Math 530 Spring 2022, Lecture 15: Trees

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

1. Trees and arborescences (cont'd)

1.1. Spanning trees (cont'd)

Last time, we gave three proofs of the following theorem:

Theorem 1.1.1. Each connected multigraph G has at least one spanning tree.

Let us outline a fourth proof of this now:

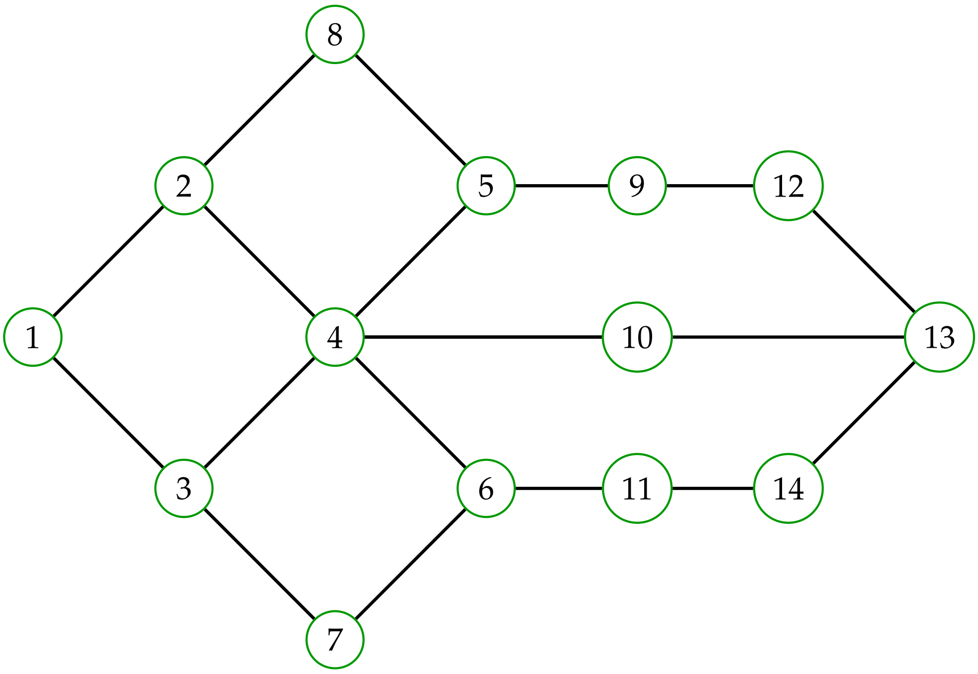
Fourth proof of Theorem 1.1.1 (sketched). We imagine a snake that slithers along the edges of G, trying to eventually bite each vertex. It starts at some vertex r, which it immediately bites. Any time the snake enters a vertex v, it makes the following step:

- If some neighbor of v has not been bitten yet, then the snake picks such a neighbor w as well as some edge f that joins w with v; the snake then moves to w along the edge f, bites the vertex w and marks the edge f.
- If not, then the snake marks the vertex v as fully digested and backtracks (along the marked edges) to the last vertex it has visited but not fully digested yet.

Once backtracking is no longer possible (because there are no more vertices left that are not fully digested), the procedure is finished. I claim that the marked edges at that moment are the edges of a spanning tree of G.

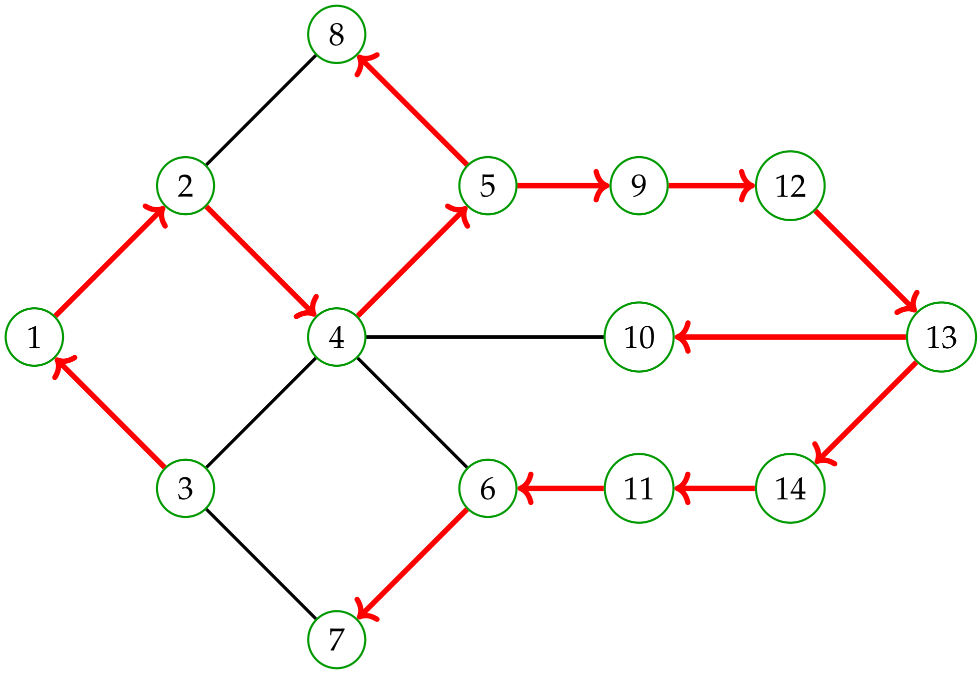
I won't prove this claim in detail, but I will give some hints. First, however, an example:

Example 1.1.2. Let G be the following connected multigraph:



.

Let our snake start its journey at $r = 3$. It bites this vertex. Then, let's say that it picks the vertex 1 as its next victim (it could just as well go to 4 or 7; the snake has many choices, but we follow one possible trip). Thus, it next arrives at vertex 1, bites it and marks the edge that brought it to this vertex. As its next destination, it necessarily picks the vertex 2 (since vertex 3 has already been bitten). It moves to vertex 2, bites it and marks the edge. Next, let's say that it picks the vertex 4 (the other option would be 8). It thus moves to 4, bites it and marks the edge. Proceeding likewise, it then moves to 5 (the other options are 6 and 10; the vertices 2 and 3 do not qualify since they are already bitten), bites 5 and marks an edge. From there, let's say it moves to 8, bites 8 and marks an edge. Now, there is no longer an unbitten neighbor of 8 to move to. Thus, the snake marks the vertex 8 as fully digested and backtracks to the last vertex not fully digested – which, at this point, is 5. From this vertex 5, it moves on to 9 (this is the only option, since 4 and 8 have already been bitten). And so on. Here is one possible outcome of this journey (there are a few more decisions that the snake can make here, so you may get a different one):



Here, the marked edges are drawn in bold red ink, and endowed with an arrow that represents the direction in which they were first used (e.g., the edge joining 2 with 4 has an arrow towards 4 because it was first used to get from 2 to 4).

Now, as promised, let me outline a proof of the above claim (that the marked edges form a spanning tree of G). To wit, argue the following four observations (ideally in this order):

1. After each step, the marked edges are precisely the edges along which the snake has moved so far.

2. After each step, the network of bitten vertices and marked edges is a tree.

3. After enough steps, each bitten vertex is fully digested.

4. At that point, the network of bitten vertices and marked edges is a spanning tree (since each neighbor of a fully digested vertex is bitten, thus fully digested by observation 3).

Details are left to the reader.

The result is that Theorem 1.1.1 is proved once again. However, more comes out of the above construction if you know where to look. The spanning tree T of G whose edges are the edges marked by the snake is called a **depth-first search ("DFS") tree**. It has the following extra property: If u and v are two adjacent vertices of G, then either u lies on the path from r to v in T, or v lies on the path from r to u in T. (This called a "lineal spanning tree". See [BenWil06, §6.1] for details.) □

Spanning trees have lots of applications:

- A spanning tree of a graph can be viewed as a kind of "backbone" of the graph, which in particular provides "canonical" paths between any two vertices. This is useful, e.g., for networking applications where having a choice between different paths would be problematic (see, e.g., the Spanning Tree Protocol).
- A w-minimum spanning tree (see Homework set #5 exercise 6) solves a global version of the cheapest-path problem. It can also be used for detecting clusters.
- Depth-first search (the algorithm used in our fourth proof of Theorem 1.1.1) can also be used as a way to traverse all vertices of a given graph and return back to the starting point. In particular, this provides an algorithmic way to solve mazes (since a maze can be modeled as a graph, where the vertices correspond to "rooms" and the edges correspond to "doors"). This appears to have been the original motivation for Trémaux to invent depth-first search back in the 19th century.

Here is a more theoretical application of spanning trees:

Definition 1.1.3. A vertex v of a connected multigraph G is said to be a **cut-vertex** if the graph $G \setminus v$ is disconnected. (Recall that $G \setminus v$ is the multigraph obtained from G by removing the vertex v and all edges that contain v.)

Proposition 1.1.4. Let G be a connected multigraph with ≥ 2 vertices. Then, there are at least 2 vertices of G that are **not** cut-vertices.

Proof. Pick a spanning tree T of G (we know from Theorem 1.1.1 that such a spanning tree exists). Then, T has at least 2 leaves (as we proved last time). But each leaf of T is a non-cut-vertex of G (why?). □

Remark 1.1.5. It is not true that conversely, any non-leaf of T is a cut-vertex of G. So we cannot get any lower bound on the number of cut-vertices. And this is not surprising: Lots of graphs (e.g., the complete graph K_n for $n \geq 2$) have no cut-vertices at all. These graphs are said to be **2-connected**.

So we have learnt that connected graphs have spanning trees. What do disconnected graphs have?

Corollary 1.1.6. Each multigraph has a spanning forest.

Proof. Apply Theorem 1.1.1 to each component of the multigraph. Then, combine the resulting spanning trees into a spanning forest. □

1.2. Centers of graphs and trees

Given a graph, we can define a "distance" between any two of its vertices, simply by counting edges on the shortest path from one to the other:

Definition 1.2.1. Let G be a multigraph.

For any two vertices u and v of G, we define the **distance** between u and v to be the smallest length of a path from u to v. If no such path exists, then this distance is defined to be ∞.

The distance between u and v is denoted by $d(u,v)$ or by $d_G(u,v)$ when the graph G is not clear from the context.

Example 1.2.2. If G is the multigraph from Example 1.1.2, then

$$d_G(1,9) = 4, \qquad d_G(4,13) = 2, \qquad d_G(4,4) = 0.$$

Remark 1.2.3. Distances in a multigraph satisfy the rules that you would expect a distance function to satisfy:

(a) We have $d(u,u) = 0$ for any vertex u.

(b) We have $d(u,v) = d(v,u)$ for any vertices u and v.

(c) We have $d(u,v) + d(v,w) \geq d(u,w)$ for any vertices u, v and w. (We understand that $\infty \geq m$ for any $m \in \mathbb{N}$.)

Also:

(d) The distances $d(u,v)$ do not change if we replace "path" by "walk" in the definition of the distance.

Proof. Part **(d)** follows from Corollary 1.1.5 in Lecture 8. The proofs of **(a)**, **(b)** and **(c)** are then straightforward (the proof of **(c)** relies on part **(d)**, because splicing two paths generally only yields a walk, not a path). □

We note that the definition of a distance becomes simpler if our multigraph is a tree: Namely, if T is a tree, then the distance $d(u,v)$ between two vertices u and v is the length of the **only** path from u to v in T. Thus, in a tree, we do not have to find the shortest path.

We can now define "eccentricities":

Definition 1.2.4. Let v be a vertex of a multigraph $G = (V, E, \varphi)$. The **eccentricity** of v (with respect to G) is defined to be the number

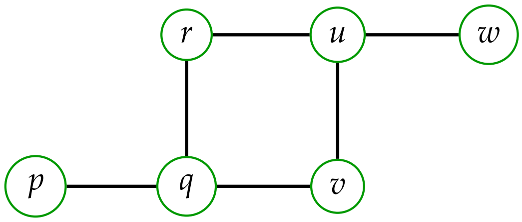
$$\max \{d(v, u) \mid u \in V\} \in \mathbb{N} \cup \{\infty\}.$$

This eccentricity is denoted by $\operatorname{ecc} v$ or $\operatorname{ecc}_G v$.

Definition 1.2.5. Let $G = (V, E, \varphi)$ be a multigraph. Then, a **center** of G means a vertex of G whose eccentricity is minimum (among all vertices).

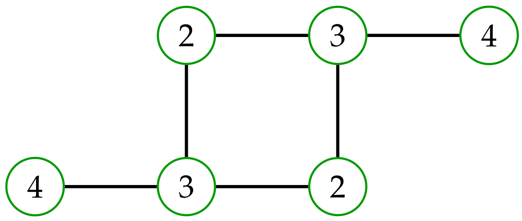
(Some authors have a slightly different definition of a "center": They define the **center** of G to be the **set** of all vertices of G whose eccentricity is minimum. That is, what they call "center" is the set of what we call "centers".)

Example 1.2.6. Let G be the following multigraph:



.

Then, the eccentricities of its vertices are as follows (we are just labeling each vertex with its eccentricity):



.

Thus, the centers of G are the vertices r and v.

Example 1.2.7. Let G be a complete graph K_n (with n vertices). Then, each vertex of G has the same eccentricity (which is 1 if $n \geq 2$ and 0 if $n = 1$), and thus each vertex of G is a center of G.

Example 1.2.8. Let G be a graph with more than one component. Then, each vertex v of G has eccentricity ∞ (because there exists at least one vertex u that lies in a different component of G than v, and thus this vertex u satisfies $d(v, u) = \infty$). Hence, each vertex of G is a center of G.

As we see from Example 1.2.8, eccentricity and centers are not very useful notions when the graph is disconnected. Even for a connected graph, Example

1.2.6 shows that the centers do not necessarily form a connected subgraph. However, in a tree, they behave a lot better:

Theorem 1.2.9. Let T be a tree. Then:

(a) The tree T has either 1 or 2 centers.

(b) If T has 2 centers, then these 2 centers are adjacent.

(c) Moreover, these centers can be found by the following algorithm:

If T has more than 2 vertices, then we remove all leaves from T (simultaneously). What remains is again a tree. If that tree still has more than 2 vertices, we remove all leaves from it (simultaneously). The result is again a tree. If that tree still has more than 2 vertices, we remove all leaves from it (simultaneously), and continue doing so until we are left with a tree that has only 1 or 2 vertices. These vertices are the centers of T.

To prove Theorem 1.2.9, we first study how a tree is affected when all its leaves are removed:

Lemma 1.2.10. Let $T = (V, E, \varphi)$ be a tree with more than 2 vertices.

Let L be the set of all leaves of T.

Let $T \setminus L$ be the induced submultigraph of T on the set $V \setminus L$. (Thus, $T \setminus L$ is obtained from T by removing all the vertices in L and all edges that contain a vertex in L.)

Then:

(a) The multigraph $T \setminus L$ is a tree.

(b) For any $u \in V \setminus L$ and $v \in V \setminus L$, we have

$$\{\text{paths of } T \text{ from } u \text{ to } v\} = \{\text{paths of } T \setminus L \text{ from } u \text{ to } v\}$$

(that is, the paths of T from u to v are precisely the paths of $T \setminus L$ from u to v).

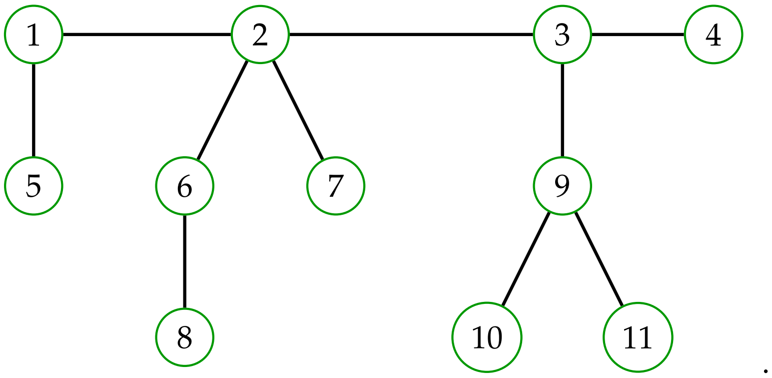
(c) For any $u \in V \setminus L$ and $v \in V \setminus L$, we have $d_T (u, v) = d_{T \setminus L} (u, v)$.

(d) Each vertex $v \in V \setminus L$ satisfies $\operatorname{ecc}_T v = \operatorname{ecc}_{T \setminus L} v + 1$.

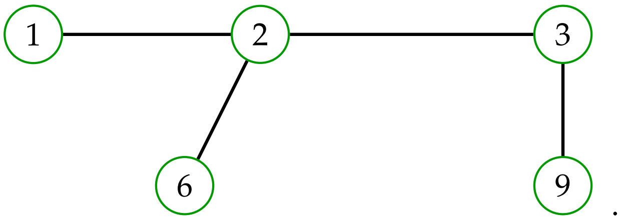
(e) Each leaf $v \in L$ satisfies $\operatorname{ecc}_T v = \operatorname{ecc}_T w + 1$, where w is the unique neighbor of v in T. (A **neighbor** of v means a vertex that is adjacent to v.)

(f) The centers of T are precisely the centers of $T \setminus L$.

Example 1.2.11. Let T be the following tree:



Then, the set L from Lemma 1.2.10 is $\{4,5,7,8,10,11\}$, and the tree $T \setminus L$ looks as follows:



Proof of Lemma 1.2.10. First, we notice that T is a forest (since T is a tree), and thus has no cycles. In particular, T therefore has no loops and no parallel edges. Also, for any two vertices u and v of T, there is a unique path from u to v in T.

Next, we introduce some terminology: If $\mathbf{p}$ is a path of some multigraph, then an **intermediate vertex** of $\mathbf{p}$ shall mean a vertex of $\mathbf{p}$ that is neither the starting point nor the ending point of $\mathbf{p}$. In other words, if $\mathbf{p} = (p_0, e_1, p_1, e_2, p_2, \ldots, e_k, p_k)$ is a path of some multigraph, then the intermediate vertices of $\mathbf{p}$ are $p_1, p_2, \ldots, p_{k-1}$. Clearly, any intermediate vertex of a path $\mathbf{p}$ must have degree ≥ 2 (since the path $\mathbf{p}$ enters it along some edge, and leaves it along another). Hence, if $\mathbf{p}$ is a path of T, then

$$\text{any intermediate vertex of } \mathbf{p} \text{ must belong to } V \setminus L \tag{1}$$

(because it must have degree ≥ 2, thus cannot be a leaf of T; but this means that it cannot belong to L; therefore, it must belong to $V \setminus L$).

(b) Let $u \in V \setminus L$ and $v \in V \setminus L$. Let $\mathbf{p}$ be a path of T from u to v. We shall show that $\mathbf{p}$ is a path of $T \setminus L$ as well.

Indeed, let us first check that all vertices of $\mathbf{p}$ belong to $V \setminus L$. This is clear for the vertices u and v (since $u \in V \setminus L$ and $v \in V \setminus L$); but it also holds for every intermediate vertex of $\mathbf{p}$ (by (1)). Thus, it does indeed hold for all vertices of $\mathbf{p}$.

We have thus shown that all vertices of $\mathbf{p}$ belong to $V \setminus L$. Hence, $\mathbf{p}$ is a path of $T \setminus L$ (since $T \setminus L$ is the induced submultigraph of T on the set $V \setminus L$).

Forget that we fixed $\mathbf{p}$. We have thus shown that every path $\mathbf{p}$ of T from u to v is also a path of $T \setminus L$. Hence,

$$\{\text{paths of } T \text{ from } u \text{ to } v\} \subseteq \{\text{paths of } T \setminus L \text{ from } u \text{ to } v\}.$$

Conversely, we have

$$\{\text{paths of } T \setminus L \text{ from } u \text{ to } v\} \subseteq \{\text{paths of } T \text{ from } u \text{ to } v\},$$

since every path of $T \setminus L$ is a path of T (because $T \setminus L$ is a submultigraph of T). Combining these two facts, we obtain

$$\{\text{paths of } T \text{ from } u \text{ to } v\} = \{\text{paths of } T \setminus L \text{ from } u \text{ to } v\}.$$

This proves Lemma 1.2.10 **(b)**.

(c) This follows from Lemma 1.2.10 **(b)**, since the distance $d_G(u,v)$ of two vertices u and v in a graph G is defined to be the smallest length of a path from u to v.

(a) The graph T is a tree, thus a forest. Hence, its submultigraph $T \setminus L$ is a forest as well (since any cycle of $T \setminus L$ would be a cycle of T). It thus remains to show that $T \setminus L$ is connected.

First, it is easy to see that $T \setminus L$ has at least one vertex[1]. It remains to show that any two vertices of $T \setminus L$ are path-connected.

Let u and v be two vertices of $T \setminus L$. Then, $u \in V \setminus L$ and $v \in V \setminus L$. Hence, Lemma 1.2.10 **(b)** yields

$$\{\text{paths of } T \text{ from } u \text{ to } v\} = \{\text{paths of } T \setminus L \text{ from } u \text{ to } v\}.$$

Thus, $\{\text{paths of } T \setminus L \text{ from } u \text{ to } v\} = \{\text{paths of } T \text{ from } u \text{ to } v\} \neq \varnothing$ (since there exists a path of T from u to v (because T is connected)). In other words, there exists a path of $T \setminus L$ from u to v. In other words, u and v are path-connected in $T \setminus L$.

We have now shown that any two vertices u and v of $T \setminus L$ are path-connected in $T \setminus L$. This entails that $T \setminus L$ is connected (since $T \setminus L$ has at least one vertex). This proves Lemma 1.2.10 **(a)**.

[1] *Proof.* We assumed that T has more than 2 vertices. In other words, there exist three distinct vertices u, v, w of T. Consider these u, v, w. If all three distances $d_T(u,v)$, $d_T(v,w)$ and $d_T(w,u)$ were equal to 1, then T would have a cycle (of the form $(u, *, v, *, w, *, u)$, where each asterisk stands for some edge); but this would contradict the fact that T has no cycles. Thus, not all of these three distances are equal to 1. Hence, at least one of them is $\neq 1$. WLOG assume that $d_T(u,v) \neq 1$ (otherwise, we permute u, v, w). Hence, the path from u to v has more than one edge (indeed, it must have at least one edge, since u and v are distinct). Therefore, this path has at least one intermediate vertex. This intermediate vertex then must belong to $V \setminus L$ (by (1)). Hence, it is a vertex of the subgraph $T \setminus L$. This shows that $T \setminus L$ has at least one vertex.

(d) If u and v are two vertices of $T \setminus L$, then the two distances $d_T(u,v)$ and $d_{T\setminus L}(u,v)$ are equal (by Lemma 1.2.10 **(c)**); thus, we shall denote both distances by $d(u,v)$ (since there is no confusion to be afraid of).

Let $v \in V \setminus L$. We must show that $\operatorname{ecc}_T v = \operatorname{ecc}_{T\setminus L} v + 1$.

Let u be a vertex of $T \setminus L$ such that $d(v,u)$ is maximum. Thus, $\operatorname{ecc}_{T\setminus L} v = d(v,u)$ (by the definition of $\operatorname{ecc}_{T\setminus L} v$). However, u is a vertex of $T \setminus L$, and thus does not belong to L. Hence, u is not a leaf of T (since L is the set of all leaves of T). Hence, u has degree ≥ 2 in T (since a vertex in a tree with more than 1 vertex cannot have degree 0).

Now, consider the path $\mathbf{p}$ from v to u in the tree T. This path $\mathbf{p}$ has length $d(v,u)$. Since u has degree ≥ 2, there exist at least two edges of T that contain u. Hence, in particular, there exists at least one edge f that contains u and is distinct from the last edge of $\mathbf{p}$ [2]. Consider this edge f. Let w be the endpoint of f other than u. Appending f and w to the end of the path $\mathbf{p}$, we obtain a walk from v to w. This walk is backtrack-free (since f is distinct from the last edge of $\mathbf{p}$) and thus must be a path (by Proposition 1.1.2 in Lecture 13, since T has no cycles). This path has length $d(v,u)+1$ (since it was obtained by appending an edge to the path $\mathbf{p}$, which has length $d(v,u)$). Hence, $d(v,w) = d(v,u)+1$. But the definition of eccentricity yields

$$\operatorname{ecc}_T v \geq d(v,w) = \underbrace{d(v,u)}_{=\operatorname{ecc}_{T\setminus L} v} + 1 = \operatorname{ecc}_{T\setminus L} v + 1. \tag{2}$$

On the other hand, let x be a vertex of T such that $d(v,x)$ is maximum. Thus, $\operatorname{ecc}_T v = d(v,x)$ (by the definition of $\operatorname{ecc}_T v$). The path from v to x has length ≥ 1 (since otherwise, we would have $x = v$ and therefore $d(v,x) = d(v,v) = 0$, which would easily contradict the maximality of $d(v,x)$). Thus, it has a second-to-last vertex. Let y be this second-to-last vertex. Then, the path from v to y is simply the path from v to x with its last edge removed. Consequently, $d(v,y) = d(v,x) - 1$. However, it is easy to see that $y \in V \setminus L$ [3]. In other words, y is a vertex of $T \setminus L$. Thus, the definition of eccentricity yields

$$\operatorname{ecc}_{T\setminus L} v \geq d(v,y) = \underbrace{d(v,x)}_{=\operatorname{ecc}_T v} - 1 = \operatorname{ecc}_T v - 1,$$

so that $\operatorname{ecc}_T v \leq \operatorname{ecc}_{T\setminus L} v + 1$. Combining this with (2), we obtain $\operatorname{ecc}_T v = \operatorname{ecc}_{T\setminus L} v + 1$. This proves Lemma 1.2.10 **(d)**.

[2] If the path $\mathbf{p}$ has no edges, then f can be any edge that contains u.

[3] *Proof.* Assume the contrary. Thus, $y \notin V \setminus L$. Hence, $y \neq v$ (since $y \notin V \setminus L$ but $v \in V \setminus L$).

However, y is the second-to-last vertex of the path from v to x. Therefore, y is either the starting point v of this path, or an intermediate vertex of this path. Since $y \neq v$, we thus conclude that y is an intermediate vertex of this path. Hence, by (1), we see that y must belong to $V \setminus L$. But this contradicts $y \notin V \setminus L$. This contradiction shows that our assumption was false, qed.

(e) If u and v are two vertices of $T \setminus L$, then the two distances $d_T(u,v)$ and $d_{T\setminus L}(u,v)$ are equal (by Lemma 1.2.10 **(c)**); thus, we shall denote both distances by $d(u,v)$ (since there is no confusion to be afraid of).

Let $v \in L$ be a leaf. Let w be the unique neighbor of v in T. We must prove that $\operatorname{ecc}_T v = \operatorname{ecc}_T w + 1$.

We first claim that

$$d(v,u) = d(w,u) + 1 \qquad \text{for each } u \in V \setminus \{v\}. \tag{3}$$

[*Proof of (3):* We have $\deg v = 1$ (since v is a leaf). In other words, there is a unique edge of T that contains v. Let e be this edge. The endpoints of e are v and w (since w is the unique neighbor of v). Thus, $v \neq w$ (since T has no loops) and $d(v,w) = 1$.

Now, let $u \in V \setminus \{v\}$. Then, the path from v to u in T must have length ≥ 1 (since $u \neq v$), and therefore must begin with the edge e (since e is the only edge that contains v). If we remove this edge e from this path, we thus obtain a path from w to u. As a consequence, the path from v to u is longer by exactly 1 edge than the path from w to u. In other words, we have $d(v,u) = d(w,u) + 1$. This proves (3).]

Now, the definition of eccentricity yields

$$\operatorname{ecc}_T v = \max\{d(v,u) \mid u \in V\}. \tag{4}$$

This maximum is clearly **not** attained for $u = v$ (since $d(v,v) = 0$ is smaller than $d(v,w) = 1$). Thus, this maximum does not change if we remove v from its indexing set V. Hence, (4) rewrites as

$$\begin{aligned}\operatorname{ecc}_T v &= \max\left\{\underbrace{d(v,u)}_{\substack{=d(w,u)+1\\ \text{(by (3))}}} \mid u \in V \setminus \{v\}\right\} \\ &= \max\{d(w,u) + 1 \mid u \in V \setminus \{v\}\} \\ &= \max\{d(w,u) \mid u \in V \setminus \{v\}\} + 1. \end{aligned} \tag{5}$$

On the other hand, the definition of eccentricity yields

$$\operatorname{ecc}_T w = \max\{d(w,u) \mid u \in V\}. \tag{6}$$

We shall now show that this maximum does not change if we remove v from its indexing set V. In other words, we shall show that

$$\max\{d(w,u) \mid u \in V\} = \max\{d(w,u) \mid u \in V \setminus \{v\}\}. \tag{7}$$

[*Proof of (7):* Assume that (7) is false. Then, the maximum $\max\{d(w,u) \mid u \in V\}$ is attained **only** at $u = v$. In other words, we have

$$d(w,v) > d(w,u) \qquad \text{for all } u \in V \setminus \{v\}. \tag{8}$$

However, the tree T has more than 2 vertices. Thus, it has a vertex u that is distinct from both v and w. Consider this u. Thus, $u \in V \setminus \{v\}$, so that (8) yields $d(w,v) > d(w,u)$. In view of $d(w,v) = d(v,w) = 1$, this rewrites as $1 > d(w,u)$, so that $d(w,u) < 1$. Therefore, $w = u$. But this contradicts the fact that w is distinct from u. This contradiction shows that our assumption was false, and thus (7) is proved.]

Now, (5) becomes

$$\begin{aligned}\operatorname{ecc}_T v &= \underbrace{\max\{d(w,u) \ \mid \ u \in V \setminus \{v\}\}}_{\substack{=\max\{d(w,u) \ \mid \ u \in V\} \\ \text{(by (7))}}} + 1 \\ &= \underbrace{\max\{d(w,u) \ \mid \ u \in V\}}_{\substack{=\operatorname{ecc}_T w \\ \text{(by (6))}}} + 1 = \operatorname{ecc}_T w + 1.\end{aligned}$$

This proves Lemma 1.2.10 **(e)**.

(f) Lemma 1.2.10 **(e)** shows that any vertex $v \in L$ has a higher eccentricity than its unique neighbor. Thus, a vertex v of T that minimizes $\operatorname{ecc}_T v$ cannot belong to L. In other words, a vertex v of T that minimizes $\operatorname{ecc}_T v$ must belong to $V \setminus L$.

However, the centers of T are defined to be the vertices of T that minimize $\operatorname{ecc}_T v$. As we just proved, these vertices must belong to $V \setminus L$. Thus, the centers of T can also be characterized as the vertices $v \in V \setminus L$ that minimize $\operatorname{ecc}_T v$. However, a vertex $v \in V \setminus L$ minimizes $\operatorname{ecc}_T v$ if and only if it minimizes $\operatorname{ecc}_{T \setminus L} v$ (because Lemma 1.2.10 **(d)** yields $\operatorname{ecc}_T v = \operatorname{ecc}_{T \setminus L} v + 1$ for any such vertex v). Thus, we conclude that the centers of T can be characterized as the vertices $v \in V \setminus L$ that minimize $\operatorname{ecc}_{T \setminus L} v$. But this is precisely the definition of the centers of $T \setminus L$. As a consequence, we see that the centers of T are precisely the centers of $T \setminus L$. This proves Lemma 1.2.10 **(f)**. □

Proof of Theorem 1.2.9. We shall prove parts **(a)** and **(b)** of Theorem 1.2.9 by strong induction on $|\mathrm{V}(T)|$:

Induction step: Consider a tree T. Assume that parts **(a)** and **(b)** of Theorem 1.2.9 are true for any tree with fewer than $|\mathrm{V}(T)|$ many vertices. We must now prove these parts for our tree T.

If $|\mathrm{V}(T)| \leq 2$, then both parts are obvious. Hence, WLOG assume that $|\mathrm{V}(T)| > 2$. Thus, the tree T has more than 2 vertices. Let L be the set of all leaves of T. Note that $|L| \geq 2$ (since we know that any tree with at least 2 vertices has at least 2 leaves). Define the multigraph $T \setminus L$ as in Lemma 1.2.10. Then, Lemma 1.2.10 **(f)** shows that the centers of T are precisely the centers of $T \setminus L$.

However, Lemma 1.2.10 **(a)** yields that $T \setminus L$ is again a tree. This tree has fewer vertices than T (since $|L| \geq 2 > 0$). Hence, by the induction hypothesis, both parts **(a)** and **(b)** of Theorem 1.2.9 are true for the tree $T \setminus L$ instead of T.

In other words, the tree $T \setminus L$ has either 1 or 2 centers, and if it has 2 centers, then these 2 centers are adjacent. Since the centers of T are precisely the centers of $T \setminus L$, we can rewrite this as follows: The tree T has either 1 or 2 centers, and if it has 2 centers, then these 2 centers are adjacent. In other words, parts **(a)** and **(b)** of Theorem 1.2.9 hold for our tree T. This completes the induction step. Thus, parts **(a)** and **(b)** of Theorem 1.2.9 are proved.

(c) This follows from Lemma 1.2.10 **(f)**. Indeed, if T has at most 2 vertices, then all vertices of T are centers of T (this is trivial to check). If not, then each "leaf-removal" step of our algorithm leaves the set of centers of T unchanged (by Lemma 1.2.10 **(f)**), and thus the centers of the original tree T are precisely the centers of the tree that remains at the end of the algorithm. But the latter tree has at most 2 vertices, and thus its centers are precisely its vertices. So the centers of T are precisely the vertices that remain at the end of the algorithm. Theorem 1.2.9 **(c)** is proven. □

1.3. Arborescences

Enough about undirected graphs.

What would be a directed analogue of a tree? I.e., what kind of digraphs play the same role among digraphs that trees do among undirected graphs?

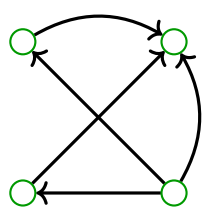
Trees are graphs that are connected and have no cycles. This suggests two directed versions:

- We can study digraphs that are strongly connected and have no cycles. Unfortunately, there is not much to study: Any such digraph has only 1 vertex and no arcs. (Make sure you understand why!)

- We can drop the connectedness requirement. Digraphs that have no cycles are called **acyclic**, and more typically they are called **dags** (short for "directed acyclic graphs").

  However, these dags aren't quite like trees. For example, a tree always has fewer edges than vertices, but a dag can have more arcs than vertices.[4]

[4]For example, here is a dag with 4 vertices and 5 arcs:



.

Here is a more convincing analogue of trees for digraphs:[5]

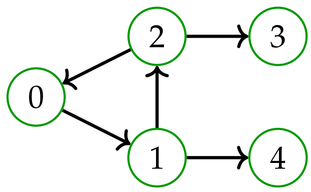
Definition 1.3.1. Let D be a multidigraph. Let r be a vertex of D.

(a) We say that r is a **from-root** (or, short, **root**) of D if for each vertex v of D, the digraph D has a path from r to v.

(b) We say that D is an **arborescence rooted from** r if r is a from-root of D and the undirected multigraph D^{und} has no cycles. (Recall that D^{und} is the multigraph obtained from D by turning each arc into an undirected edge. Parallel arcs are not merged into one!)

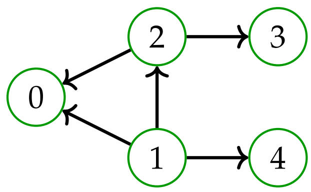
Of course, there are analogous notions of a "to-root" and an "arborescence rooted towards r", but these are just the same notions that we just defined with all arrows reversed. So we need not study them separately; we can just take any property of "rooted from" and reverse all arcs to make it into a property of "rooted to".

Example 1.3.2. The multidigraph



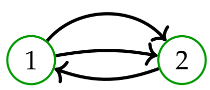
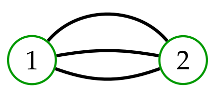
has three from-roots (namely, 0, 1 and 2). It is not an arborescence rooted from any of them, because turning each arc into an undirected edge yields a graph with a cycle.

If we reverse the arc from 0 to 1, then we obtain a multidigraph

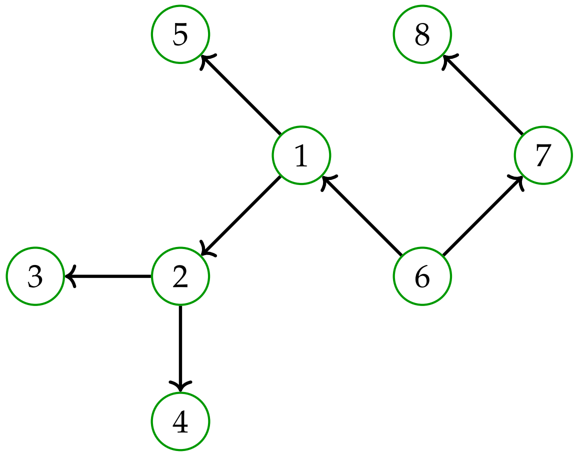


which has only one from-root (namely, 1) and is still not an arborescence (for the same reason as before).

[5] We recall that we defined a multigraph D^{und} for every multidigraph D (at the end of Lecture 9). Roughly speaking, this multigraph D^{und} is obtained by "forgetting the directions" of the arcs of D. Parallel arcs are not merged into one. For example,

if $D =$  , then $D^{\text{und}} =$  .

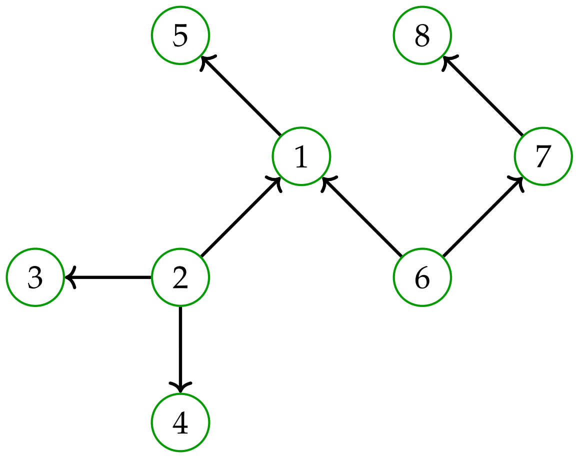
Example 1.3.3. Consider the following multidigraph:



.

This is an arborescence rooted from 6. Indeed, it has paths from 6 to all vertices, and turning each arc into an undirected edge yields a tree.

If we reverse the arc from 1 to 2, we obtain a multidigraph



,

which is **not** an arborescence, because it has no from-root anymore.

We note that an arborescence rooted from r is basically the same as a tree, whose all edges have been "oriented away from r". More precisely:

Theorem 1.3.4. Let D be a multidigraph, and let r be a vertex of D. Then, the following two statements are equivalent:

- **Statement C1:** The multidigraph D is an arborescence rooted from r.
- **Statement C2:** The undirected multigraph D^{und} is a tree, and each arc of D is "oriented away from r" (this means the following: the source of this arc lies on the unique path between r and the target of this arc on D^{und}).

Proof. We will prove this next time. □

Here is another bunch of equivalent criteria for arborescences, imitating the tree equivalence theorem:

Theorem 1.3.5 (The arborescence equivalence theorem)**.** Let $D = (V, A, \psi)$ be a multidigraph with a from-root r. Then, the following six statements are equivalent:

- **Statement A1:** The multidigraph D is an arborescence rooted from r.
- **Statement A2:** We have $|A| = |V| - 1$.
- **Statement A3:** The multigraph D^{und} is a tree.
- **Statement A4:** For each vertex $v \in V$, the multidigraph D has a unique walk from r to v.
- **Statement A5:** If we remove any arc from D, then the vertex r will no longer be a from-root of the resulting multidigraph.
- **Statement A6:** We have $\deg^- r = 0$, and each $v \in V \setminus \{r\}$ satisfies $\deg^- v = 1$.

Proof. We will prove the implications A1$\Longrightarrow$A4$\Longrightarrow$A5$\Longrightarrow$A6$\Longrightarrow$A2$\Longrightarrow$A3$\Longrightarrow$A1. Since these implications form a cycle that includes all six statements, this will entail that all six statements are equivalent.

Before we prove these implications, we introduce a notation: If a is any arc of D, then $D \setminus a$ shall denote the multidigraph obtained from D by removing this arc a. (Formally, this means that $D \setminus a := \left(V,\ A \setminus \{a\},\ \psi\mid_{A \setminus \{a\}}\right)$.)

We now come to the proofs of the promised implications.

Proof of the implication A1$\Longrightarrow$A4: Assume that Statement A1 holds. Thus, D is an arborescence rooted from r. In other words, r is a from-root of D and the undirected multigraph D^{und} has no cycles.

We must show that for each vertex $v \in V$, the multidigraph D has a unique walk from r to v. The existence of such a walk is clear (because r is a from-root of D). It is the uniqueness that we need to prove.

Assume the contrary. Thus, there exists a vertex $v \in V$ such that two distinct walks $\mathbf{u}$ and $\mathbf{v}$ from r to v exist. However, the multigraph D has no loops (since any loop of D would be a loop of D^{und}, and thus create a cycle of D^{und}, but we know that D^{und} has no cycles). Hence, any walk of D is automatically a backtrack-free walk of D^{und} (indeed, it is backtrack-free because the only way two consecutive arcs of a walk in a **digraph** can be equal is if they are loops). Therefore, the two walks $\mathbf{u}$ and $\mathbf{v}$ of D are two backtrack-free walks of D^{und}.

Thus, there are two distinct backtrack-free walks from r to v in D^{und} (namely, $\mathbf{u}$ and $\mathbf{v}$). Theorem 1.1.3 from Lecture 13 thus lets us conclude that D^{und} has a cycle. But this contradicts the fact that D^{und} has no cycles.

This contradiction shows that our assumption was wrong. Hence, we have proved that for each vertex $v \in V$, the multidigraph D has a unique walk from r to v. In other words, Statement A4 holds.

Proof of the implication A4$\Longrightarrow$A5: Assume that Statement A4 holds.

Let now a be any arc of D. We shall show that r is not a from-root of the multidigraph $D \setminus a$.

Indeed, let s be the source and t the target of the arc a. We shall show that the digraph $D \setminus a$ has no path from r to t.

Indeed, assume the contrary. Thus, $D \setminus a$ has some path $\mathbf{p}$ from r to t. This path does not use the arc a (since it is a path of $D \setminus a$).

On the other hand, we have assumed that Statement A4 holds. Applying this statement to $v = s$, we conclude that the multidigraph D has a unique walk from r to s. Let $(v_0, a_1, v_1, a_2, v_2, \ldots, a_k, v_k)$ be this walk. By appending the arc a and the vertex t to its end, we extend it to a longer walk

$$(v_0, a_1, v_1, a_2, v_2, \ldots, a_k, v_k, a, t),$$

which is a walk from r to t. We denote this walk by $\mathbf{q}$.

We have now found two walks from r to t in the digraph D: namely, the path $\mathbf{p}$ and the walk $\mathbf{q}$. These two walks are distinct (since $\mathbf{q}$ uses the arc a, but $\mathbf{p}$ does not). However, Statement A4 (applied to $v = t$) yields that the multidigraph D has a **unique** walk from r to t. This contradicts the fact that we just have found two distinct such walks.

This contradiction shows that our assumption was false. Hence, the digraph $D \setminus a$ has no path from r to t. Thus, r is not a from-root of $D \setminus a$.

Forget that we fixed a. We have now proved that if a is any arc of D, then r is not a from-root of $D \setminus a$. In other words, if we remove any arc from D, then the vertex r will no longer be a from-root of the resulting multidigraph. Thus, Statement A5 holds.

Proof of the implication A5$\Longrightarrow$A6: Assume that Statement A5 holds. We must prove that Statement A6 holds. In other words, we must prove that $\deg^- r = 0$, and that each $v \in V \setminus \{r\}$ satisfies $\deg^- v = 1$.

Let us first prove that $\deg^- r = 0$. Indeed, assume the contrary. Thus, $\deg^- r \neq 0$, so that there exists an arc a with target r. We shall show that r is a from-root of $D \setminus a$.

The arc a has target r. Thus, a path that starts at r cannot use this arc a (because this arc would lead it back to r, but a path is not allowed to revisit any vertex), and therefore must be a path of $D \setminus a$. Thus we have shown that any path of D that starts at r is also a path of $D \setminus a$. However, for each vertex v of D, the digraph D has a path from r to v (since r is a from-root of D). This path is also a path of $D \setminus a$ (since any path of D that starts at r is also a path of $D \setminus a$). Thus, for each vertex v of $D \setminus a$, the digraph $D \setminus a$ has a path from r to v. In other words, r is a from-root of $D \setminus a$. However, we have assumed that Statement A5 holds. Thus, in particular, if we remove the arc a from D, then the vertex r will no longer be a from-root of the resulting multidigraph. In other words, r is not a from-root of $D \setminus a$. But this contradicts the fact that r is a from-root of $D \setminus a$.

This contradiction shows that our assumption was false. Hence, $\deg^- r = 0$ is proved.

Now, let $v \in V \setminus \{r\}$ be arbitrary. We must show that $\deg^- v = 1$.

Indeed, assume the contrary. Thus, $\deg^- v \neq 1$. Using the fact that r is a from-root of D, it is thus easy to see that $\deg^- v \geq 2$ [6]. Hence, there exist two distinct arcs a and b with target v. Consider these arcs a and b.

We are in one of the following three cases:

Case 1: The digraph $D \setminus a$ has a path from r to v.

Case 2: The digraph $D \setminus b$ has a path from r to v.

Case 3: Neither the digraph $D \setminus a$ nor the digraph $D \setminus b$ has a path from r to v.

Let us first consider Case 1. In this case, the digraph $D \setminus a$ has a path from r to v. Let $\mathbf{p}$ be such a path.

We have assumed that Statement A5 holds. Thus, in particular, if we remove the arc a from D, then the vertex r will no longer be a from-root of the resulting multidigraph. In other words, r is not a from-root of $D \setminus a$. In other words, there exists a vertex $w \in V$ such that the digraph $D \setminus a$ has no path from r to w (by the definition of a "from-root"). Consider this vertex w.

The digraph D has a path $\mathbf{q}$ from r to w (since r is a from-root of D). Consider this path $\mathbf{q}$. If the path $\mathbf{q}$ did not use the arc a, then it would be a path of $D \setminus a$ as well, but this would contradict the fact that $D \setminus a$ has no path from r to w. Thus, the path $\mathbf{q}$ must use the arc a.

Consider the part of $\mathbf{q}$ that comes after the arc a. This part must be a path from v to w (since the arc a has target v, whereas the path $\mathbf{q}$ has ending point w). Let us denote this path by $\mathbf{q}'$. Thus, the path $\mathbf{q}'$ does not use the arc a (since it was defined as the part of $\mathbf{q}$ that comes after a). Hence, $\mathbf{q}'$ is a path of $D \setminus a$.

Now, we know that the digraph $D \setminus a$ has a path $\mathbf{p}$ from r to v as well as a path $\mathbf{q}'$ from v to w. Splicing these paths together, we obtain a walk $\mathbf{p} * \mathbf{q}'$ from r to w. So we know that $D \setminus a$ has a walk from r to w. According to Corollary 1.1.5 from Lecture 8, we thus conclude that $D \setminus a$ has a path from r to w. This contradicts the fact that $D \setminus a$ has no path from r to w.

We have thus obtained a contradiction in Case 1.

[6] *Proof.* Since r is a from-root of D, we know that the digraph D has a path from r to v. Since $v \neq r$ (because $v \in V \setminus \{r\}$), this path must have at least one arc. The last arc of this path is clearly an arc with target v. Thus, there exists at least one arc with target v. In other words, $\deg^- v \geq 1$. Combining this with $\deg^- v \neq 1$, we obtain $\deg^- v > 1$. In other words, $\deg^- v \geq 2$.

The same argument (but with the roles of a and b interchanged) results in a contradiction in Case 2.

Let us finally consider Case 3. In this case, neither the digraph $D \setminus a$ nor the digraph $D \setminus b$ has a path from r to v. However, the digraph D has a path $\mathbf{p}$ from r to v (since r is a from-root of D). Consider this path $\mathbf{p}$. If this path $\mathbf{p}$ did not use the arc a, then it would be a path of $D \setminus a$, but this would contradict our assumption that the digraph $D \setminus a$ has no path from r to v. Thus, this path $\mathbf{p}$ must use the arc a. For a similar reason, it must also use the arc b. However, the two arcs a and b have the same target (viz., v) and thus cannot both appear in the same path (since a path cannot visit a vertex more than once). This contradicts the fact that the path $\mathbf{p}$ uses both arcs a and b. Hence, we have found a contradiction in Case 3.

We have now found contradictions in all three Cases 1, 2 and 3. This contradiction shows that our assumption was false. Hence, $\deg^- v = 1$ is proved.

We have now proved that each $v \in V \setminus \{r\}$ satisfies $\deg^- v = 1$. Since we have also shown that $\deg^- r = 0$, we thus have proved Statement A6.

Proof of the implication A6$\Longrightarrow$A2: Assume that Statement A6 holds. We must prove that Statement A2 holds. However, Proposition 2.2.3 from Lecture 9 yields

$$\begin{aligned} |A| = \sum_{v \in V} \deg^- v &= \underbrace{\deg^- r}_{\substack{=0 \\ \text{(by Statement A6)}}} + \sum_{v \in V \setminus \{r\}} \underbrace{\deg^- v}_{\substack{=1 \\ \text{(by Statement A6)}}} \\ &= 0 + \sum_{v \in V \setminus \{r\}} 1 = \sum_{v \in V \setminus \{r\}} 1 = |V \setminus \{r\}| = |V| - 1. \end{aligned}$$

Hence, Statement A2 holds.

Proof of the implication A2$\Longrightarrow$A3: Assume that Statement A2 holds. We must prove that Statement A3 holds.

For each $v \in V$, the digraph D has a path from r to v (since r is a from-root of D). Thus, for each $v \in V$, the graph D^{und} has a path from r to v (since any path of D is a path of D^{und}). Therefore, any two vertices u and v of D^{und} are path-connected in D^{und} (because we can get from u to v via r, according to the previous sentence). Therefore, the graph D^{und} is connected (since it has at least one vertex[7]). Moreover, its number of edges is $|A| = |V| - 1$ (by Statement A2). Therefore, the multigraph D^{und} satisfies the Statement T4 of the tree equivalence theorem (Theorem 1.2.4 in Lecture 13). Consequently, it satisfies Statement T1 of that theorem as well. In other words, it is a tree. This proves Statement A3.

Proof of the implication A3$\Longrightarrow$A1: Assume that Statement A3 holds. We must prove that Statement A1 holds.

[7] This is because $r \in V$.

The multigraph D^{und} is a tree (by Statement A3), and thus is a forest; hence, it has no cycles. Since we also know that r is a from-root of D, we thus conclude that D is an arborescence rooted from r (by the definition of an arborescence). In other words, Statement A1 is satisfied.

We have now proved all six implications in the chain
A1$\Longrightarrow$A4$\Longrightarrow$A5$\Longrightarrow$A6$\Longrightarrow$A2$\Longrightarrow$A3$\Longrightarrow$A1. Thus, all six statements A1, A2, . . ., A6 are equivalent. This proves Theorem 1.3.5. □

1.4. Teaser

Next time, we will show the following:

- Theorem 1.3.4 (this is pretty easy, but requires some notation).
- Each multidigraph with a root r has a spanning arborescence rooted from r. (This is an analogue of Theorem 1.1.1 for digraphs.)
- The **BEST theorem**: Let $D = (V, A, \psi)$ be a balanced multidigraph (i.e., we have $\deg^- v = \deg^+ v$ for each $v \in V$) such that each vertex has indegree > 0. Fix an arc a of D, and let r be its target. Let $\tau(D, r)$ be the number of spanning arborescences of D rooted from r. Let $\varepsilon(D, a)$ be the number of Eulerian circuits of D whose last arc is a. Then,

  $$\varepsilon(D, a) = \tau(D, r) \cdot \prod_{u \in V} \left(\deg^- u - 1\right)!.$$

  Combined with a formula for $\tau(D, r)$ that we will prove later, this makes $\varepsilon(D, a)$ efficiently computable!

References

[BenWil06] Edward A. Bender, S. Gill Williamson, *Foundations of Combinatorics with Applications*, Dover 2006.
`https://mathweb.ucsd.edu/~ebender/CombText/index.html`