CCS Discrete III | Professor: Padraic Bartlett

Lecture 3: Algebra and Graphs

Weeks 5-6 | *UCSB 2015*

We turn here to a different aspect of graph theory; the intersection of **graphs** and **algebra**! We start by something that's a fairly natural transition from our work with flows over the past few weeks: the concept of an **algebraic** flow!

1 Algebraic Flows

1.1 Definitions and Fundamental Results

Definition. Let A be an abelian group. An **A-circulation** on a graph G is any function $f : V(G) \times V(G) \to A$ with the following properties:

1. For any $x, y \in V(G)$ with $\{x, y\} \notin E(G)$, we have $f_{xy} = 0$.
2. For any $x, y \in V(G)$ we have $f_{xy} = -f_{yx}$.
3. For any vertex $x \in V(G)$, we satisfy Kirchoff's law at x: that is,

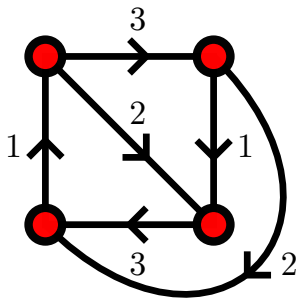
$$\sum_{y \in N(x)} f_{x,y} = 0.$$

This is kind-of like a flow from before, except that we don't have a capacity function, nor do we have special source/sink vertices; we instead require our graph to satisfy Kirchoff's law everywhere!

Definition. Let A be an abelian group. An **A-flow** on a graph G is an A-circulation where $f_{xy} \neq 0$ for any edge $\{x, y\}$ in $E(G)$.

An A-flow is simply a circulation where we make sure that our flow along every edge is nonzero; in other words, the flow doesn't "pretend" some of our edges are not there.

A natural question to ask here is the following: for a given group A, when does a graph G admit an A-flow? For example, the graph K_4 admits a $\mathbb{Z}/4\mathbb{Z}$ flow, as drawn below:



(As before, to define a flow on a graph, we place orientations on our edges and label our edges with the flow along that edge in that orientation.)

However, you can easily check (do this on pen and paper!) that G does not admit a $\mathbb{Z}/3\mathbb{Z}$ flow; given any way to label our edges with 1's and 2's, it's impossible to actually label everything consistently in a way that satisfies Kirchoff's law.

Surprisingly, it turns out that A itself doesn't really matter here! In particular, the only aspect of A that matters is the size of A, as we state in the following theorem:

Theorem. Take any two abelian groups A, B, with $|A| = |B|$. Then a graph G admits an A-flow if and only if it admits a B-flow.

We prove this with the following stronger result:

Theorem 1. *For any graph G, there is a polynomial P such that for any abelian group A, the number of distinct A-flows on G is given by $P(|A|)$. As a consequence, the only thing that matters for determining whether or not a graph admits a A-flow is the size of the abelian group A.*

Proof. We actually prove this claim for all multigraphs, instead of just graphs — that is, we let our graph G also have loops and multiple edges. We prove our claim by induction on the number of non-loop edges in G. (A question you might have, in this setting, is how loops interact with flows. Intuitively, we would think that any flow leaving a vertex on a loop edge must also return to that vertex on a loop edge: accordingly, we can really put **any value** on a loop edge without changing whether or not we satisfy Kirchoff's laws!)

Our base case is when our graph only has loops. In any such graph, as discussed above, we can put any non-zero value on each loop edge and still satisfy Kirchoff's laws. There are $|A| - 1$ many such values; so if there are m loops, then there are $(|A| - 1)^m$ many A-flows on this graph, which is clearly a polynomial in $|A|$.

For induction: suppose that our graph has an edge $e_0 = \{x, y\}$ between two distinct vertices. Consider the two graphs $G \setminus \{e_0\}$, where you just delete the edge, and $G/\{e_0\}$, where you contract the edge e_0. (If you've forgotten how edge contraction works, check out our notes on the four-color theorem where we introduced this definition!)

To each of these graphs, apply our inductive hypothesis, to get a polynomial $P_1(x)$ for the graph $G/\{e_0\}$ and another polynomial $P_2(x)$ for the graph $G \setminus \{e_0\}$. Now, make the following observations:

- There is a 1-1 correspondence between A-flows on $G \setminus \{e_0\}$ and A-circulations on G where $f_{xy} = 0$ and $f_{vw} \neq 0$ for any other edge $\{v, w\}$; to see this, take any such A-circulation on G and simply copy it over onto $G \setminus \{e_0\}$! It's still a circulation, as we haven't changed any of our values on our non- $\{x, y\}$ edges and therefore still satisfy Kirchoff's laws; it's also now a flow, as we deleted the only edge with a zero flow on it!

- Similarly, there is a 1-1 correspondence between A-flows on $G/\{e_0\}$ and A-circulations on G where $f_{vw} \neq 0$ for any edge $\{v, w\} \neq \{x, y\}$, and f_{xy} could be anything; either 0 or nonzero. (You are asked to prove this on the HW!)

By definition, we know that the number of A -flows on G is just the number of A-circulations where $f_{vw} \neq 0$ for any edge $\{v, w\} \neq \{x, y\}$, and $f_{xy} \neq 0$ as well; in other words, it's just $P_1(|A|) - P_2(|A|)$. So such a polynomial exists for G as well, and we've proven our claim! □

In a sense, we have reduced our study of A-flows to the study of $\mathbb{Z}/k\mathbb{Z}$-flows, which should make our life remarkably easier. One thing we might wonder, now, is whether we can turn these $\mathbb{Z}/k\mathbb{Z}$-flows into actual $\mathbb{Z}$-flows: i.e. whether by being clever, we can transform the weaker requirement of

$$\sum_{y \in N(x)} f_{xy} \equiv 0 \mod k$$

into the stronger requirement

$$\sum_{y \in N(x)} f_{xy} = 0 \mod k.$$

As it turns out, we can!

Definition. A k-flow f on a graph G is a $\mathbb{Z}$-flow such that $|f_{xy}| < k$ for every edge $\{x, y\}$.

Theorem 2. *A graph G has a k-flow iff it has a $\mathbb{Z}/k\mathbb{Z}$-flow.*

Proof. (Sketch:) Any k-flow on G satisfies the relation

$$\sum_{w \in N^+(v)} g(v, w) - \sum_{w \in N^-(v)} g(w, v) = 0.$$

at every vertex v. Therefore, this flow trivially induces a $\mathbb{Z}/k\mathbb{Z}$-flow by interpreting all of its values as elements of $\mathbb{Z}/k\mathbb{Z}$ (because if a sum of things is equal to 0, it's certainly equal to $0 \mod k$.)

To go the other way: take any flow $f : E(G) \to \mathbb{Z}/k\mathbb{Z}$. Interpret f as a function $E(G) \to \{1, \ldots k-1\}$. Then all we have to do is for every edge e, decide whether we map it to $f(e)$ or $f(e) - k$, in a sufficiently consistent way that we insure Kirchoff's laws are still obeyed. Doing this is an exercise for the homework! □

1.2 Some Calculations

To recap: we've proven that for an abelian group A with $|A| = k$, a graph G has an A flow iff it has a $\mathbb{Z}/k\mathbb{Z}$-flow iff it has a k-flow. Therefore, in a sense, the only interesting questions to be asked here is the following: given a graph G, for what values of k does G have a k-flow?

We classify some cases here:

Proposition 3. *A graph G has a 1-flow iff it has no edges.*

Proof. This is trivial, as a 1-flow consists of a mapping of G's edges to 0, such that no edge is mapped to, um, 0. □

Proposition 4. *A graph G has a 2-flow iff the degree of all of its vertices are even.*

Proof. A 2-flow consists of a mapping of G's edges to $\{\pm 1\}$, or equivalently a mapping of G's edges to 1 in $\mathbb{Z}/2\mathbb{Z}$, in a way that satisfies Kirchoff's law. However, we trivially satisfy Kirchoff's laws in the $\mathbb{Z}/2\mathbb{Z}$ case iff the degree of every vertex is even; so we know that this condition is equivalent to having a 2-flow. □

Definition. For later reference, call any graph where the degrees of all of its vertices are even a **even** graph; relatedly, call any graph where the degree of any of its vertices is 3 a **cubic** graph.

Proposition 5. *A cubic graph G has a 3-flow iff it is bipartite.*

Proof. A 3-flow consists of a mapping of G's edges to $\{1, 2\}$ in $\mathbb{Z}/3\mathbb{Z}$, in a way that satisfies Kirchoff's law. Suppose that we have some such flow on our graph: call it f.

Take any cycle $(v_1, v_2, , v_n)$ in this graph, and consider any two consecutive edges $(v_1, v_2), (v_2, v_3)$. Suppose f assigned these the same value, and let w be v_2's third distinct neighbor: then, by Kirchoff's law, we know that

$$-f_{v_1,v_2} + f_{v_2,v_3} + f_{v_2,w} = 0 \Rightarrow f_{v_2,w} = 0.$$

But f is a $\mathbb{Z}/3\mathbb{Z}$-flow: so this cannot happen! Therefore, the values 1 and 2 have to occur alternately on this cycle, and therefore it must have even length. Having all of your cycles be of even length is an equivalent condition to being bipartite, so we know that our graph must be bipartite.

To see the other direction: suppose that G is bipartite, with bipartition V_1, V_2. Define $f_{x,y} = 1$ and $f_{y,x} = -1 = 2$, for all $x \in V_1, y \in V_2$; this evaluation means that for any $x \in V_1$, we have

$$f_{x,y_1} + f_{x,y_2} + f_{x,y_3} = 1 + 1 + 1 \equiv 0 \mod 3$$

and for any $y \in V_2$, we have

$$f_{y,x_1} + f_{y,x_2} + f_{y,x_3} = 2 + 2 + 2 \equiv 0 \mod 3$$

by summing over their three neighbors in the other part of the partition. So this is a 3-flow, and we've proven the other direction of our claim. □

For simplicity's sake, we introduce the function $\varphi(G)$ to denote the smallest value of k for which G admits a k-flow: if no such value exists, we say that $\varphi(G) = \infty$.

We now list here a series of claims whose proofs we leave for the HW:

Proposition 6. $\varphi(K_2) = \infty$.

Proposition 7. *More generally, a connected graph G has $\varphi(G) = \infty$ whenever G has a **bridge**: i.e. there is an edge $e \in E(G)$ such that removing e from G disconnects G.*

Proposition 8. $\varphi(K_4) = 4$.

Proposition 9. *If n is even and not equal to 2 or 4, then $\varphi(K_n) = 4$.*

Proposition 10. *A graph G has a 4-flow if and only if we can write it as the **union** of two even graphs: i.e. there are a pair of graphs G_1, G_2, with possibly overlapping edge and vertex sets, such that $V(G) = V(G_1) \cup V(G_2), E(G) = E(G_1) \cup E(G_2)$.*

Proposition 11. *A cubic graph G has a 4-flow if and only if it is three-edge colorable.*

Corollary 12. *The Petersen graph has no 4-flow.*

On the HW, you (hopefully) showed that the Petersen graph does have a 5-flow earlier in the week; therefore, we know that $\varphi(\text{Pete}) = 5$.

1.3 Open Conjectures

Surprisingly, when you start computing more of these numbers, it seems pretty much impossible to find anything that's got a value of φ bigger than 5 and not yet infinity. This motivated Tutte to make the following conjectures on k-flows, which are still open to this day!

Conjecture 13. *Every bridgeless graph has a 5-flow.*

More surprisingly, it seems like the Petersen graph, in a sense, is unique amongst graphs that have 5-flows:

Conjecture 14. *Every bridgeless graph that doesn't contain the Petersen graph as a minor has a 4-flow.*

Strange!

2 Using Graphs to Study Algebra: Cayley Graphs

This is not the only intersection of graph theory and algebra! Instead of looking at what happens when we stick groups on top of existing graphs, we can think about ways to create graphs that correspond to given groups! Specifically, over the next few classes we will develop the concepts of **Cayley graphs** and **Schreier diagrams**, use them to study various kinds of groups, and from there prove some very deep and surprising theorems from abstract algebra!

In specific: this course kind-of has a natural split into two parts, (a) exploring the concepts that link groups and graphs, and (b) using those concepts to prove results! This section falls into the (a) camp; we're going to mostly study a large stack of definitions and examples here.

For the most part, I'm assuming everyone here remembers groups from the fall. However, there are some specific group concepts that I want people to specifically recall for this class: **free groups**, **generating sets**, **presented groups**, and **cosets**.

2.1 Preliminary Concepts

Definition. The **free group** on n generators $a_1, \dots a_n$, denoted

$$\langle a_1, \dots a_n \rangle,$$

is the following group:

- The elements of the group are all of the strings of the form

$$a_{i_1}^{k_1} a_{i_2}^{k_2} \dots a_{i_l}^{k_l},$$

  where the indices $i_1, \dots i_l$ are all valid indices for the $a_1, \dots a_n$ and the $k_1, \dots k_l$ are all integers.

- We also throw in an identity element e, which corresponds to the "empty string" that contains no elements.

- Given two strings s_1, s_2, we can **concatenate** these two strings into the word $s_1 s_2$ by simply writing the string that consists of the string s_1 followed by the string s_2.

- Whenever we have a^k in a string, we think of this as being $\overbrace{a \cdot a \cdot \dots a}^{k \text{ copies}}$, i.e. k copies of a. If we have multiple consecutive strings of a's, we can combine them together into one such a^k: for example, the word $a^3 a a^2$ is the same thing as the word a^6.

- Finally, if we ever have an aa^{-1} or an $a^{-1}a$ occurring next to each other in a string, we can simply replace this pairing with the empty string e.

For example, the free group on two generators $\langle a, b \rangle$ contains strings like

$$a^6 b^4 a^{-2} b^3 a^1, b^{12}, a^{-1} b^{-2} a^4 b, \dots$$

As described earlier, we concatenate strings by simply placing one after the other: i.e.

$$a^2 b^{-2} a^3 b a^3 \cdot a^{-3} b^{-1} a^1 b^3 = a^2 b^{-2} a^3 b a^3 a^{-3} b^{-1} a^1 b^3.$$

As described above, we typically simplify this right-hand string by canceling out terms and their inverses, and grouping together common powers of our generators:

$$a^2 b^{-2} a^3 b a^3 \cdot a^{-3} b^{-1} a^1 b^3 = a^2 b^{-2} a^3 \cancel{b} \cancel{a^3} \cancel{a^{-3}} \cancel{b^{-1}} a^1 b^3 = a^2 b^{-2} a^4 b^3$$

This is a group! In particular, concatenation is associative, the empty string e is clearly an identity, and we can "invert" any word $a_{i_1}^{k_1} a_{i_2}^{k_2} \dots a_{i_l}^{k_l}$ by simply reversing it and switching the signs on the k_i's: i.e.

$$\cancel{a_{i_1}^{k_1}} \cancel{a_{i_2}^{k_2}} \dots \cancel{a_{i_l}^{k_l}} \cdot \cancel{a_{i_l}^{-k_l}} \dots \cancel{a_{i_2}^{-k_2}} \cancel{a_{i_1}^{-k_1}} = e$$

Definition. Given a group G, we say that it is **generated** by some collection of elements $a_1, \dots a_n \in G$ if we can create any element in G via some combination of the elements $a_1, \dots a_n$ and their inverses. Note that some groups have multiple different sets of generators: i.e. $\langle \mathbb{Z}, + \rangle$ is generated both by the single element 1 and also by the pair of elements $\{2, 3\}$

Definition. In our above discussion, we have primarily defined groups by giving a set and an operation on that set. There are other ways of defining a group, though! A **group presentation** is a collection of n generators $a_1, \dots a_n$ and m words $R_1, \dots R_m$ from the free group $\langle a_1, \dots a_n \rangle$, which we write as

$$\langle a_1, \dots a_n \mid R_1, \dots R_m \rangle.$$

We associate this presentation with the group defined as follows:

- Start off with the free group $\langle a_1, \dots a_n \rangle$.
- Now, declare that within this free group, the words $R_1, \dots R_m$ are all equal to the empty string: i.e. if we have any words that contain some R_i as a substring, we can simply "delete" this R_i from the word.

You have actually seen some groups defined via a presentation before:

Example. Consider the group with presentation

$$\langle a \mid a^n \rangle.$$

This is the collection of all words written with one symbol a, where we regard $a^n = e$: i.e. it's just

$$e, a, a^2, a^3, \dots a^{n-1}.$$

This is because given any string $a^k \in \langle a \rangle$, we have $a^k = a^l$ for any $k \equiv l \mod n$. This is because we can simply concatenate copies of the strings a^n, a^{-n} as many times as we want without changing a string, as $a^n = e$!

You have seen this group before: this is just $\mathbb{Z}/n\mathbb{Z}$ with respect to addition, if you replace a with 1 and think of $\overbrace{11\dots1}^{k \text{ times}}$ as k.

Often, we will give a group with a presentation in the form

$$\langle a_1, \dots a_n \mid R_1 = R_2, R_3 = R_4, \dots, \dots R_{m-1} = R_m \rangle,$$

because it is easier sometimes to think of saying that certain kinds of words are equal rather than other kinds of words are the identity; this is equivalent to the group presentation

$$\langle a_1, \dots a_n \mid R_1(R_2)^{-1}, R_3(R_4)^{-1}, \dots, \dots R_{m-1}(R_m)^{-1} \rangle.$$

Definition. Suppose that G is a group, $s \in G$ is some element of G, and H is a subgroup of G. We define the **right coset** of H corresponding to s as the set

$$Hs = \{hs \mid h \in H\}.$$

We will often omit the "right" part of this definition and simply call these objects cosets.

Example. Consider the group $G = \langle \mathbb{Z}, + \rangle$. One subgroup of this group is the collection of all multiples of 5: i.e.

$$H = \{\dots -15, -10, -5, 0, 5, 10, 15 \dots\}$$

This subgroup has several cosets:

- $s = 0$: this forms the coset

  $$H + 0 = \{\dots -15, -10, -5, 0, 5, 10, 15 \dots\},$$

  which is just H itself.

- $s = 1$: this forms the coset

$$H + 1 = \{\ldots - 14, -9, -4, 1, 6, 11, 16 \ldots\}.$$

- $s = 2$: this forms the coset

$$H + 2 = \{\ldots - 13, -8, -3, 2, 7, 12, 17 \ldots\}.$$

- $s = 3$: this forms the coset

$$H + 3 = \{\ldots - 12, -7, -2, 3, 8, 13, 18 \ldots\}.$$

- $s = 4$: this forms the coset

$$H + 4 = \{\ldots - 11, -6, -1, 4, 9, 14, 19 \ldots\}.$$

Notice that this collection of cosets above is indeed the collection of **all** of the possible cosets of H within G: if we take any other element in $\mathbb{Z}$, like say 13, we'll get one of the five cosets above: i.e.

$$H + 13 = \{\ldots - 2, 3, 8, 13, 18 \ldots\} = H + 3.$$

In general, $H + x = H + y$ for any $x \equiv y \mod 5$.

Example. Consider the group $G = \langle(\mathbb{Z}/7\mathbb{Z})^\times, \cdot\rangle$, i.e. the nonzero integers mod 7 with respect to the multiplication operation. This has the set

$$H = \{1, 6\}$$

as a subgroup (check this if you don't see why!)

This group has the following cosets:

- $s = 1$, which creates the cosets $H \cdot 1 = H$,

- $s = 2$, which creates the coset

$$H \cdot 2 = \{2, 5\}.$$

- $s = 3$, which creates the coset

$$H \cdot 3 = \{3, 4\}.$$

- $s = 4$, which creates the coset

$$H \cdot 4 = \{4, 3\}.$$

Notice that this coset is the same as $H \cdot 3$.

- $s = 5$, which creates the coset

$$H \cdot 5 = \{5, 2\}.$$

  Notice that this coset is the same as $H \cdot 2$.

- $s = 6$, which creates the coset

$$H \cdot 6 = \{6, 1\}.$$

  Notice that this coset is the same as H.

Example. Consider the group S_3. This group has the subgroup

$$H = \{id, (123), (132)\}$$

as a subgroup. This subgroup has two possible distinct cosets:

- $H \cdot id = H \cdot (123) = H \cdot (132)$ are all the same coset, which is just H.
- $H \cdot (12) = H \cdot (13) = H \cdot (23) = \{(12), (13), (23)\}$.

With these definitions set down, we can actually start to do some graph theory:

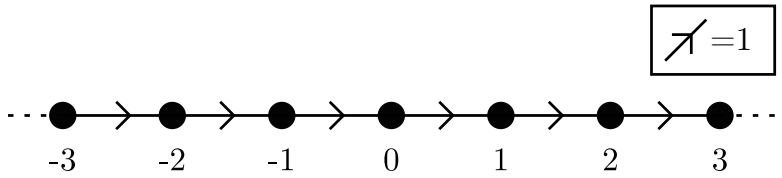
2.2 Cayley Graphs and Groups

Definition. Take any group A along with a generating set S. We define the **Cayley graph** $G_{A,S}$ associated to A as the following directed graph:

- Vertices: the vertices of G_A are precisely the elements of A.
- Edges: for two vertices x, y, create the oriented edge (x, y) if and only if there is some generator $s \in S$ such that $x \cdot s = y$. If this happens, we decorate the edge (x, y) with this generator s, so that we can keep track of how we have formed our connections.

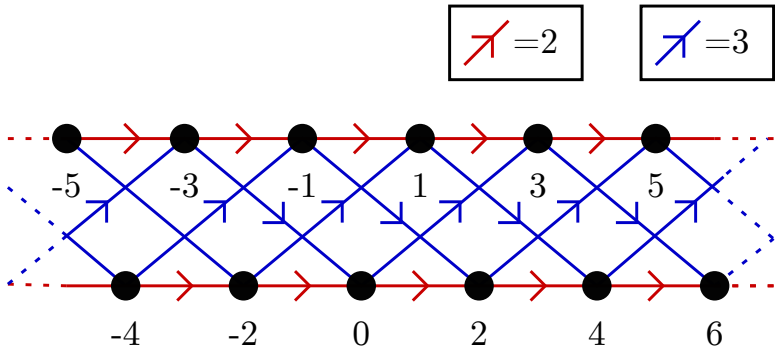
We consider a few examples here:

Example. The integers $\mathbb{Z}$ with the generator 1 have the following very simple Cayley graph:



This is not hard to see: we have one vertex for every element in our group (i.e. every integer,) and an edge (x, y) for each pair x, y such that $x = y + 1$, by definition. Because this is a Cayley graph, we label each of these edges with the generator that created that edge: for this graph, because there's only one generator this is pretty simple (we just label every edge with a 1.)

Example. The integers $\mathbb{Z}$ with the generating set $\{2, 3\}$ have the following Cayley graph:

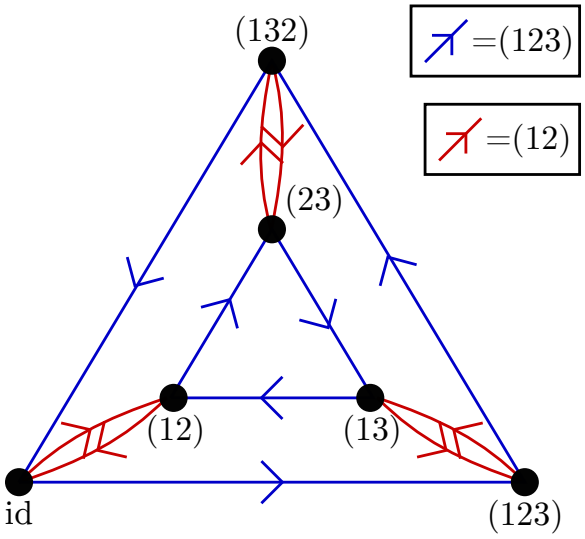


Again, our vertices are just the integers. However, this time we have two generators: the generator 2 connects any two integers that differ by 2, while the generator 3 connects any two integers that differ by 3. Notice that this graph is not the same as the graph above: in general, a group can have many markedly different Cayley graphs depending on the generators that you pick for it.

Example. Consider the symmetric group S_3 with generators $(12), (123)$. First, we calculate how these generators interact with our group elements when composed together:

group elt. $\circ$ generator	id	(12)	(13)	(23)	(123)	(132)
(12)	(12)	id	(123)	(132)	(13)	(23)
(123)	(123)	(23)	(12)	(13)	(132)	(123)

We can use this table to create the Cayley graph for this group and generating set:



Example. Consider the group given by the presentation

$$\langle a, b \mid a^3 = b^2 = (ab)^2 = id \rangle.$$

Because we do not know all of the elements in this group ahead of time, it is not necessarily obvious how to create this group's Cayley graph; unlike in our earlier examples, we cannot simply write down all of the vertices and then draw edges corresponding to our generators.

Instead, to find the Cayley graph corresponding to this group, we can use the following procedure:

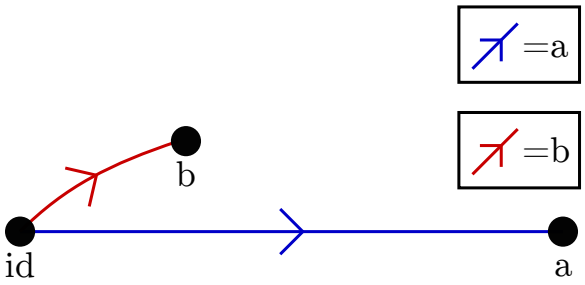
0. Start by placing one vertex that corresponds to the identity.

1. Take any vertex corresponding to a group element g that we currently have in our graph. Because our graph is a Cayley graph, it must have one edge leaving that vertex for each generator in our generating set. Add edges and vertices to our graph so that this property holds.

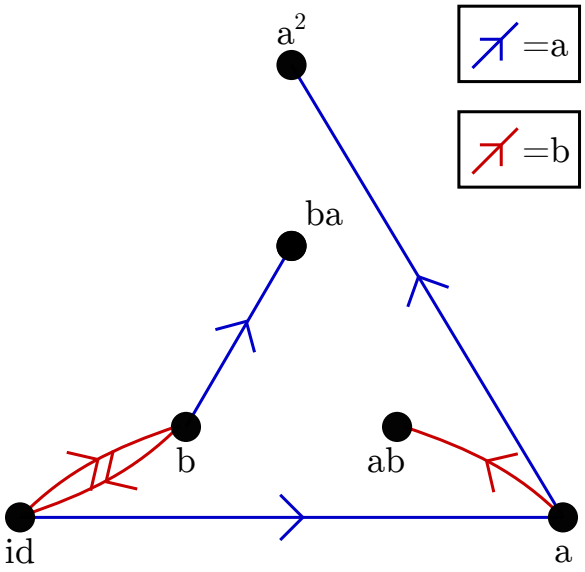
2. If some word R_i is a word that is equal to the identity in our group, then in our graph the path corresponding to that word must be a **cycle**: this is because if this word is the identity, then multiplying any element in our group by that word (i.e. taking the walk on our graph corresponding to that word) should not change that element (i.e. our walk should not take us somewhere new, and therefore should return to where it started!)

   Identify vertices only where absolutely necessary to insure that this property holds at every vertex. (This is the computationally "difficult" part of this algorithm. In general, finding the Cayley graph, or even more simply determining whether two arbitrary words in a presented graph are equal, is an **undecidable** problem: it is provable that no algorithm exists that will always solve this problem. Look up things like the halting problem if you want more examples of such things.)

So: if we do this here, we would start by drawing the following graph.



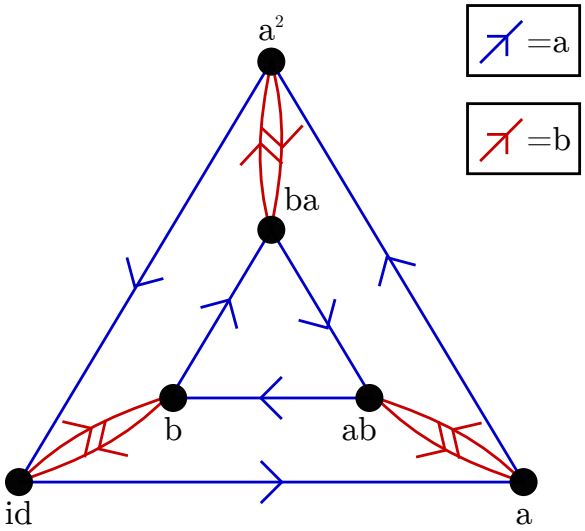
We add edge/vertex pairs to both of these added vertices a, b, that correspond to our generators. Notice that the relation $b^2 = id$ tells us that our b-edge leaving b must return to id, and that none of our other relations apply at this current stage (as they correspond to walks of length at least 3.)



Now, we draw new edge from the vertices ab, ba, a^2. Notice that the relation $a^3 = id$ tells us that the a-edge leaving a^2 returns to the identity, and that the relation $b^2 = id$ tells us that the b edge leaving ab returns to a. Furtheromre, the relation $abab = id$, along with the observations that $b^2 = id \Rightarrow b = b^{-1}, a^3 = id \Rightarrow a^2 = a^{-1}$ gives us a number of new relations:

- $abab = id \Rightarrow bab = a^{-1} = a^2$, and therefore the b-edge leaving ba goes to a^2. Furthermore, this also tells us that the b-edge leaving a^2 goes to ba, because the walk corresponding to b^2 starting from ba must return to ba.
- $abab = id \Rightarrow aba = b^{-1} = b$, and therefore that the a-edge leaving ab goes to b. Furthermore, this also tells us that the a-edge leaving ba goes to ab, because the walk corresponding to a^3 starting at ab must return to ab.

This gives us the following graph:



At this stage, we have satisfied our second property (that there is an edge leaving each vertex for each generator,) and we have only identified vertices when absolutely forced to do so by our relations. From visual inspection, it is clear that we satisfy the three relations $a^3 = b^2 = abab = id$ at every vertex; so this is the Cayley graph corresponding to our group!

3 Schreier Diagrams

This class's lecture continues last's class's discussion of the interplay between groups and graphs. In specific, we define the **Schreier diagram** in these notes, calculate some examples, and (if there is time) look at some applications of these techniques!

3.1 Schreier graphs

Definition. Take a group G, a subgroup H of G, and some collection of elements S that generate G. We create the **Schreier diagram** corresponding to this collection of information as follows:

- Vertices: the various right cosets of H in G.
- Edges: connect two cosets K, L with an edge if and only if there is some element $s \in S$ such that $Ks = L$.

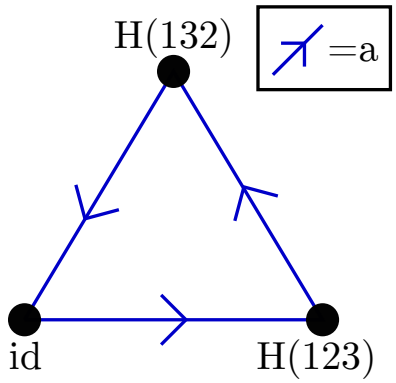
In this sense, a Cayley graph is simply a Schreier diagram where we set $H = \{id\}$.

We consider a pair of examples:

Example. Let's take $G = S_3$ as before, with the subgroup $H = \{id, (12)\}$ and generating set $a = (123)$. This group has three possible cosets for H to bounce between:

$$\begin{aligned} H = H \cdot (12) &= \{id, (12)\}, \\ H \cdot (13) = H \cdot (132) &= \{(13), (132)\}, \\ H \cdot (23) = H \cdot (123) &= \{(23), (123)\}. \end{aligned}$$

This gives us a fairly simple Schreier diagram, if we use the fact that $a^2 = (132)$:



Example. Consider the group $G = D_8 =$ the collection of all symmetries of a square. We denote its eight elements, defined in last week's lecture notes, as the set

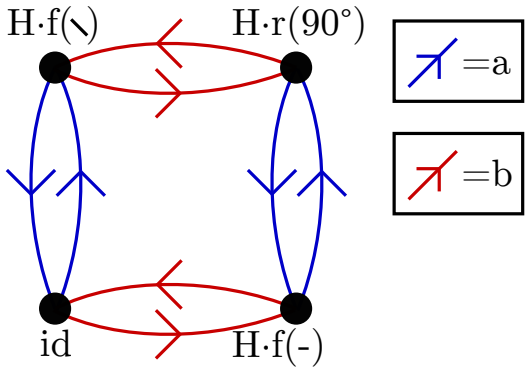
$$\{id, \text{rot}(90°), \text{rot}(180°), \text{rot}(270°), \text{flip}(\,|\,), \text{flip}(-), \text{flip}(\diagdown), \text{flip}(\diagup)\}.$$

By the "flip(line)" expressions, we mean the four symmetries of the square that consist of flipping the square over some axis, with the appropriate axis given in parentheses next to each flip.

Take the subgroup $H = \{id, \text{rot}(180°)\}$ along with the generators $S = \{a = \text{flip}(\diagdown), b = \text{flip}(-)\}$. Our subgroup has four possible cosets:

$$\begin{aligned} H = H \cdot \text{rot}(180°) &= \{id, \text{rot}(180°)\}, \\ H \cdot \text{rot}(90°) = H \cdot \text{rot}(270°) &= \{\text{rot}(90°), \text{rot}(270°)\}, \\ H \cdot \text{flip}(\,|\,) = H \cdot \text{flip}(-) &= \{\text{flip}(\,|\,), \text{flip}(-)\}, \\ H \cdot \text{flip}(\diagdown) = H \cdot \text{flip}(\diagup) &= \{\text{flip}(\diagdown), \text{flip}(\diagup)\}. \end{aligned}$$

This gives us another fairly simple Schreier diagram:



The ease of the above two calculations indicates part of the reason why we might like Schreier diagrams: they are often easier to calculate than Cayley graphs. In exchange, however, we're only getting information about the cosets of H instead of the elements of our group — but if we only care about the elements of our group "up to" the elements H of our coset, this is still pretty great!

To illustrate a situation where working with the Schreier diagram is markedly easier than the Cayley graph, consider the following problem:

Problem. Consider the presented group

$$\langle a, b \mid a^2 = b^5 = (ba)^3 = id \rangle,$$

which has $< b \mid b^5 = id >= \{id, b, b^2, b^3, b^4\}$ as a subgroup. What is the Schreier diagram of this group with the generators $\{a, b\}$?

Answer. We use the same heuristics to find this Schreier graph that we used to find the Cayley graph for a presented group. We copy these heuristics from our earlier set of notes here:

0. Start by placing one vertex that corresponds to the "identity" coset H.

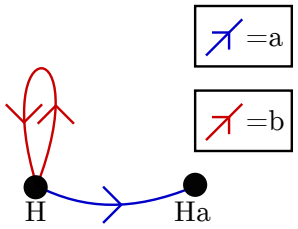
1. Take any vertex corresponding to a coset K that currently has a corresponding vertex in our graph. Because our graph is a Schreier graph, it must have one edge leaving that vertex for each generator in our generating set. Add edges and vertices to our graph so that this property holds.

2. If some word R_i is a word that is equal to the identity in our group, then in our graph the path corresponding to that word must be a **cycle**: this is because if this word is the identity, then multiplying any element in our group by that word should not change that element.

   Identify vertices only when forced by our relations to insure that this property holds at every vertex.

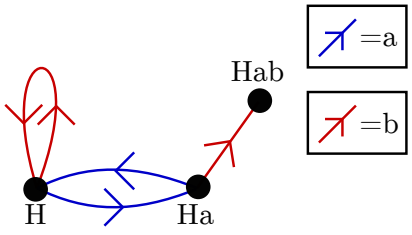
3. Also, when we draw edges out of the "identity" coset H, we may be forced to have some of those edges return to H: i.e. in the group above, $Hb = H$. So for this first vertex, we might be forced to have some self-loops that don't correspond to our words. This can sometimes happen later as well, depending on what H is (see problem 2 in HW 12 for an example!); watching out for this is one of the "fun" parts of drawing a Schreier diagram!

We run this process here. We start with one vertex corresponding to the coset H:



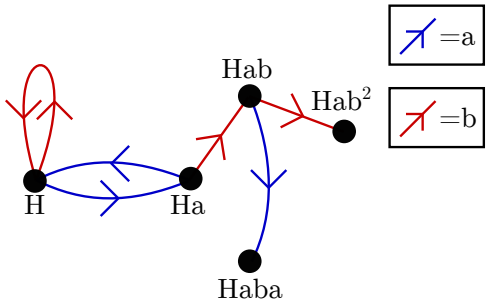
Note that because $Hb = H$, the b-edge leaving H returns to H itself, forming a loop. (This illustrates some of the slightly trickier aspects of working with cosets instead of groups. This, however, is the only time this will come up, which perhaps illustrates that cosets aren't so bad after all.)

We now take our one new vertex Ha and draw the two a, b-edges leaving Ha:

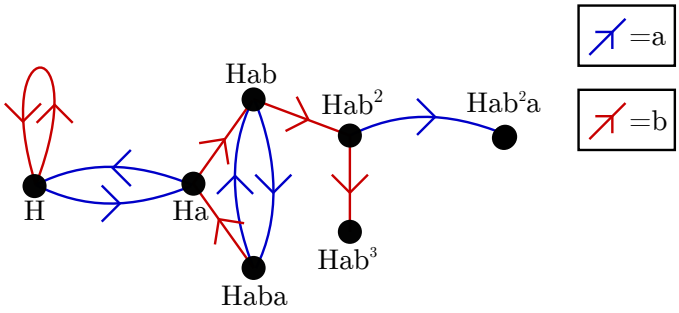


Here, we use the relation $a^2 = id$ to conclude that $Ha^2 = H$.

We now draw the edges leaving Hab:



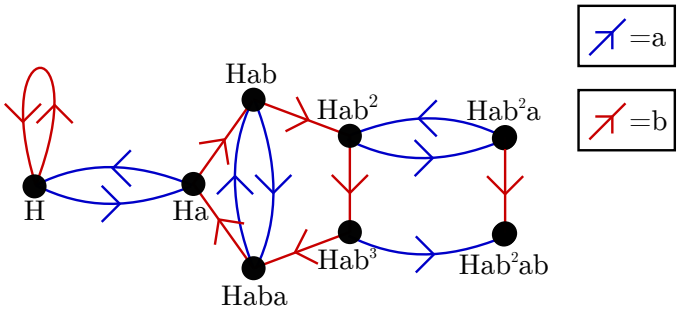
And repeat this process on $Hab^2, Haba$:



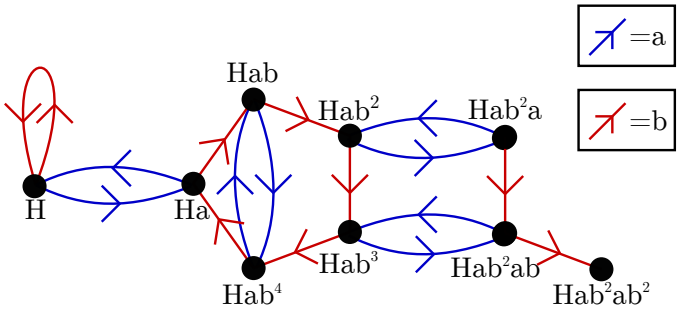
Notice here that the relation $a^2 = id$ means that the a-edge leaving $Haba$ returns to Hab; in general, this property will always insure that these a-edges come in pairs, and we will use this identification throughout the rest of this proof without calling it out.

More interestingly, note that $Habab = Ha$. This is because $bababa = id$ is equivalent to asking that the walk corresponding to $bababa$ starting at the origin returns to the origin. After the first four steps, we are at $Haba$; to return to H along an a-edge, we must go to Ha, which forces our connection.

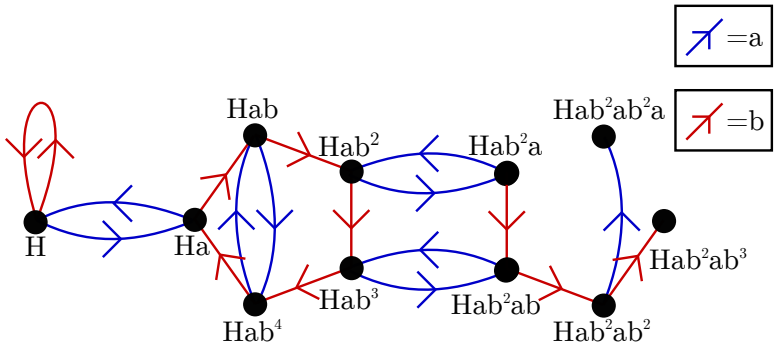
We draw more edges:



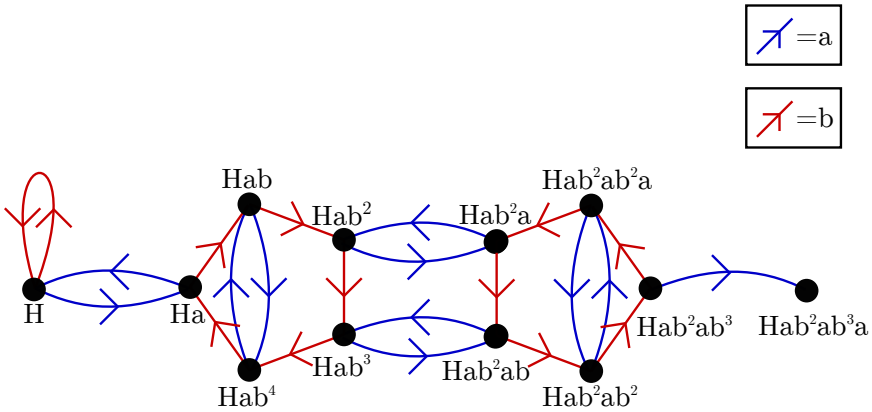
Notice that $Haba = Hab^4$; this is because if we start at $Haba$ and take the walk of length 5 given by the b-edges, we should return to ourselves. Also notice that $Hab^3a = Hab^2ab$; this is because the walk $bababa$ starting at Hab^3 must return to itself, and therefore that the a-edge leaving Hab^3 must go to whatever b-edge leaves Hab^2a.



Nothing nontrivial was identified above, so we continue our process:

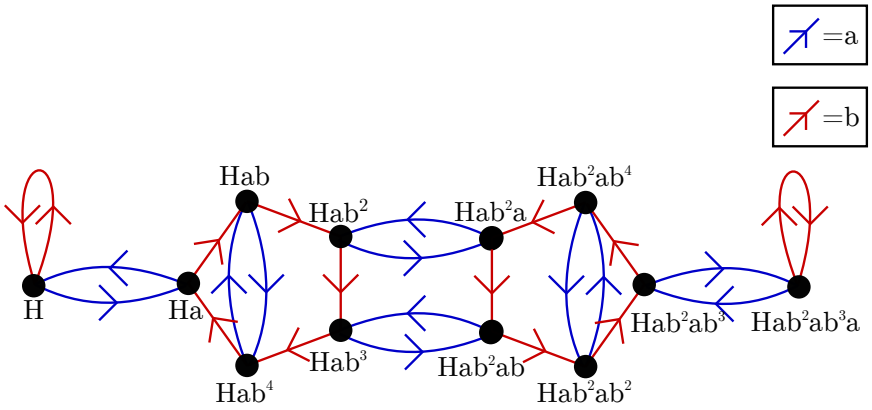


Still nothing. More edges!



Ok, now some interesting things have happened. Notice that we've identified Hab^2ab^2a with Hab^2ab^4; this is again because of the walk $bababa = id$, starting this time from the vertex Hab^2. In particular, because walking $baba$ from Hab^2 takes us to Hab^2ab^2a and walking ba more must return us to Hab^2, we know that our b-edge leaving Hab^2ab^2a must go to Hab^2a. Similarly, taking the walk b^5 starting from this Hab^2ab^2a vertex must return us to ourselves, forcing the b-edge leaving Hab^2ab^3 to go to Hab^2ab^2a.

We draw our last batch of edges:



Note that the b-edge leaving Hab^2ab^3a must return to itself, as the walk $bababa = id$ starting from the vertex Hab^2ab^3 forces the b-edge leaving Hab^2ab^3a to return to itself.

This gives us a ton of useful information about our group: it tells us that there are 60 elements (as we have 12 cosets, each containing 5 elements), and moreover it tells us how these cosets get moved around by a and b (in particular, looking at our graph tells us that b keeps two cosets constant and moves the other 10 around in two groups of 5.) For those of you who have done some group theory before, this actually is enough to tell us what this group is in its entirety (it's A_5, the alternating group on 5 elements!)

It turns out that adding a bit more information to our diagram can make them even more useful:

3.2 Decorated Schreier Diagrams

Definition. Given a Schreier diagram for a group G with subgroup H and generators S that we've labeled our edges with, we can **decorate** it! We do this as follows:

- Take all of the vertices of our Schreier diagram. Each vertex corresponds to a coset K. Pick some element $k \in K$, and use that element to **decorate** the vertex corresponding to that coset.

  Notice that if we have decorated a coset K with some element $k \in K$, then we can actually write $K = Hk$. So this decoration is a pretty natural one to use.

- Now, suppose that there is an a-edge going from one coset $K = Hk$ to another coset $L = Hl$. We decorate this edge with the group element α such that $ka = \alpha l$.

  Notice that because $L = Ka = Hka$, we can write $l = hka$ for some $h \in H$, and thus have $ka = \alpha hka \Rightarrow \alpha = h^{-1}$. In particular, this means that all of the edge decorations (1) exist, as we found a formula to find them, and (2) are all elements from our coset H.

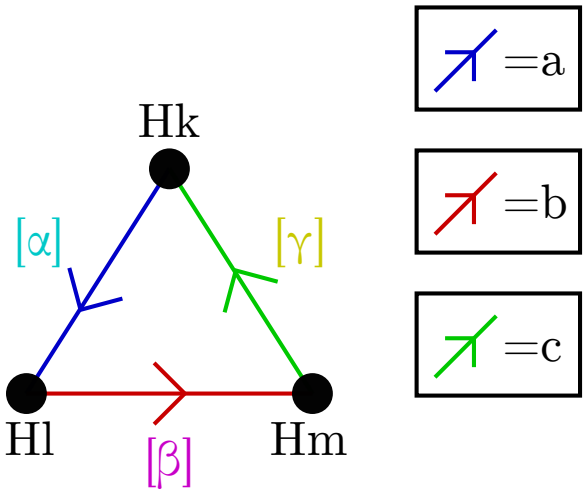
Decorated Schreier diagrams satisfy a fairly interesting property:

Proposition. Take any Schreier diagram for a group G with subgroup H. Decorate it. Take any closed walk in our Schreier diagram that starts and ends at the H-vertex[1]. The

[1]In a directed walk, this is potentially ambiguous. For this talk, we mean any subgraph that when we forget the orientations of our edges, we get something that would be a closed walk in an unoriented graph.

product of the group elements used to label the edges of this closed walk, in the order given by our closed walk, is the same thing as the product of the group elements used to decorate our edges (in the order given by our closed walk.)

Proof. To illustrate the idea, let's take an arbitrary decorated three-vertex cycle starting from some coset Hk, where the edges are oriented as drawn below:



A decorated three-cycle from within some Schreier graph. The vertices Hk, Hl, Hm are all decorated via their representatives k, l, m. There is an edge $Hk \to Hl$ given by the generator a, $Hl \to Hm$ given by the generator b, and $Hm \to Hk$ given by c; as well, these three edges are decorated by the labels α, β, γ.

Notice that because the a-edge $Hk \to Hm$ is decorated with an α, we have $ka = \alpha l$; similarly, because the b-edge $Hl \to Hm$ is decorated with β, we have $lb = \beta m$, and because the c-edge $Hm \to Hk$ is decorated with γ, we have $mc = \gamma k$.

Consequently, if we look at the product $kabc$, we have

$$kabc = \alpha lbc = \alpha\beta mc = \alpha\beta\gamma k.$$

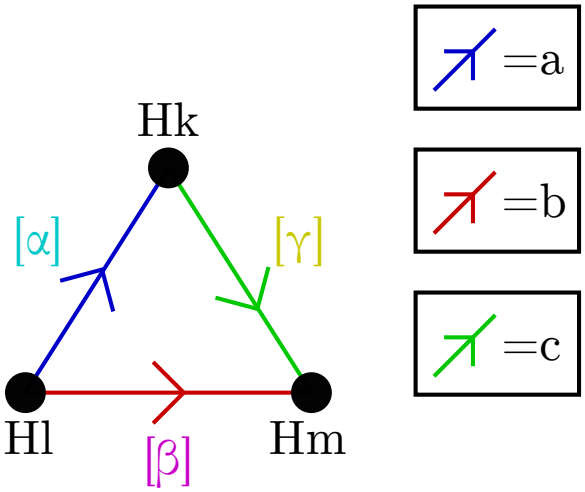
In particular, if $k = id$ — in other words, if $Hk = H$ — we have $abc = \alpha\beta\gamma$. In other words, the product of the "labels" on our cycle is the same thing as the product of the "decorations" on our cycle!

This proof generalizes to oriented cycles of length n by almost exactly the same proof: simply take any cycle with vertices decorated $k_1, \ldots k_n$, edges $k_i \to k_{i+1}$ labeled a_i and decorated α_i. Then by the exact same argument as above, we have

$$k_1 a_1 a_2 \ldots a_n = \alpha_1 k_2 a_2 \ldots a_n = \ldots \alpha_1 \alpha_2 \ldots \alpha_n k_1,$$

which gives us $a_1 a_2 \ldots a_n = \alpha_1 \alpha_2 \ldots \alpha_n$ in the case that the vertex corresponding to k_1 is the subgroup H (i.e. $k_1 = id$.)

We finally note that because the condition $ka = \alpha l$ is equivalent to the request $\alpha k = la^{-1}$, we can deal with the situation where edges are oriented in the "wrong" directions by simply replacing the a, α's with their inverses. For example, suppose we returned to our triangle from before, but messed with some of the orientations:



In this situation, we would use the relations

- $la = \alpha k \Rightarrow ka^{-1} = \alpha^{-1} l$,
- $lb = \beta m$,
- $kc = \gamma m \Rightarrow mc^{-1} = \gamma^{-1} k$

to perform the transformation

$$ka^{-1}bc^{-1} = \alpha^{-1} lbc^{-1} = \alpha^{-1}\beta mc^{-1} = \alpha^{-1}\beta\gamma^{-1}k.$$

Again, setting $k = 1$ gives us that the product of the labels of edges in our closed walk is the same as the product of the decorations of edges in our closed walk, provided that we interpret the "orientation" of each edge as telling us whether a group element is represented by itself or its inverse. □

One convenient way to decorate a Schreier diagram is via the following process:

Proposition. Take any Schreier diagram for a group G with subgroup H. The following process induces a unique decoration of this diagram:

- Decorate the H-vertex with the element $id \in H$.
- Pick out some spanning tree[2] T in our graph. Decorate all of the edges in this spanning tree with the element $id \in H$.

Proof. This is not too hard to see. Look at any vertex K that is distance 1 from H, where we measure distance from the origin via our spanning tree: i.e. we are declaring that a vertex is distance n from H if there is a path of length n from H to that vertex in our spanning tree T. Because T is a spanning tree, this gives a well-defined distance function.

[2]Recall that a **spanning tree** of a graph G is a subgraph of G that (1) is a tree, and (2) contains every vertex in our graph. In this setting, where we are dealing with directed graphs, this notion might again be ambiguous; for this talk, we further define a tree as any subgraph that when we forget the orientations of our edges, we get something that would be a tree in an unoriented graph.

Suppose that the edge in our spanning tree connecting K to the origin is labeled a, and goes from $H \to K$. If we want H to be decorated as id and this a-edge to be labeled id, we are asking that the decoration of K is some element $k \in K$ such that $id \cdot a = k \cdot id$: i.e. that each of these vertices K has a unique decoration, given (in this particular case) by the edge-labeling that led to that coset.

The other case, where the edge goes from K to H, is similar; if we want H decorated as id and the a-edge $K \to H$ to be decorated id, then we must have K decorated with a k such that $ka = id \cdot 1 = id$, which again uniquely determines k. (This is like the orientations-corresponding-to-inverses relationship we saw in our earlier result.)

Now, suppose that we have decorated all of the vertices out to distance n, and want to decorate vertices at distance $n+1$. Take any K at distance $n+1$: because T is a spanning tree, there is some unique edge connecting a previously-decorated vertex L at distance n to our vertex K via an edge in T. Assume this edge is labeled with some element a, decorated by id, and that L is decorated with some element l.

Then, if the edge goes from $L \to K$, K must be decorated with an element k such that $la = id \cdot k$; similarly, if the edge goes from $K \to L$, K must be decorated with some k such that $ka = id \cdot l$. Notice that this uniquely defines K's labeling. Furthermore, notice that this labeling is conflict-free: because T is a tree, there is no way for us to have two conflicting claims as to what K's decoration should be.

This decorates all of the vertices in our graph. Now, take any edge $K \to L$ in our graph that we have not yet labeled (i.e. any edge not in the spanning tree.) Consider the closed walk formed by starting at H, walking to K along the unique path to K in our spanning tree, taking the edge $K \to L$, and walking back to H via the unique path back to H in our spanning tree. This is a closed walk; therefore, the product of the decorations of edges on this walk must be equal to the product of the labelings of edges on this walk!

But every edge in our walk is decorated by 1's, except for the $K \to L$ edge which we're trying to decorate. Therefore, this gives us a unique decoration of this edge, given by the labelings of the walks $H \to K$ and $L \to H$. So we've decorated our graph! □

This method of decoration has an interesting consequence:

Theorem. Take any Schreier diagram for a group G with subgroup H, along with a generating set S for G. Decorate this diagram. Then the subgroup H is generated by the decorations of the edges in our graph.

Proof. Take any element $h \in H$. Because S generates G, we can write h as some product $s_1 \dots s_n$ of elements (possibly repeated and with inverses) from S. This corresponds to a walk on our Schreier graph: furthermore, because $s_1 \cdot \ldots \cdot s_n = h \in H$, this walk must start and end at H.

Decorate our Schreier diagram (say, using the decoration given above.) Now, the product of labels on this walk must be equal to the product of the decorations of the edges on this walk: in other words, we can write h as the product of some of the decorations of the edges in our graph! So any h can be written as the product of decorations in our graph.

Furthermore, by using walks that start at H and walk along edges in the spanning tree to get to any edge in our graph, walking on that edge, and then returning along our spanning tree edges, we can see that the decoration of any edge in our graph is an element in our subgraph. Therefore H is generated by these decorations, as claimed! □

This theorem has the following very beautiful extension:

Corollary. Take any Schreier diagram for a group G with subgroup H, along with a generating set S for G. Decorate this diagram. Suppose that G has a presentation $\langle a_1, a_2, \ldots \mid R_1, R_2, \ldots \rangle$. Then the subgroup H has a remarkably nice presentation:

$$H = \langle d_1, d_2, \ldots \mid D_{1,1}D_{1,2}\ldots, D_{2,1}, D_{2,2}, \ldots \rangle,$$

where

- The generators $d_1, d_2, \ldots$ are all of the decorations of edges in our graph.
- The relations $D_1, D_2, \ldots$ are given by the following process: take any relation R_i from G. R_i corresponds to a labeled walk in G's Cayley graph, that starting from any vertex must return to that vertex. In other words, in our group, the product of the labelings on this walk's edges must be the identity.

  Now, we know that the product of the labels on this walk must be equal to the product of the decorations on this walk. In other words, a relation R_i on our generators can create several relations on the generators $d_1, d_2, \ldots$! Call these new relations $D_{i,1}, \ldots D_{i,n}$.

The main point of the above discussion is that all of the relations on H must come, in some sense, from pre-existing relations in G. (Think for a bit if this isn't clear; there is some nonobvious mathematics going on in this statement!)

If we consider the case where G is a free group (i.e. a group with no relations) we get the following result "for free:"

Corollary. Any subgroup of a free group is free.

For those of you who haven't done group theory before, this might not be very surprising, and seem like it should be a relatively "trivial" result. This is far from the truth; what we've presented here is the closest to a purely algebraic proof that is known, and is one of the simplest proofs I am aware of[3]!

[3]The other one I know goes through algebraic topology, and is similar in difficulty.