15th Lecture: 24/2

Class Equation in Groups. Before ending this second part of the course, we mention the simplest application of Burnside's Lemma within the theory of groups itself.

Recall from the proof of Theorem 14.5 that every group acts on itself via the group multiplication (from left or right). There is another natural right action of a group on itself given by[1]

$$(s,\, g) \mapsto g^{-1}sg := s^g.$$

This is indeed a right action since

(1) $s^1 = 1^{-1}s1 = 1s1 = s$

(2) $(s^g)^h = h^{-1}(s^g)h = h^{-1}(g^{-1}sg)h = (h^{-1}g^{-1})s(gh) = (gh)^{-1}s(gh) = s^{gh}$.

This action is called *conjugation (from the right)* in G. The element $g^{-1}sg$ is called the *conjugate of s by g*. The orbits of the action are called the *conjugacy classes* of G.

The standard notation for the number of conjugacy classes in a group G is $k(G)$. Now if G is finite then, by Burnside's Lemma,

$$k(G) = \frac{1}{|G|}\sum_{g\in G}|F_g(G)|. \tag{15.1}$$

By definition,

$$F_g(G) = \{s \in G : g^{-1}sg = s\} = \{s \in G : sg = gs\} = \{s \in G : [s,\, g] = 1\}.$$

In other words, F_g consists of those elements of G which commute with g. As noted in Homework 2, Exercise 10, this set is in fact a subgroup of G, and usually denoted $C_G(g)$. It is called the *centralizer* of g in G. Thus we can rewrite (15.1) as

$$k(G) = \frac{1}{|G|}\sum_{g\in G}|C_G(g)|. \tag{15.2}$$

This is called the *(conjugacy) class equation* for the group G. A few remarks about it:

1. Another way to look at (15.1) and (15.2) is that, since G is acting on itself by conjugation, fixed point sets are the same thing as stabilizers[2]. In other words, $C_G(g) = \mathrm{Stab}_G(g)$. Hence, (15.2) follows more directly from (14.1) in this setting than in the general case of Burnside's Lemma, since if we substitute (14.1) into the right hand side of (15.2) we get the sum $\sum_{g\in G}\frac{1}{|g^G|}$. As in the proof of Burnside's Lemma, we can directly see that this sum equals $k(G)$ since each conjugacy class will contribute one to the sum.

2. If we divide both sides of (15.2) by $|G|$ we get

$$\frac{k(G)}{|G|} = \frac{1}{|G|^2}\sum_{g\in G}|C_G(g)|. \tag{15.3}$$

The RHS of (15.3) has a natural interpretation as the *probability* that a randomly chosen ordered pair of elements of G commute. For the number of possible pairs is $|G|^2$, by

[1] The corresponding left action is $(s,\, g) \mapsto gsg^{-1}$. As usual, just be consistent !

[2] More precisely, this relies on the fact that g commutes with s if and only if s commutes with g.

MP, and the sum just counts the total number of pairs that commute. This probability is commonly denoted $\Pr(G)$ and called the *commuting probability* of G. It is a rational number in $(0,\, 1]$ and can be thought of as a kind of "measure" as to how close G is to being abelian, since $\Pr(G) = 1$ if and only if G is abelian. Hence (15.3) says that

$$\Pr(G) = \frac{k(G)}{|G|}. \tag{15.4}$$

3. The obvious use of (15.2) is as a formula for computing $k(G)$. Perhaps the most important situation where this invariant of a finite group arises is in the so-called *representation theory of finite groups*. The fundamental theorem in that field states that the number of so-called *irreducible* representations of a finite group G equals $k(G)$ and that the sum of the squares of their so-called *degrees* equals $|G|$. It is beyond the scope of this course to explore such matters any further, but it's standard material if you wish to look it up.

4. Sometimes you will see the class equation (15.2) written somewhat differently. As shown in Proposition 14.8, given any action of a group G on a set S we get a partition of the latter into G-orbits. Assume both G and S are finite. If we choose one representative from each orbit, say $s_1,\, s_2,\, \ldots,\, s_k$, where k is the total number of orbits, then

$$|S| = \sum_{i=1}^{k} |s_i G|.$$

Applying this to the case of a finite group G acting on itself by conjugation, we get

$$|G| = \sum_{i=1}^{k(G)} |g_i^G| = \sum_{i=1}^{k(G)} \frac{|G|}{|C_G(g_i)|}. \tag{15.5}$$

Recall (see Demo3) that the *center* $Z(G)$ of a group G is defined as

$$Z(G) = \{g \in G : [g,\, h] = 1\ \forall\, h \in G\}.$$

I.e.: $Z(G)$ consists of those elements of G that commute with everything. It is easy to show that $Z(G)$ is a subgroup of G, so in the case of a finite group, by Theorem 9.13, $|Z(G)|$ divides $|G|$. Moreover, $g \in Z(G)$ if and only if g is in an orbit by itself under conjugation. Thus we can write (15.4) as

$$|G| = |Z(G)| + \sum_{g_i \notin Z(G)} \frac{|G|}{|C_G(g_i)|}, \tag{15.6}$$

where the sum is taken over one representative of each conjugacy class outside $Z(G)$. Eq. (15.6) is also referred to as the *class equation* of G. See, for example, Homework 2, Exercise 11(c) for an application.

Part 3: Graph Theory

Thre's a lot of basic notation and terminology in graph theory, which I don't want to spend lots of time going through because (a) you have probably encountered quite a bit of it before (b) all the terms are easy to understand intuitively so it probably suffices to just use them as we encounter them (c) you can always read through the first few pages of Chapter 6, Vol. 1 of the book for yourselves.

I will, however, throughout these notes consistently use the five following terms in the same way as they are used in the book - see Pages 141-142 of Vol. 1. The five terms all refer to sequences of edges in graphs, but with various restrictions.

Walk: A walk in a graph is an arbitrary sequence of edges such that the endpoint of one edge is the starting point of the next one. So "walking" is the most natural thing one can do in a graph. There are no restrictions as to how you can "walk", except that you can't "jump".

Trail: A trail is a walk that doesn't use the same edge twice.

Path: A path is a trail that doesn't visit any node more than once.

Circuit: A circuit is a closed trail, i.e.: a trail that starts and ends at the same node.

Cycle: A cycle is a closed path.

According to folklore, graph theory originated in the *Bridges of Königsberg* problem. The legend asserts that Euler, who at the time was a professor in Saint Petersburg, was visiting Königsberg (Kaliningrad) and heard about the problem from the locals. The question was whether it was possible to take a walk around town in such a way that one crossed every bridge exactly once - see Figure 15.1. It's quite easy to work out that the answer is No, especially after one boils he problem down to its esstnials, by modelling it with a (multi)graph, as in Figure 15.2. Then the issue is whether the graph can be drawn without lifting pen from paper. Being a mathematician, Euler wondered whether there was a general criterion for deciding for which graphs this was possible. The resulting theorem is considered the first theorem in graph theory (though it has competition from Euler's theorem on plane graphs, which we might come to later).

Definition 15.1. An *Euler trail (resp. circuit)* in a (undirected, multi-)graph is a trail (resp. circuit) that uses every edge exactly once.

Theorem 15.2. *Let G be a (undirected, multi-)graph. Then*
(i) G has an Euler circuit if and only if G is connected and every vertex of G has even degree.
(ii) G has an Euler trail between distinct nodes v and w if and only if G is connected, $deg(v)$ and $deg(w)$ are odd and every other vertex has even degree.

PROOF: That the conditions are necessary is clear. Since one cannot use any edge

more than once, each visit to a node must use up two of the edges incident to it. Hence the total number of edges incident to the node must be even. The only exceptions are if the trail starts and finishes in two different nodes v and w, since then the first time one exits v only one edge is used and similarly for the last time one enters w. Hence the degree of each of these nodes must be odd.

We now prove sufficiency for part (i). The proof for part (ii) is completely analogous, and we will briefy address this at the end. We prove, by strong induction on the number m of edges in G, that a connected (multi)graph in which every vertex has even degree possesses an Euler circuit. If $m = 0$, then the result is trivial: just do nothing ! So suppose the result holds for all graphs on at most m edges and let G be a connected graph on $m + 1$ edges in which every vertex has even degree.

Pick any vertex v and start walking from v - since G is connected, you can at least get going. Keep walking at random, but ensuring not to use any edge twice. Stop when you get stuck - this happens when you enter a vertex and have already utilised all the incident edges so there is no way out. Now, since you've been careful not to use any edge twice, each visit to a vertex has so far used a pair of incident edges. Since every vertex has even degree, this means that you cannot get stuck anywhere except at the starting point v.

Now if, when you get stuck, you have walked along every edge in G, then your walk is already an Euler circuit and we're done. Otherwise, let G^* be the graph on the same vertex set as G but with all the edges in this first closed walk removed. This first walk used up an even number of edges through each vertex it visited, including v since it returned there. Hence in G^* every vertex still has even degree. On the other hand there is a priori no guarantee that G^* will be connected. Say that it is the disjoint union of connected components $G_1, \ldots, G_k$. Then in each G_i it is still the case that every vertex has even degree. Thus, by the induction hypothesis, each G_i contains an Euler circuit.

Finally then, we obtain an Euler circuit in G by following the initial circuit which started and ended at v and inserting the Euler circuits in the G_i as "detours". More precisely, follow the initial circuit but insert the Euler circuit in G_i the first time you hit a vertex in that component. Clearly, this gives an Euler circuit in G.

For part (ii), the proof is completely analogous. One begins walking at random from one of the two nodes of odd degree. The only place one can get stuck is at the other node of odd degree. When this initial trail is removed from the graph, each vertex in what remains has even degree, so we can once again insert as detours Euler circuits from the components of what remains.

Remark 15.3. Note that

(i) the statement of Theorem 15.2 gives a criterion for existence or not of an Euler circuit/trail which is simple to check - you just have to count the number of edges incident to each vertex. Hence the algorithmic *decision problem*, the yes/no problem of whether an Euler circuit/trail exists in a graph given as input, is simple to solve

(ii) moreover, the proof of Theorem 15.2 is *constructive*, i.e.: it describes an algorithm for actually finding an Euler circuit/trail in a graph for which the decision algorithm returns "yes". It's again a very simple procedure which just involves recursively making random walks, removing them, making new random walks in what remains and finally putting it all together in a nested set of detours. This is an example of a so-called

greedy algorithm, since you can always get away with picking off as much as you can chew. No matter how you walk, there will always be a way to insert detours if you get stuck.

As we will see in coming lectures, in graph theory many problems are algorithmical in nature and come in two parts:

Decision Problem: The yes/no problem of deciding whether a given input graph G has a certain property or not. One wants a general procedure which can give the answer as quickly as possible.
Search Problem: If the property in question is the existence of a certain type of configuration, then one would like to actually find such a configuration when the decision program answers "yes".

The problem of Euler circuits/trails is an example where both the decision problem and the search problem have very efficient algorithmic solutions. Indeed, the proof of the correctness of the decision criterion itself yields a solution to the search problem. As we will see, this happens in many situations. But equally importantly, there are many examples of simply stated decision problems which seem to be very hard. Graph theory is a rich source of simply stated, but algorithmically difficult problems.

Finally today we note a simple but useful general fact about graphs:

Theorem 15.4. (Degree Equation) *In any (multi)graph $G = (V,\ E)$ one has*

$$\sum_{v \in V} deg(v) = 2 \cdot |E|. \tag{15.7}$$

In particular, every graph possesses an even number of vertices of odd degree.

PROOF: When we sum the degrees of the vertices, every edge of G will be counted twice since it is incident to two vertices.

Example 15.5. (NFL Problem) This is apparently (based on) a true story. NFL stands for National Football League, the organisation which runs professional American football. At some point in the 1960s, the league consisted of 26 teams, divided evenly into two so-called conferences, AFC and NFC. The league bosses wanted to make a schedule for the upcoming season in which

Each team would play 14 matches, of which 11 would be against teams in their own conference and 3 against teams in the other conference. There is no requirement that teams meet at most once.

It turns out (and was a big fiasco at the time !) that no such schedule is possible. Note that the requirement that each team play 3 games against teams in the other conference *can* be satisfied - I leave it an exercise for you to make such a schedule. The problem is the games within a conference.

Take one of the conferences, say AFC, and suppose a schedule meeting our demands existed. Now consider the graph $G = (V, E)$ in which V is the set of 13 AFC teams and each edge represents a match in the schedule. Then G would be a graph with 13 nodes, each of degree 11, which contradicts Theorem 15.4.