MA2C03 - DISCRETE MATHEMATICS - TUTORIAL NOTES
Brian Tyrrell
09/02/2017

Summary: Sometimes we can't see the forest for the trees. However sometimes we actually want to look at trees instead of forests - this is what we did this week.

1 Circuits

Recall what we're attempting to prove:

Corollary 1.1. *If the degree of every vertex of a graph is even, then the graph admits an Eulerian circuit.*

We can work through the proof systematically in a series of lemmas:

Lemma 1.1. *If the degree of each vertex is even, then there exists a circuit in the graph.* ■

Lemma 1.2. *If the degree of each vertex is even and if there exist edges not in the circuit incident to a vertex in the circuit, we can construct another circuit.*

Moral of the proof:
We look at the subgraph of edges ignored by the circuit. We are the same scenario as *Lemma 1.1*, so we can construct another ('independent') circuit, as required. ■

Lemma 1.3. *If we have two circuits with at least one vertex in common, we can combine them.*

Moral of the proof:
We are assuming no edges of the graph are traversed by both circuits; i.e. we're in the same scenario as *Lemma 1.2*. Then we can concatenate the two circuits at the joint vertex. ■

Finally, the most complicated to state and visualise:

Lemma 1.4. *Let (V, E) be a connected graph, and let some trail T in this graph be given. Suppose that* ***no*** *vertex of the graph has the property that* not all the edges *of the graph incident to that vertex are traversed by T.*
Then the given trail is an Eulerian trail.

Moral of the proof:
That is, there does not exist a vertex v such that there exists an edge coming from v not traversed

by T. So let V_1 be the set of vertices through which the trail passes, and V_2 the set of vertices through which the trail doesn't pass. Our goal is to show V_2 is empty.

We show no edge in V_1 can join an edge in V_2, which is a contradiction as the graph is connected. Thus $V_2 = \emptyset$ as required. ■

Using these, the main theorem (**Euler's Theorem**) is proven:

Theorem 1.2. *A nontrivial* ***connected*** *graph contains an Eulerian circuit if the* ***degree of every vertex of the graph is even****.*

bold = conditions needed for an Eulerian circuit.

Remark 1.1. By *Corollary 1.1* this is an 'if and only if' condition on connected graphs.

Proof.
By *Lemma 1.1*, we have a circuit. We can consider the circuit of maximum length. If the circuit is not Eulerian by *Lemma 1.2* we can construct a second circuit about a vertex v **on the circuit** - i.e., there is a vertex on the circuit such that not all the edges incident to it are traversed by this circuit (thus leaving us edges to spare to construct the second circuit). By *Lemma 1.3* we can concatenate, gaining a larger circuit - a contradiction to maximality.

We conclude **no** vertex that belongs to the circuit of maximal length has the property that *not all edges* incident to it are traversed by the circuit of maximal length. By *Lemma 1.4*, the maximal circuit is Eulerian, as required - the previous sentence shows why we need the specific and difficult formulation of *Lemma 1.4*. ■

2 Forests and Trees

Definition 2.1. We'll quickly parse through the following terms:

(1) Call a graph a *forest* if it contains **no** circuits (AKA *acyclic*).

(2) Call a graph a *tree* if it is a connected forest.

Theorem 2.1. *Every forest contains one isolated* $(\deg = 0)$ *or pendant* $(\deg = 1)$ *vertex.*

Proof.
If $\deg v \geq 2$ for every vertex v then by a previous theorem the graph admits a circuit, a contradiction. ■

Theorem 2.2. *A (non-trivial) tree contains at least one pendant vertex.*

Moral of the proof:
The tree is non trivial, so contains at least two vertices. The tree must be connected, so no isolated vertices. As a tree is a forest, it has a pendant vertex by *Theorem 2.1*. ■

Theorem 2.3. *Let (V, E) be a tree. Suppose v, w are two distinct vertices. There exists a unique path from v to w in (V, E).*

Moral of the proof:
If (V, E) contains two distinct paths, it contains a circuit (by a previous proposition); a contradiction. ■

QUESTION: Given a graph, how can we make a tree from it?
ANSWER: By studying spanning trees!

Definition 2.2. A *spanning tree* in a graph (V, E) is a subgraph of the graph (V, E) which itself is a tree and contains every vertex in V.

Theorem 2.4. *Every connected graph (V, E) contains a spanning tree.* ■

The proof of this theorem is not constructive; that is, it doesn't *provide* you with the spanning tree in its proof, it just proves the spanning tree exists. However we can convert this theorem into an algorithm that will give us the spanning tree of a graph.

First, we will need one result; the "⇒" direction comes from an inductive proof and the "⇐" direction comes from *Theorem 2.4*.

Theorem 2.5. *(V, E) is a tree $\Leftrightarrow \#(E) = \#(V) - 1$.* ■

Let (V, E) be a connected graph.

Algorithm #1 Delete edges, one at a time, to remove all circuits.

Algorithm #2 Start with solely one edge vw. Add back in one edge per vertex $v \in V \setminus \{v, w\}$ such that at each step the subgraph of (V, E) is connected and a tree.

I call Algorithm #1 the '**destructive approach**' and Algorithm #2 the '**constructive approach**'.

At the end of Algorithm #1, note we use *Theorem 2.5* to check if in fact we have a tree. At every stage i of Algorithm #2, we stick to the '1 vertex and 1 edge' rule to ensure $\#(E_i) = \#(V_i) - 1$.

This also guarantees we don't accidentally make a circuit (as then we wouldn't have to add in a vertex) and the graph is connected at each stage (as we only add *one vertex* and need to draw *one edge*).

(Examples of both given in class)