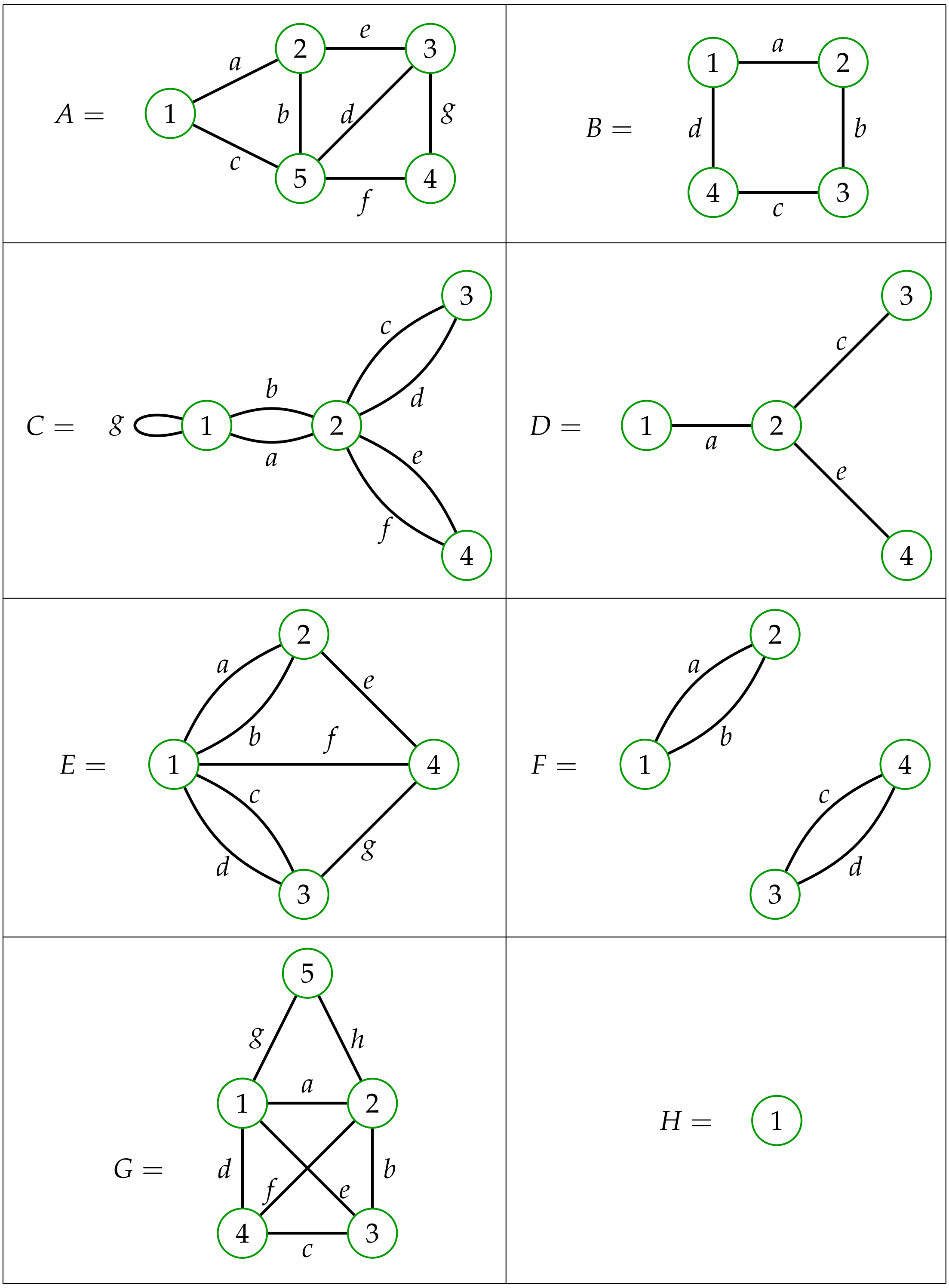
(a) A walk of G is said to be **Eulerian** if each edge of G appears exactly once in this walk.

(In other words: A walk $(v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k)$ of G is said to be **Eulerian** if for each edge e of G, there exists exactly one $i \in \{1, 2, \ldots, k\}$ such that $e = e_i$.)

(b) An **Eulerian circuit** of G means a circuit (i.e., closed walk) of G that is Eulerian. (Strictly speaking, the preceding sentence is redundant, but we still said it to stress the notion of an Eulerian circuit.)

Unlike for Hamiltonian paths and cycles, an Eulerian walk or circuit is usually not a path or cycle. Also, finding an Eulerian walk in a multigraph G is not the same as finding an Eulerian walk in the simple graph G^{simp}. (Nevertheless, some authors call Eulerian walks "Eulerian paths" and call Eulerian circuits "Eulerian cycles". This is rather confusing.)

Example 1.2.2. Consider the following multigraphs:



- The multigraph A has an Eulerian walk $(3, d, 5, b, 2, e, 3, g, 4, f, 5, c, 1, a, 2)$. But A has no Eulerian circuit. The easiest way to see this is by observing that A has a vertex of odd degree (e.g., the vertex 2). If an Eulerian circuit were to exist, then it would have to enter this vertex as often as it exited it; but this would mean that the degree of this vertex would be even (because each edge containing this vertex would be used exactly once either to enter or to exit it, except for loops, which would be used twice). So, more generally, any multigraph that has a vertex of odd degree cannot have an Eulerian circuit.

- The multigraph B has an Eulerian circuit $(1, a, 2, b, 3, c, 4, d, 1)$, and thus of course an Eulerian walk (since any Eulerian circuit is an Eulerian walk).

- The multigraph C has an Eulerian circuit $(1, g, 1, b, 2, c, 3, d, 2, e, 4, f, 2, a, 1)$.

- The multigraph D has no Eulerian walk. Indeed, it has four vertices of odd degree. If v is a vertex of odd degree, then any Eulerian walk has to either start or end at v (since otherwise, the walk would enter and leave v equally often, but then the degree of v would be even). But a walk can only have one starting point and one ending point. This allows for two vertices of odd degree, but not more than two. So, more generally, any multigraph that has more than two vertices of odd degree cannot have an Eulerian walk.

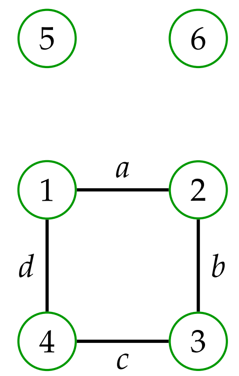
- The multigraph E has no Eulerian walk. The reason is the same as for D. Note that E is the famous multigraph of bridges in Königsberg, as studied by Euler in 1736 (see the Wikipedia page for "Seven bridges of Königsberg" for the backstory).

- The multigraph F has no Eulerian walk, since it has two components, each containing at least one edge. (An Eulerian walk would have to contain both edges b and c, but there is no way to walk between them, since they belong to different components.)

- The multigraph G has an Eulerian walk, namely $(3, b, 2, h, 5, g, 1, a, 2, f, 4, d, 1, e, 3, c, 4)$. It has no Eulerian circuit, since it has two vertices of odd degree.

- The multigraph H has an Eulerian circuit, namely (1).

Remark 1.2.3. For the pedants: A multigraph can have an Eulerian circuit even if it is not connected, as long as all its edges belong to the same component (i.e., all but one components are just singletons with no edges). Here is an example:



How hard is it to find an Eulerian walk or circuit in a multigraph, or to check if there is any? Surprisingly, this is a lot easier than the same questions for Hamiltonian paths or cycles. The second question in particular is answered (for connected multigraphs) by the **Euler–Hierholzer theorem**:

Theorem 1.2.4 (Euler, Hierholzer)**.** Let G be a connected multigraph. Then:

(a) The multigraph G has an Eulerian circuit if and only if each vertex of G has even degree.

(b) The multigraph G has an Eulerian walk if and only if all but at most two vertices of G have even degree.

We already proved the "$\Longrightarrow$" directions of both parts **(a)** and **(b)** in Example 1.2.2. As for the "$\Longleftarrow$" directions, I don't think Euler proved them in his 1736 paper, but Hierholzer did in 1873. The "standard" proof can be found in many texts, such as [Guicha20, Theorem 5.2.2 and Theorem 5.2.3]. I will sketch a different proof, which I learnt from [LeLeMe18, Problem 12.35]. We begin with the following definition:

Definition 1.2.5. Let G be a multigraph. A **trail** of G means a walk of G whose edges are distinct.

So a trail can repeat vertices, but cannot repeat edges.

Thus, an Eulerian walk has to be a trail. A trail cannot be any longer than a Eulerian walk. So a reasonable way to try constructing an Eulerian walk is to start with some trail, and make it progressively longer until it becomes Eulerian (hopefully).

This suggests the following approach to proving the above theorem: We pick the longest trail of G and argue that (under the right assumptions) it has to be Eulerian, since otherwise there would be a way to make it longer. Of course, we need to find such a way. We will do this next time. Here is the first step:

Lemma 1.2.6. Let G be a multigraph with at least one vertex. Then, G has a longest trail.

Proof. Clearly, G has at least one trail (e.g., a length-0 trail from a vertex to itself). Moreover, G has only finitely many trails (since each edge of G can only be used once in a trail, and there are only finitely many edges). So the maximum principle proves the lemma. $\square$

References

[Guicha20] David Guichard, *An Introduction to Combinatorics and Graph Theory*, 17 May 2022.
`https://www.whitman.edu/mathematics/cgt_online/book/`

[LeLeMe18] Eric Lehman, F. Thomson Leighton, Albert R. Meyer, *Mathematics for Computer Science*, revised Tuesday 6th June 2018,
`https://courses.csail.mit.edu/6.042/spring18/mcs.pdf` .