Typing with Weird Keyboards Notes

Yakov Berchenko-Kogan

August 25, 2012

Abstract

Consider a language with an alphabet consisting of just four letters, *a*, *b*, *ɐ*, and *q*. There is a spelling rule that says that whenever you see an *a* next to an *ɐ*, you cross those two letters out. Similarly, if you see a *b* next to a *q*, you cross those two letters out as well. For example, if you see *aqaɐbb*, you'd cross out *aɐ* to get *aqbb*, and then youd cross out *qb* to get *ab*.

For some inexplicable reason, members of the Mafia use weird keyboards when typing in this language. Instead of having keys labeled *a*, *ɐ*, *b*, and *q*, they have keys labeled *aaq*, $b^{\text{ɐɐ}}$, *bbɐ*, and ^{a}*qq*. A member of the Mafia can, for example, type the word *aabɐ* by pressing the *aaq* key followed by the *bbɐ* key to get *aaqbbɐ*, and then crossing out the *qb*.

One day, Ruthi gets accused of being in the Mafia. However, she recently wrote a newspaper article containing the word *aab*. Ruthi claims that nobody in the Mafia can type *aab*, and thus she must be innocent. How can we verify Ruthi's claim?

We'll see how we can easily figure out what words a given keyboard can produce using directed graphs (dots connected by arrows). We'll see how to use these graphs to answer other questions. For example, we'll see how to tell when one keyboard can type every word another keyboard can, and we'll see how, given two keyboards, we can make a keyboard that can type only the words that both of the given keyboards can type.

We'll put these concepts into their larger group theoretic context, so it will be helpful (but not required) to be familiar with the basic group theory definitions.

1 Free Groups: Typing with normal keyboards

We start with an alphabet, for example $\{a, b, c\}$. We write words using the letters in the alphabet either right side up or upside-down.

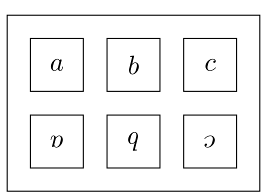
Example 1.1. In this alphabet, we can write the word *aɔaɐcbɐ*.

There is a spelling rule that whenever we see a letter right side up next to the same letter upside-down, we can erase both of them.

Example 1.2. When we apply the spelling rule, *aɔaɐcbɐ* becomes *aɔcbɐ*, which then becomes *abɐ*.

Hannah wants to write a song in this language, and she has a keyboard that looks like this:

Figure 1: Hannah's keyboard.

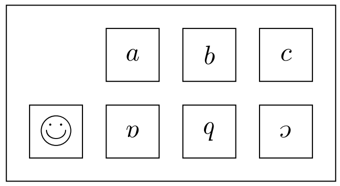


Exercise 1.3. Hannah accidentally typed *aqab*. She wants to erase it, but there's no backspace button on her keyboard! What should she type next to erase it?

Exercise 1.4. Hannah accidentally typed *abcabc* when she really wanted to type *abɔabc*. What should she type next to fix her mistake?

Alfonso's keyboard looks just like Hannah's, but it has an extra smiley face key:

Figure 2: Alfonso's keyboard.



When Alfonso holds down the smiley face key, pushing another key causes that letter to appear at the end of the word *and* at the beginning of the word flipped over.

Example 1.5. Alfonso can type *abɐ* by pressing *b* and then holding down the smiley key and pressing *ɐ*.

Exercise 1.6. If Alfonso types *ababbaa* and then types the first few letters of the word while holding the smiley key, what happens? Remember to apply the spelling rule.

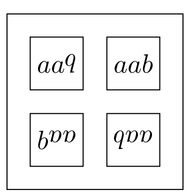
Definition 1.7. The set of all finite-length words you can get after applying the spelling rule is called the *free group* over the alphabet X. We will let F denote the free group, and sometimes by $F(X)$ if we want to make it clear what the alphabet is. In our examples above, $X = \{a, b, c\}$. The "spelling rule" is usually called *free cancellation*, and most people write $\bar{a}$ or a^{-1} instead of *ɐ*, because writing upside-down letters is hard.

If you know group theory, you might be wondering what the group operation is. The group operation here is *concatenation*, that is, writing one word after the other (and then applying free cancellation if necessary). If you did Exercise 1.3, you should be able to figure out what the inverses are.

2 Finitely Generated Subgroups: Typing with silly keyboards

Susan doesn't like typing with normal keyboards like Hannah's. Instead, she has keyboards that look like something like this:

Figure 3: Susan's keyboard.



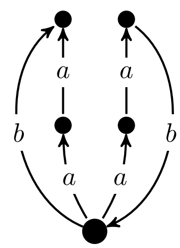
The first row of the keyboard is a collection of words, and the second row is the words in the first row upside-down.

What words can Susan type? There are infinitely many, but she probably can't type every word, so we'd like an easy way to test whether or not a given word, such as *aabbb*, can be produced with this keyboard. One way to do this would be to systematically make a list of the words that can be typed by pressing one key, then two keys, then three keys, and so forth. If we find *aabbb* on this list, then we're done. Eventually, the words on the list will become very long, much longer than *aabbb*. If we don't see *aabbb* by that point, we can be reasonably sure that it can't be typed on this keyboard. The problem is that, because of free cancellation, even though we might press a lot of keys, the resulting word might still be very short.

Exercise 2.1. In fact, it is possible to type *aabbb* with this keyboard. Find a way to do it.

There's a much better way of testing whether or not a word can be typed on the keyboard. We first make a labeled directed graph out of loops labeled with the words in the first row, like this:

Figure 4: Making the graph corresponding to Susan's keyboard, step 1: loops.



The left loop corresponds to *aaq*, and the right loop corresponds to *aab*. The vertex where all of the loops start and end is drawn larger. Notice that we represent an upside-down letter by reversing the direction of the edge.

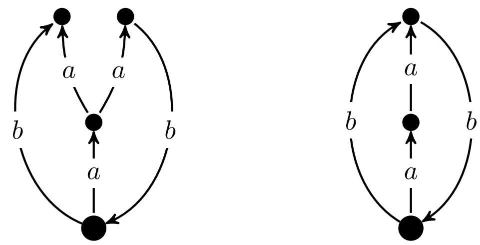
Pressing a key on the keyboard corresponds to going around one of these loops, starting and ending at the big vertex. (Pressing the keys in the bottom row corresponds to going around the loop in the other direction.) Thus, a word can be typed on this keyboard if and only if you can read it off when going around the graph starting and ending at the big vertex. However, if we try to use this graph to figure out whether or not a given word can be typed on this keyboard, we will run into some problems.

Example 2.2. There are two edges labeled *a* going from the big vertex. If the word we are testing starts with *a*, we don't know which edge to take. We'd have to try both possibilities. If the word is very long, we might end up with too many possibilities to go through.

Example 2.3. You can type *bb* using this keyboard because it's what you get after applying free cancellation to *bɐɐaab*, which you can read off by going around the graph. However, if we were asked to test whether or not *bb* could be typed on this keyboard, we wouldn't have any way of knowing to insert *ɐɐaa* in the middle, except by guessing.

We can fix these problems by *folding* the graph, like this:

Figure 5: Making the graph corresponding to Susan's keyboard, step 2: folding.



Whenever you see two edges with the same label either both going from the same vertex or both going to the same vertex, you fold them together. Once you can't fold the graph any further, the graph is called *folded*. Now, we can't run into the problems we had before, because given a vertex and a letter there's at most one way to go, and there are no paths where a letter is followed by its inverse.

Sometimes, there will be more than one way to fold a graph, but, at the end, the resulting folded graph will always be the same.

Exercise 2.4. For each of the following words, check if Susan's keyboard can type them using the corresponding folded graph.

1. *aabbb*
2. *bbbaa*
3. *bbbɒɒ*
4. *bɒɒb*
5. *b*

Exercise 2.5. Come up with a few other words that this keyboard can and can't type.

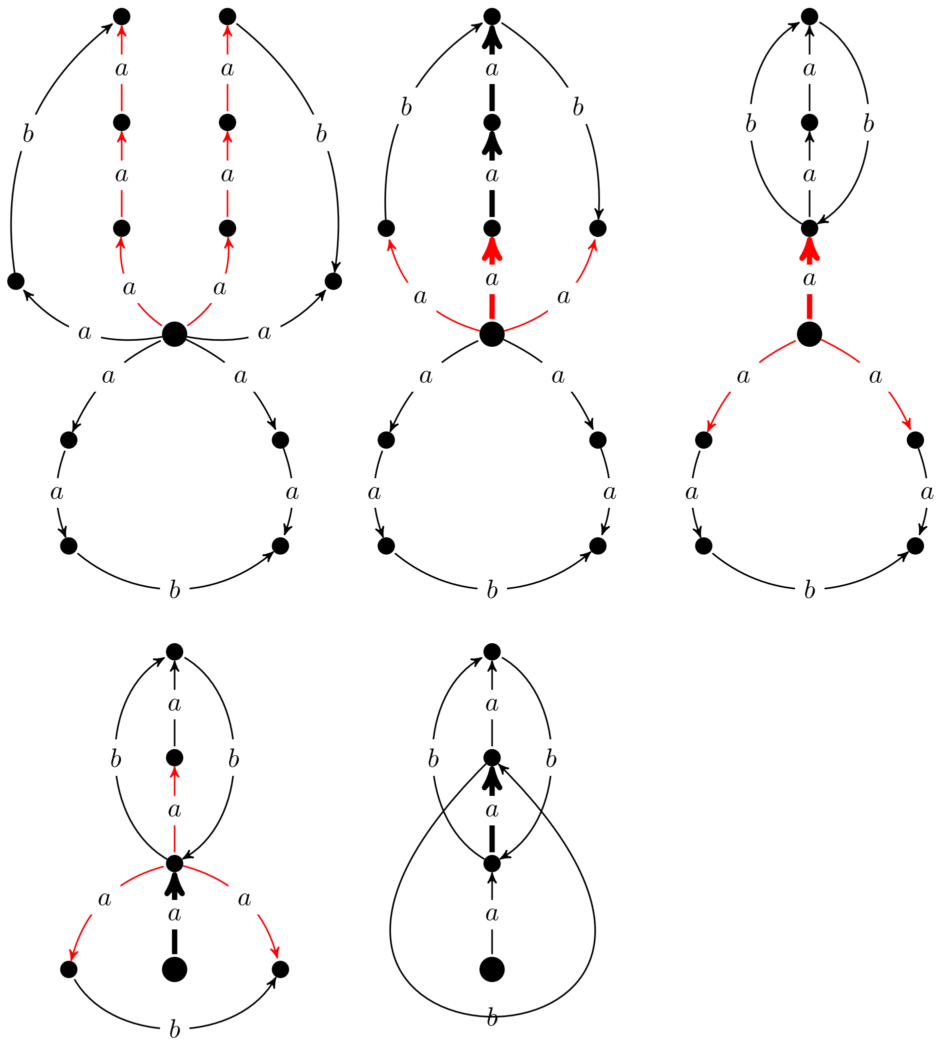
Example 2.6. Here's Kevin's keyboard:

Figure 6: Kevin's keyboard.

aaaqɒ	*aaabɒ*	*aabɒɒ*
aƅɒɒɒ	*aqɒɒɒ*	*aaqɒɒ*

We construct the corresponding folded graph below. Note that we do a couple folding steps at a time. The edges marked in red are the ones about to be folded, and the resulting edges are thick.

Figure 7: Folding the graph corresponding to Kevin's keyboard.



Definition 2.7. Let H be the set of words that can be typed on a keyboard with a set S of words in the first row and the corresponding upside-down words in the second row. We call H the *subgroup of* $F(X)$ *generated by* S, and we write $H = \langle S \rangle$. We call S a set of *generators* of H. (Note that a subgroup has many possible generating sets.) If S is finite, then H is *finitely generated*. (In this class we'll almost entirely be dealing with finitely generating subgroups.)

Example 2.8. Kevin's keyboard corresponds to the subgroup $\langle aaa\bar{b}\bar{a}, aaab\bar{a}, aab\bar{a}\bar{a} \rangle$.

Definition 2.9. Given a set of letters X, an X*-digraph* is a directed graph whose edges are labeled with elements of X. Edges from a vertex back to itself are allowed, as are multiple edges between two vertices. The endpoints of an edge are its *origin* and *terminus*. Many of our X-digraphs will have a special *base vertex* which is drawn larger. (Note that digraph stands for *directed graph*.)

Example 2.10. All of the graphs in Figures 4, 5, and 7 are X-digraphs with $X = \{a, b\}$.

Definition 2.11. Given an X-digraph Γ with base vertex v, the *language* of the X-digraph with respect to the base vertex v, denoted $L(\Gamma, v)$, is the set of labels of the reduced paths from v to v.

Another way to describe our algorithm above is to say that it takes a finitely generated subgroup H and outputs a folded graph whose language is precisely H.

Definition 2.12. Given a subgroup H of a free group $F(X)$, the corresponding folded graph is called the *Stallings subgroup X-digraph of H* and is denoted $\Gamma(H)$.

Stallings subgroup graphs and the folding process now known as Stallings folding were developed by geometric group theorist and topologist John Stallings (1935-2008) in a 1983 paper. Stallings graphs had applications far beyond making a much faster algorithm for testing whether or not a word is in a subgroup, and they dramatically changed the way in which subgroups of free groups are studied.

Example 2.13. If $H = \langle aaa\bar{b}\bar{a}, aaab\bar{a}, aab\bar{a}\bar{a} \rangle$, then $\Gamma(H)$ is the last graph above in Figure 7.

Exercise 2.14. Construct the Stallings subgroup digraph for the following subgroups.

1. $\langle a\bar{b}, ab \rangle$
2. $\langle \bar{a}bbaa, \bar{a}bba \rangle$
3. $\langle ababb, abababb \rangle$
4. $\langle aa, bb, ab\bar{a}\bar{b} \rangle$
5. $\langle abca, \bar{c}\bar{b}a, cc \rangle$

Exercise 2.15. Go back to the question in the blurb. Is it possible for Ruthi to be in the Mafia?

Exercise 2.16. Pick your favorite words over your favorite alphabet, and construct the corresponding Stallings digraph.

Exercise 2.17. If the Stallings digraph of a subgroup has a vertex that has just one edge (like in the Stallings graph of Kevin's keyboard), what can you say about the generators of the subgroup?

The starred exercises below are a preview of upcoming topics, but are not necessarily harder.

Exercise 2.18 (*). There are lots of X-digraphs, but only some of them are the Stallings X-digraph of some subgroup. For instance, Stallings X-digraphs are always folded. First find a folded X-digraph that is not a Stallings X-digraph of any subgroup, and then try to come up with a property that determines whether or not an X-digraph is a Stallings X-digraph of some subgroup.

Exercise 2.19 (*). If someone gives you a Stallings X-digraph, how can you find a keyboard that it corresponds to? In other words, given a folded digraph, how can you find the generators of the corresponding subgroup? (Note that there are lots of possible keyboards/generating sets for a given subgroup, and this question asks you how to find just one of them).

Exercise 2.20 (*). Just by looking at the corresponding graphs, given two keyboards how can you tell if one keyboard can type everything the other can?

Exercise 2.21 (*). Given two keyboards, how can you find a graph corresponding to the set of words that can be typed by both of them? In other words, given two subgroups, how can you find the graph corresponding to their intersection? (Hint: Use the graphs corresponding to the two subgroups.)

Exercise 2.22 (*). Roxana has a keyboard. One day, Tom takes each key and adds a to the end of the word on the key and v to the beginning. What sorts of things can happen to the corresponding graph?

Exercise 2.23 (*). Let $X = \{a, b\}$, and let H be the set of words in $F(X)$ where a and $\bar{a}$ appear the same number of times, and b and $\bar{b}$ appear the same number of times. Can you come up with a corresponding X-digraph for this subgroup? That is, find a folded X-digraph Γ such that the label of every path from the base vertex to the base vertex is in H, and such that every word in H is the label of some path in Γ from the base vertex to the base vertex. (Hint: The subgroup H is not finitely generated, so Γ is not going to be a finite graph.)

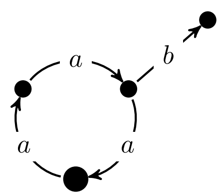
3 Finding Generators: Manufacturing silly keyboards

We first answer Exercise 2.18: What sorts of X-digraphs are Stallings X-digraphs of some subgroup?

Definition 3.1. A *reduced* path in a digraph is a path that never backtracks. That is, you never go along an edge and the immediately go along that edge in the other direction. (You can, however, go back along that edge later in the path.)

Notice that every vertex of a Stallings X-digraph is on some reduced path from the base vertex back to the base vertex. That is, you never have a situation like this:

Figure 8: A graph that cannot be a Stallings digraph because it is not core.



Definition 3.2. A digraph where every vertex is on some reduced path from the base vertex back to itself is called *core*. (Note that whether or not a graph is core depends on which vertex is the base vertex, so sometimes we say *core with respect to* v, where v is the base vertex.)

The properties folded and core are enough to characterize Stallings X-digraphs: If an X-digraph is both folded and core, then it is the Stallings X-digraph of the subgroup containing the labels of all of the reduced paths from the base vertex back to itself. We can say this fact as a theorem, whose proof we will leave out.

Theorem 3.3. *The Stallings graph of a subgroup H of a free group $F(X)$ is the unique folded core X-digraph with language H.*

Now we address Exercise 2.19. Given an X-digraph that is folded and core, how can we find a keyboard that corresponds to that graph? The idea is to find paths from the base vertex to the base vertex that go once around every "hole" in the graph, and then these paths can generate every other path. However, this vague rule will become hard to keep track of in more complicated graphs, so we need something more precise.

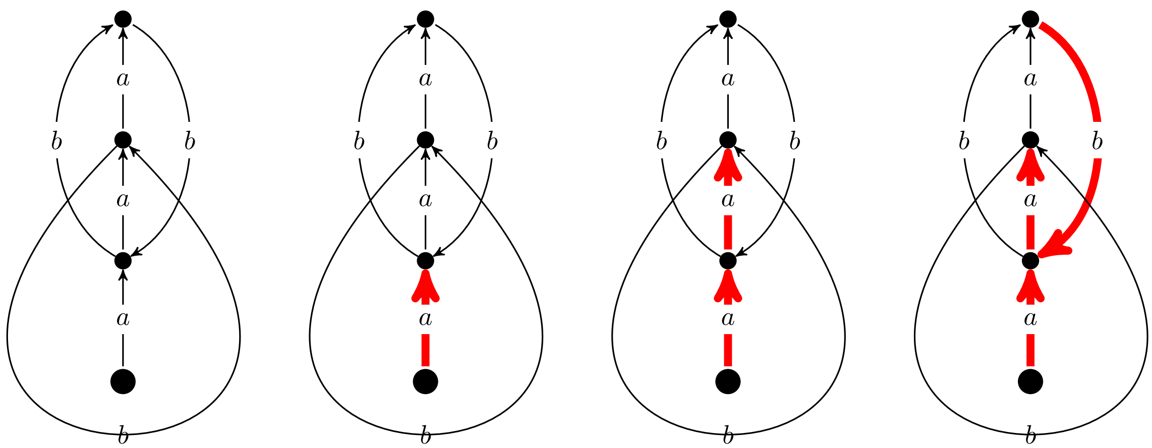
Definition 3.4. A *tree* is a connected graph with no cycles. That is, there are no reduced paths from a vertex back to itself.

Exercise 3.5. Show that, in a tree, there is a unique path between any two vertices.

Given an X-digraph, we can construct a subgraph that contains all of the vertices and is a tree. This subgraph is called a *maximal subtree* of the graph.

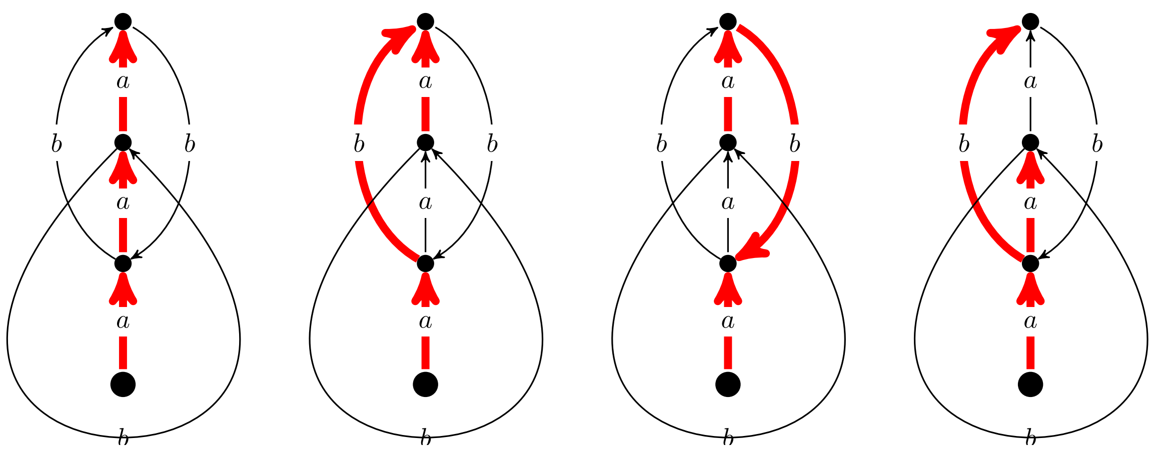
Example 3.6. We will construct a subtree of the Stallings graph of Kevin's keyboard. We start with the base vertex, and then attach one vertex at a time with one edge.

Figure 9: Growing a maximal subtree of the Stallings graph of Kevin's keyboard.



Of course, there are lots of other ways we could have "grown" our maximal subtree. Here are some other possible trees:

Figure 10: Other maximal subtrees of Kevin's keyboard.

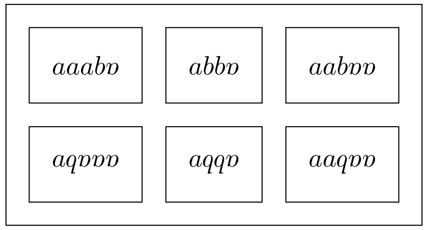


For each edge e *not* in the tree, there is a path from the base vertex to the base vertex that contains e once and whose other edges are all in the tree. The label of this path is a key in the top row on the keyboard we make, and we make one key in the top row for every edge not on the tree.

Exercise 3.7. Show that this path is unique (up to reversing the direction of the path).

Example 3.8. In the rightmost graph in Figure 9, there are three edges not on the tree: one labeled a, a loop labeled b, and another edge labeled b. The key corresponding to the edge labeled a is *aaabɒ*, the key corresponding to the loop labeled b is *aabɒɒ*, and the key corresponding to the other edge labeled b is *abbɒ*. Thus, the corresponding keyboard looks like this:

Figure 11: The keyboard corresponding to the maximal subtree in Figure 9.



Notice that this is different from Kevin's keyboard that we used to make this graph. How then do we know that this new keyboard can type the same words as Kevin's keyboard? Since the keys of the new keyboard are labels of paths from the base vertex to the base vertex, we know that Kevin's keyboard can type them. We still need to show the other direction, namely, given a word that Kevin's keyboard can type, we need to find a way to type them with our new keyboard.

Given a word that Kevin's keyboard can type, we have a path with that label from the base vertex to the base vertex in the Stallings graph. To find out which keys to press on the new keyboard, we use our tree again. We look at the edges of the path not on the tree, in order, and we hit the keys that correspond to those edges.

Example 3.9. Consider the word $abbb\bar{a}b\bar{a}\bar{b}\bar{b}\bar{a}$, which Kevin's keyboard can type, as we can check by looking at the corresponding path on the Stallings graph. If we look at the maximal tree in Figure 9, we can pick out the edges that are *not* on the tree in the path corresponding to our word. We write the corresponding labels in bold: $a\mathbf{b}b\mathbf{b\bar{a}b}\bar{a}\bar{b}\mathbf{\bar{b}}\bar{a}$. Each of these edges, along with the direction we go along it, corresponds to a key on the new keyboard. We can thus convert the five non-tree edges of the path into keypresses: $abb\bar{a}\; abb\bar{a}\; a\bar{b}\bar{a}\bar{a}\bar{a}\; aab\bar{a}\bar{a}\; a\bar{b}\bar{b}\bar{a}$. You can check that this word reduces to our original word.

Exercise 3.10. For each of the four maximal subtrees in Figure 10, find the corresponding keyboard. Compare the keyboards to each other and to Kevin's original keyboard.

Exercise 3.11. For each of the graphs you made in Exercise 2.14, use this method to find a generating set for the subgroup that is different from the one given in Exercise 2.14. You might need to try a couple different trees.

Exercise 3.12. For which graphs that you made in Exercise 2.14 is it possible to find a tree that gives you back the generating set in Exercise 2.14?

Exercise 3.13 (*)**.** For the graphs you made in Exercise 2.14, make another vertex the base vertex and find a generating set for the corresponding subgroup. Compare your answer with your answer in Exercise 3.11. (It will be easier to see what's going on if you use the same tree.)

Exercise 3.14 (*)**.** In Example 3.9, we took a word and used a maximal tree to find a way type that word using the keyboard corresponding to that tree. Show that our method works in general.

That is, consider an arbitrary subgroup and a maximal tree of its Stallings graph, which gives us a keyboard whose keys correspond to edges not on the tree. Given an arbitrary reduced path from the base vertex to the base vertex, we construct a word by hitting the keys corresponding to the edges of the path not on the tree. Show that when we reduce this word, we get back the label of our path.

You may find it helpful to use the uniqueness of reduced paths between two vertices of a tree.

4 Intersections of Subgroups: Keeping in touch

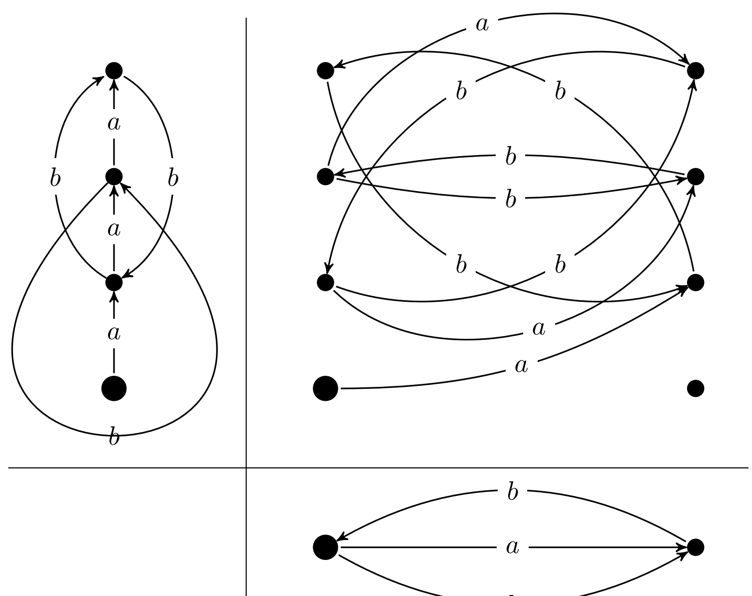
After Mathcamp, Kevin and Nic want to keep in touch. Unfortunately, they have different keyboards! This is a problem if they want to reply about something the other person has typed. Perhaps, however, they have some words they can both type, so they can talk about those. Are there any such words, and how can they find them? To answer this question, we define the *product graph*

Definition 4.1. If Γ and Δ are two X-digraphs, we construct the *product graph* $\Gamma \times \Delta$ as follows:

- For every *pair* containing a vertex of Γ and a vertex of Δ, we draw a vertex of $\Gamma \times \Delta$.
- For every *pair* containing an edge of Γ and an edge of Δ *with the same label*, we draw an edge of $\Gamma \times \Delta$.
- If e is an edge of Γ with origin v and f is an edge of Δ with origin w, then the origin of the edge (e, f) in $\Gamma \times \Delta$ is the vertex (v, w). The terminus of (e, f) is defined similarly.
- The label of (e, f) is the label of e, which is the same as the label of f. (This is why we required the edges of $\Gamma \times \Delta$ to be pairs of edges with the same label.)

Example 4.2. Here is the product of two graphs. Notice that the product graph has $4 \cdot 2 = 8$ vertices, $3 \cdot 1 = 3$ edges labeled a, and $3 \cdot 2 = 6$ edges labeled b.

Figure 12: Constructing the product graph.



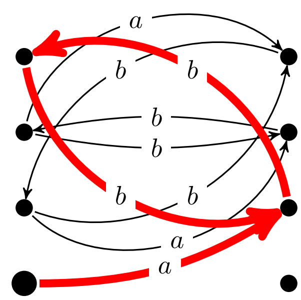
Exercise 4.3. Using Example 4.2 to guide you, convince yourself that a path in the product graph $\Gamma \times \Delta$ corresponds to a path in Γ and a path in Δ with the same labels. Convince yourself of the other direction: given a path in Γ and a path in Δ with the same label, there is a path with the same label in $\Gamma \times \Delta$.

The two factors correspond to Kevin's subgroup $\langle aaa\bar{b}\bar{a}, aaab\bar{a}, aab\bar{a}\bar{a} \rangle$ and Nic's subgroup $\langle ab, a\bar{b} \rangle$. The words they can both type correspond to paths in the two graphs with the same label. By the previous exercise, these correspond to a path in the product graph with the same label. Thus the set of labels of paths from the base vertex to the base vertex in the product graph is the set of words that both Kevin and Nic can type. This would mean that the product graph is the Stallings digraph of the subgroup of words that they both can type, but there is a problem: The product graph is not core.

Definition 4.4. If Γ is an X-digraph with base vertex v, then we construct a new graph $\text{Core}(\Gamma, v)$ called the *core of* Γ *at* v by removing everything that is not on a reduced path from v to v.

Example 4.5. We highlight the core of the product graph in Figure 12 at the base vertex.

Figure 13: The core of the product graph in Figure 12 is the Stallings graph of the intersection of the subgroups.



Once we take the core of the product graph, we find the graph we were looking for: the graph that corresponds to the words that can be typed by both keyboards.

Definition 4.6. Given two subgroups H and K, the set of words in both of them is called the *intersection* of H and K, and is denoted $H \cap K$.

Thus we can restate our result above like this:

$$\Gamma(H \cap K) = \text{Core}(\Gamma(H) \times \Gamma(K), (v, w)),$$

where v is the base vertex of $\Gamma(H)$ and w is the base vertex of $\Gamma(K)$.

Example 4.7. In Examples 4.2 and 4.5, we found the graph of $\langle aaa\bar{b}\bar{a}, aaab\bar{a}, aab\bar{a}\bar{a}\rangle \cap \langle ab, a\bar{b}\rangle$. By inspecting the graph or by using Section 3, we see that $\langle aaa\bar{b}\bar{a}, aaab\bar{a}, aab\bar{a}\bar{a}\rangle \cap \langle ab, a\bar{b}\rangle = \langle abb\bar{a}\rangle$. Thus Kevin and Nic can only type copies of the word $abb\bar{a}$ or its inverse aqqv to each other. They might want to find another way to communicate.

Exercise 4.8. Find the intersection of the following subgroups by computing the product graph, finding its core at the base vertex, and then finding a set of generators.

1. $\langle a, bb\rangle \cap \langle aa, b\rangle$.
2. $\langle aa\rangle \cap \langle aa\rangle$.
3. $\langle ba, c\rangle \cap \langle b\bar{a}, c\rangle$.
4. $\langle ba, c\rangle \cap \langle b\bar{a}, ac\bar{a}\rangle$.
5. $\langle a, bbb, bba\bar{b}\rangle \cap \langle b, aa, abba\rangle$. (Leave lots of space to draw this one.)

Notice that in the last example, the number of generators of the intersection is more than the number of generators in each of the factors. Paradoxically, in order to type a smaller set of words, you might need more keys. In fact, there is a statement called the Hanna Neumann conjecture that states that if H has m generators, K has n generators, and $H \cap K$ has s generators, then $s - 1 \leq (m-1)(n-1)$. Hanna Neumann proved in 1957 that $s - 1 \leq 2(m-1)(n-1)$, but the conjecture remained unresolved until a proof by Igor Mineyev 2011.

Exercise 4.9. If H is a subgroup of K, what is $\mathrm{Core}(\Gamma(H) \times \Gamma(K))$ (with respect to the usual base vertex)?

Exercise 4.10. Using Exercise 4.9, show that if H is a subgroup of K then there is a function from the vertices and edges of $\Gamma(H)$ to the vertices and edges of $\Gamma(K)$ that preserves labels of edges, sends the origin and terminus of an edge to the origin and terminus of its image, respectively, and sends the base vertex of one graph to the base vertex of the other.

Definition 4.11. If ϕ is a function from the vertices and edges of one X-digraph to another that preserves labels of edges and sends the origin and terminus of an edge to the origin and terminus of its image, then ϕ is called a *morphism* of X-digraphs.

Exercise 4.12. Let the alphabet be $X = \{a, b, c\}$, and let K be the entire free group $\langle a, b, c\rangle$. Draw $\Gamma(K)$ and show how to construct a morphism from any other X-digraph to $\Gamma(K)$.

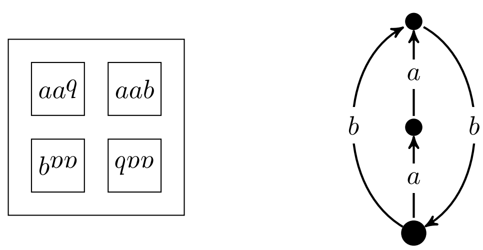
Exercise 4.13. Prove the converse of Exercise 4.10. That is, if H and K are two subgroups and there exists a morphism from $\Gamma(H)$ to $\Gamma(K)$ that sends the base vertex of $\Gamma(H)$ to the base vertex of $\Gamma(K)$, then H is a subgroup of K.

5 Conjugate Subgroups: Modifying keyboards

We now come back to Exercise 3.13. What happens when we move the base vertex? Let's start with an example:

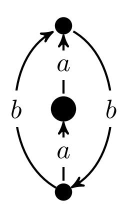
Example 5.1. Susan's keyboard and the corresponding graph look like this:

Figure 14: Susan's keyboard and its Stallings graph.



Let's move the base vertex up. We get a graph that looks like this:

Figure 15: Moving the base vertex in the Stallings graph of Susan's keyboard.



Using Section 3 or just by looking at the graph, we see that $\{a\bar{b}a, aba\}$ is a generating set for the subgroup. Compare it with the original generating set $\{aa\bar{b}, aab\}$. The first letter moved to the end! Another way of thinking about it is that we wrote $\bar{a}$ at the beginning of each generator and a at the end. (Notice that $\bar{a}aa\bar{b}a = a\bar{b}a$ and $\bar{a}aaba = aba$.) Why the letter a and not some other letter? The edge from the original base vertex to the new base vertex is labeled a.

Exercise 5.2. In Example 5.1, move the base vertex to the top and find a generating set.

Definition 5.3. Writing $\bar{a}$ at the beginning of a word w and a at the end is called *conjugating w by a*. Be aware that some other people call this conjugating w by $\bar{a}$. We can also conjugate by words by doing it one letter at a time. For example, conjugating w by ab gives $\bar{b}\bar{a}wab$. (Or, you can think of it as writing ab at the end and then flipping ab over to write qɐ at the beginning.)

An important property of conjugation is that it is an automorphism. That is, if v and w are two words, then we can conjugate both of them by a and then concatenate them or concatenate them and then conjugate the result by a, with the same result, because $(\bar{a}va)(\bar{a}wa) = \bar{a}(vw)a$.

Definition 5.4. If H is a subgroup, then we can conjugate the entire subgroup by conjugating all of its elements. We let $a^{-1}Ha$ denote H conjugated by a. Two subgroups H and K are called *conjugate* to each other if there is a word w such that $H = w^{-1}Kw$.

Example 5.5. In the Example 5.1, we conjugated the generators by a. As a result, the subgroup they generate was also conjugated by a.

Exercise 5.6. In the graph of Susan's keyboard, there are three natural ways of moving the base vertex to the top: along aa, along b, and along $\bar{b}$. As was suggested earlier, this corresponds to conjugating the keys on Susan's keyboard by aa, b, and $\bar{b}$, respectively. Try it. You get three different pairs of generators. Do they all generate the subgroup corresponding to the graph in Figure 14 with the base vertex on top?

Exercise 5.7. What happens if we conjugate the generators of Susan's keyboard by $\bar{a}$? How about ab? Compare the resulting Stallings graphs.

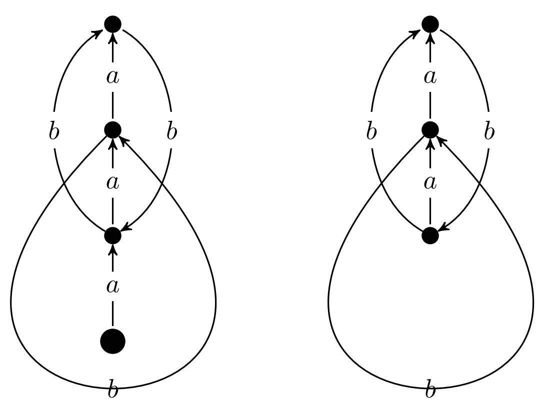
Exercise 5.8. Think of a simple way to take a path from one vertex back to itself and turn it into a path from another vertex back to itself. What happens to the labels of the path?

Exercise 5.9. Given two finitely generated subgroups H and K, how could you test if they are conjugate to each other?

Definition 5.10. Given an X-digraph Γ that is core with respect to some vertex, the *type* of Γ, denoted $\mathrm{Type}(\Gamma)$ is the graph we get by removing any vertices with just one edge, as many times as necessary until every vertex of the graph has at least two edges. Equivalently, we take the intersection of $\mathrm{Core}(\Gamma, v)$ over all vertices v of Γ. The type of a graph does *not* have a designated base vertex.

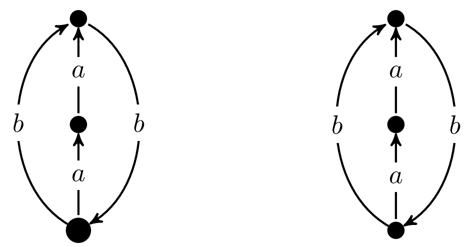
Example 5.11. Here is the graph of Kevin's keyboard and the type of that graph:

Figure 16: The Stallings graph of Kevin's keyboard and the type of the Stallings graph of Kevin's keyboard.



Example 5.12. Here is the graph of Susan's keyboard and the type of that graph. Notice that nothing changes, except that there is no longer a special base vertex.

Figure 17: The Stallings graph of Susan's keyboard and the type of the Stallings graph of Susan's keyboard.



As you might have realized from the above exercises, two subgroups H and K are conjugate to each other if and only if the graphs $\text{Type}(\Gamma(H))$ and $\text{Type}(\Gamma(K))$ are the same. The path from the base vertex of $\Gamma(H)$ to the base vertex of $\Gamma(K)$ is the word that is conjugated by.

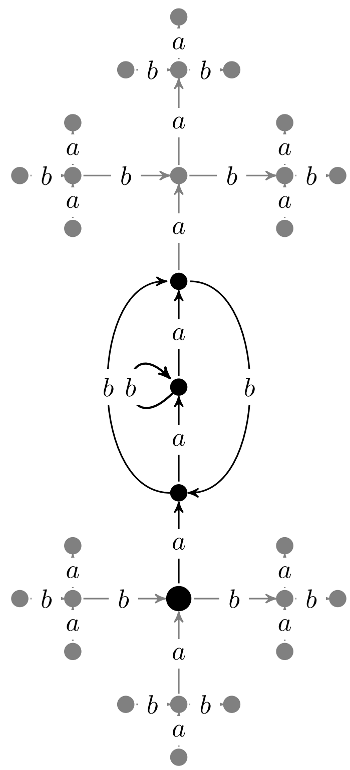
6 Graphs of Infinitely Generated Subgroups and Applications

So far, we've used Stallings graphs to create algorithms to test membership, compute intersections, and test for conjugacy. We can also use Stallings graphs to prove theorems about subgroups of free groups by translating properties of subgroups into properties of the corresponding Stallings graphs. Before we do that, though, we need to define the Stallings graphs of arbitrary subgroups of $F(X)$, even ones that are not finitely generated.

When testing if a word is in a subgroup, we moved around the Stallings graph, and sometimes we got stuck when a vertex did not have an edge with the appropriate label and direction. We can add in the "missing" edges. That is, whenever we have a vertex that does not have an edge with a certain label and direction, we attach that missing edge and create a new vertex at the other end of the edge. Of course, the new vertex is missing some edges, so we have to repeat this process indefinitely. At the end, what we have is a Stallings graph with some infinte trees attached.

Example 6.1. Here is an example of the result of this process with the Stallings graph of Kevin's keyboard.

Figure 18: The Stallings graph of Kevin's subgroup $\langle aaa\bar{b}\bar{a}, aaab\bar{a}, aab\bar{a}\bar{a}\rangle$, in black, with infinite gray trees added so that every vertex has edges of every label and direction. The graph together with the trees is the covering graph of this subgroup.



We call this new graph the *covering graph* of the subgroup. It has the property that any word is the label of some path starting from the base vertex. That is, when reading a word by going along the graph, you will never get stuck. Another way of saying this fact is that the graph is *X-regular*, which means that every vertex has exactly one edge with each label and direction.

Also, the covering graph has the property that Stallings graphs have that a word is in the subgroup if and only if it is the label of a reduced path from the base vertex to the base vertex. Indeed, if we take the core of the covering graph with respect to the base vertex, the trees all get removed, and we get the Stallings graph back.

There is a second way of defining a covering graph that does not use Stallings graphs:

Definition 6.2. A *right coset* of a subgroup H of a group F is the set $\{hw \mid h \in H\}$, where w is some word in F. The coset is written Hw.

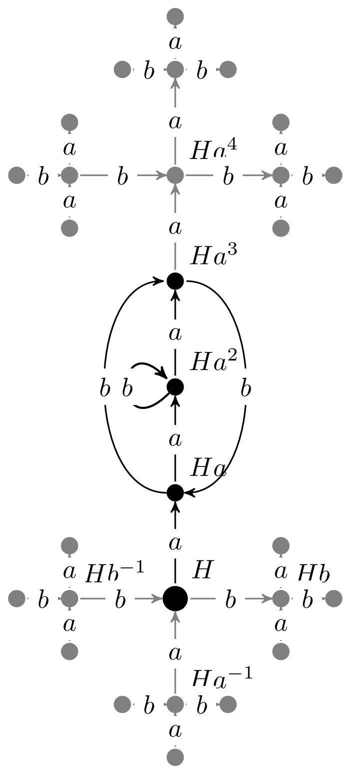
Definition 6.3. The *covering graph* of a subgroup H of F is an X-digraph with

- a vertex for each right coset of H, and
- an edge labeled x from the coset Hv to the coset Hvx for all cosets Hv and all labels x in X.

It is not immediately obvious that this definition of covering graphs agrees with our earlier notion of attaching trees to Stallings graphs, though we can see how it might be true in our example:

Example 6.4. Here we have the covering graph of Kevin's subgroup as before, with the vertices corresponding to cosets. Note that the way that we write a coset is not unique. For example, for this subgroup, the coset Ha^3 is equal to Hab, and the coset Ha^2 is equal to Ha^2b.

Figure 19: The covering graph of Kevin's subgroup $\langle aaa\bar{b}\bar{a}, aaab\bar{a}, aab\bar{a}\bar{a}\rangle$, with the vertices labeled with the cosets they represent.



We can see that, for example, there is an edge from Ha to Ha^3 labeled b. We can check that, indeed, $Ha^3 = Hab$, because $a^3b^{-1}a^{-1} \in H$, and so $Ha^3(ab)^{-1} = H$.

To show that this notion of covering graph coincides with our earlier notion, we first prove the following:

Proposition 6.5. *Given a subgroup H of a free group F, the language of the covering graph of H (defined with cosets) is H itself.*

Proof. Given a word $w \in H$, let $w = w_1w_2 \cdots w_k$, where the w_i are letters or inverse letters. Reading the word along the graph starting at the base vertex, we will move along the vertices H, Hw_1, Hw_1w_2, all the way to $Hw_1 \cdots w_k = Hw$. But $w \in H$, so $Hw = H$. Hence, once we're done reading the word, we're back at the base vertex, giving us a reduced path from the base vertex to the base vertex with label w.

Conversely, given a reduced path from the base vertex to the base vertex with label $w = w_1 \cdots w_k$, the sequence of vertices is, as before, H, Hw_1, Hw_1w_2, all the way to $Hw_1 \cdots w_k = Hw$. But this path ends back at the base vertex, so $Hw = H$, and so $w \in H$. □

Now, we can prove the next theorem:

Theorem 6.6. *Given a finitely generated subgroup H of a free group F, the core of the covering graph of H is the Stallings graph of H.*

Proof. The covering graph is folded with language H. Thus, the core of the covering graph is a folded core graph with language H. By Theorem 3.3, this graph must, in fact, be the Stallings graph of H. □

Corollary 6.7. *Our two constructions of the covering graph by attaching trees to the Stallings graph and by using right cosets are the same.*

Proof. The theorem tells us that the Stallings graph is the core of the covering graph. The only way to extend the Stallings graph without adding any new reduced paths is by attaching trees, and one can check that there is only one way to attach trees so that every vertex has exactly one edge with every label and direction. Thus, the Stallings graph is the core of exactly one X-regular graph. Both of our constructions of the covering graph are X-regular and their cores are the Stallings graph, so we conclude that our two constructions of the covering graph are the same. □

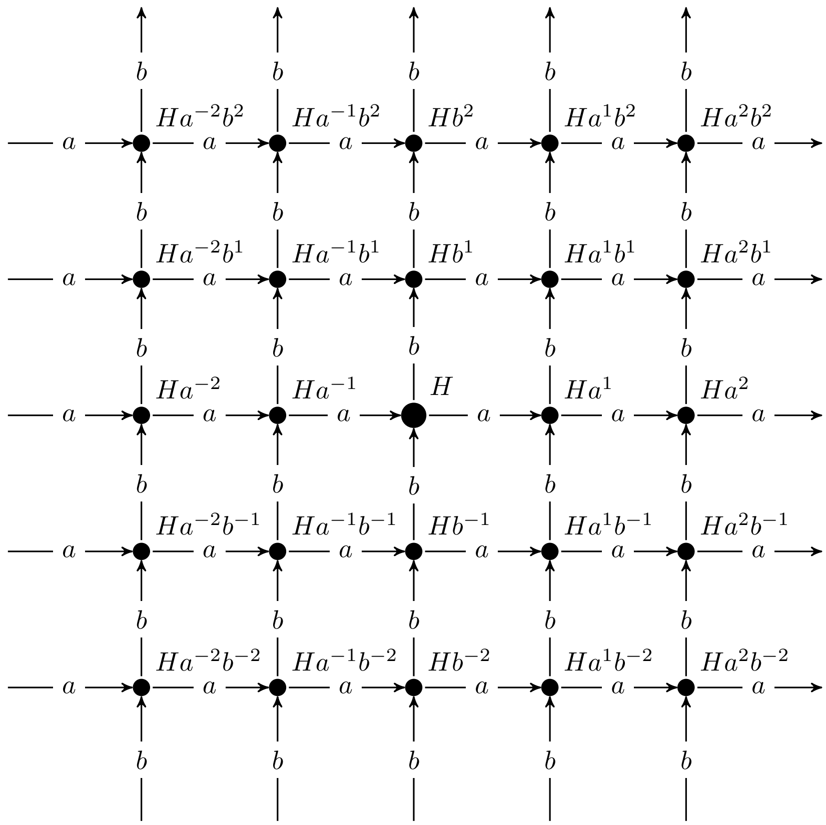
Note that in order to construct the covering graph via cosets and take its core, we never use the fact that H is finitely generated. In fact, we can drop this assumption, and use this procedure as the *definition* of Stallings graphs for infinitely generated subgroups, and our theorem says that this definition would agree with our earlier definition of the Stallings graph for finitely generated subgroups.

Definition 6.8. Given a subgroup H of the free group $F(X)$, not necessarily finitely generated, the *Stallings graph* of H is the core of the covering graph of H with respect to the base vertex corresponding to the identity coset H.

Example 6.9. An important example of the Stallings graph of an infinitely generated subgroup H is in the free group generated by two generators a and b when H is the set of words where the number of times a appears is the same as the number of times a^{-1} appears, and the number of times b appears is the same as the number of times b^{-1} appears.

One can check that there is a coset of H for every pair of integers, where the first integer represents how many more times a appears than a^{-1}, and the second integer represents how many more times b appears than b^{-1}. One can verify that H is closed under concantenation and taking inverses. The covering graph then looks like this:

Figure 20: The covering graph of the subgroup H containg all words in which a appears the same number of times as a^{-1} and b appears the same number of times as b^{-1}.



This graph is core, so the Stallings graph of H is the same graph.

Now that we've defined the Stallings graph for arbitrary subgroups, it might be useful to characterise when a subgroup actually is finitely generated, in terms of its Stallings graph. The answer turns out to be simple.

Theorem 6.10. *Let H be a subgroup of a free group $F(X)$. Then H is finitely generated if and only if the Stallings graph $\Gamma(H)$ is finite.*

Proof. If H is finitely generated, then we can use our initial method of constructing the Stallings graph via drawing cycles for the gennerators and folding. The graph we get this way is finite.

Coversely, if we have a finite folded graph, then, in particular, there are finitely many edges that are not on some maximal subtree. As discussed in Section 3, we conclude that there are finitely many generators for the subgroup. □

OK, so we've characterised the finitely generated subgroups in terms of their Stallings graphs. What other properties of subgroups can we characterise in terms of the Stallings graphs? Here's one:

Definition 6.11. The *index* of a subgroup H in a free group F is the number of cosets of H, denote $[F : H]$.

In general, we can't expect the index of a subgroup to be finite. The concept of a finite index subgroup is fairly important in group theory: Many theorems have the phrase "finite index" somewhere in either the assumptions or the conclusions of the theorem (or both!). Can we determine whether a subgroup of a free group has finite index in terms of its Stallings graph? Indeed, we can:

Theorem 6.12. *Let H be a subgroup of a free group $F(X)$ where the alphabet X is finite. Then H has finite index in F if and only if the Stallings graph $\Gamma(H)$ is finite and X-regular.*

Proof. The subgroup H having finite index is the same thing as saying that the covering graph of H has finitely many vertices, since, by definition, the covering graph has a vertex for every coset of H.

If the covering graph is finite, then I claim that it is already core. Indeed, consider any edge labeled x from a vertex v_0 to a vertex v_1. We need to find a reduced path that contains this edge. But since covering graphs are X-regular, there is an edge labeled x from v_1 to some vertex v_2. Likewise, there is an edge labeled x from v_2 to some vertex v_3, and so forth. Since the graph is finite, we will eventually repeat a vertex. One can check that the folded condition implies that the first time we repeat a vertex we must actually be at the initial vertex v_0. So now we have a reduced path p from v_0 to v_0. We can create a path from the base vertex to the base vertex like we did in Section 5 by going along a path q from the base vertex to v_0, along our cycle p, and then back via the inverse q^{-1}. One can check that when we reduce this path qpq^{-1}, we still go along all of the edges of p. Hence, our arbitrary edge is on a reduced path from the base vertex to the base vertex, and hence is in the core of the graph. We conclude that the covering graph is core.

Since the covering graph is core, it is equal to the Stallings graph. In particular, the Stallings graph is finite and X-regular.

Conversely, say the Stallings graph is finite and X-regular. Since the graph is X-regular, when we attach the "missing" edges to make the covering graph, we have no edges to attach. Therefore, once again, the Stallings graph is equal to the covering graph. In particular, the covering graph is finite, as desired. □

Our characterisations of finite index subgroups and finitely generated subgroups yield an immediate corollary.

Corollary 6.13. *Let H be a subgroup of a free group $F(X)$ where the alphabet X is finite. If H has finite index in F, then H is finitely generated.*

Proof. If H has finite index, then $\Gamma(H)$ is finite by Theorem 6.12, and so H is finitely generated by Theorem 6.10. □

This result seems counterintuitive. Finite index subgroups are "almost as big" as the entire group F. Indeed, if H has index d in F, then that means that d "copies" of H cover all of F. On the other hand, finitely generated subgroups are built from a finite list of words, so they are intuitively small. Our corollary then says that if a subgroup is big, then it must be small.

Instead, perhaps a better intuition is that finitely generated groups are tidy whereas infinitely generated groups are messy. A finite index subgroup is close to the entire group F, which makes it pretty tidy.

Our result claims that if H has finite index, then it is finitely generated. But how many generators does H have? Can we answer this question in terms of the index of H and the number of letters in the alphabet X?

Exercise 6.14. Let F be a free group over an alphabet X with n letters, and let H be a subgroup of $F(X)$ with index d. Prove that the number of generators of H is exactly $(n-1)d+1$. This result is known as Schreier's formula named after the early 20th century Austrian mathematician Otto Schreier. Hint: Count the number of vertices and edges of the Stallings graph.

We can also use Stallings graphs to say something about the intersection of two finite index subgroups.

Theorem 6.15. *Let H and K be two subgroups of a free group F. If H and K have finite indices c and d, respectively, in F, then the index of $H \cap K$ is also finite and at most the product cd.*

Proof. As discussed in the proof of Theorem 6.12, for finite index subgroups, the covering graphs and Stallings graphs are the same. Thus, $\Gamma(H)$ has c vertices and $\Gamma(K)$ has d vertices, so the product graph $\Gamma(H) \times \Gamma(K)$ has cd vertices. One can check that since $\Gamma(H)$ and $\Gamma(K)$ are X-regular, the product $\Gamma(H) \times \Gamma(K)$ is also X-regular.

From Section 4, we know that $\Gamma(H \cap K)$ is the core of $\Gamma(H) \times \Gamma(K)$. As we showed in the proof of Theorem 6.12, a connected finite X-regular graph is automatically core, so taking the core of $\Gamma(H) \times \Gamma(K)$ amounts to taking the connected component containing the base vertex. Therefore, the core of $\Gamma(H) \times \Gamma(K)$ is still X-regular.

We conclude that $\Gamma(H \cap K)$ is X-regular and finite with at most cd vertices, so $H \cap K$ has finite index at most cd in F. □

As a side note, this theorem is true even if F is replaced by an arbitrary group that is not necessarily free.

Exercise 6.16. Use this result about the index of an intersection along with Schreier's formula to help you find examples that show that the Hanna Neumann bound is as strong as possible. That is, given arbitrary positive integers r and s, construct a subgroup H with r generators and a subgroup K with s generators such that the number of generators of $H \cap K$ is exactly $(r-1)(s-1)+1$.

We now return to the question of conjugating subgroups. As we discussed earlier, conjugating a subgroup corresponds to moving the base vertex in the Stallings graph. More precisely, conjugating by x moved the base vertex along the edge labeled x. Of course, there was a caveat: Sometimes, the base vertex didn't have an edge labeled x. Then we needed to move the base vertex "off" the Stallings graph to a new vertex attached to the old graph by a single edge.

Now that you know about covering graphs, you see the big picture. What's actually going on is that, as we conjugate the subgroup, the base vertex moves around the covering graph. When we go back to the Stallings graph by taking the core, most of the covering graph disappears, and we're left with the type of the graph possibly with a spur attaching it to the base vertex. Here we want to use the more general definition of the type of a graph, so that we can use it both for Stallings graphs and for covering graphs:

Definition 6.17. Given an X-digraph Γ, the *type* of Γ, denoted $\text{Type}(\Gamma)$, is the intersection of $\text{Core}(\Gamma, v)$ over all vertices v of Γ. The type of a graph does not have a designated base vertex.

Roughly speaking, the type of a graph contains the interesting part of a graph. Namely, it contains all of the cycles and cuts off the spurs and trees. Note that the type of the covering graph of a nontrivial subgroup is the same as the type of its Stallings graph.

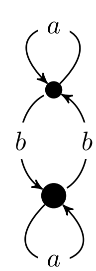
In group theory, a very important concept is a subgroup that doesn't change when it is conjugated:

Definition 6.18. A subgroup N of a free group F is *normal* if $g^{-1}Ng = N$ for all $g \in F$. Equivalently, we have $gN = Ng$ for all g.

If you conjugate an element of a normal subgroup N, it will in general become a different element, but it will stay inside N. It is the set N that remains fixed by conjugation.

Example 6.19. The subgroup $H = \langle a, bb, bab \rangle$ is normal in the free group over the alphabet $\{a, b\}$. Here is its Stallings graph:

Figure 21: The Stallings graph of the subgroup $H = \langle a, bb, bab \rangle$.



Normal subgroups are very important in group theory because we can construct another group called the quotient group by "setting everything in N to the identity". If the subgroup is not normal, we quickly run into problems. After all, if we conjugate the identity, we should get the identity back.

Like we did for finitely generated and finite index subgroups, we can characterise normal subgroups of a free group F using Stallings graphs.

Theorem 6.20. *Let H be a nontrivial subgroup of a free group $F(X)$. Then H is normal if and only if the Stallings graph $\Gamma(H)$ is X-regular and looks the same no matter which vertex is selected as the base vertex.*

Proof. As discussed earlier, conjugating a subgroup corresponds to moving the base vertex of the covering graph. Assume that H is normal. Since conjugating H gives us H again, we know that the covering graph looks the same no matter which vertex is picked as the base vertex. In particular, the type of the covering graph must be either the entire covering graph or empty, because otherwise the graph would look different based at a vertex in the type or based at a vertex not in the type. We assumed that H is nontrivial, so the type of the covering graph cannot be empty.

Therefore, the covering graph is equal to its type, and hence the covering graph is core with respect to every vertex. Since the Stallings graph is the core of the covering graph with respect to the base vertex, we conclude that the Stallings graph of H is equal to the covering graph of H. In particular, the Stallings graph of H is X-regular and looks the same no matter which vertex is selected as the base vertex.

Conversely, assume that $\Gamma(H)$ is X-regular and looks the same from each vertex. As discussed earlier, if the Stallings graph is X-regular, then there are no missing edges to attach, and so the covering graph is equal to the Stallings graph. As a result, the covering graph looks the same no matter which vertex is selected as the base vertex. Therefore, if we conjugate H, we move the base vertex in the covering graph, and are left with the same graph. Since H and its conjugate have the same covering graph (and hence the same Stallings graph), we conclude that H is equal to its conjugate, as desired. Therefore, H is normal. □

It is important to note that the notion of a normal subgroup depends heavily on the free group it is in. For example, the subgroup $\langle a, bb, bab \rangle$ in Example 6.19 is normal in the free group over the alphabet $\{a, b\}$, but it is not normal in the free group over the alphabet $\{a, b, c\}$. Indeed, its Stallings graph is $\{a, b\}$-regular, but not $\{a, b, c\}$-regular.

Earlier, we used our characterisations of finitely generated and finite index subgroups to show that a finite index subgroup is finitely generated. We can now have a partial converse.

Corollary 6.21. *Let N be a nontrivial normal subgroup of a free group F. Then N is finitely generated if and only if it has finite index in F.*

Proof. If N has finite index, then we know that it is finitely generated by Corollary 6.13. Conversely, if N is finitely generated, then $\Gamma(N)$ is finite by Theorem 6.10. Since N is nontrivial and normal, the Stallings graph $\Gamma(N)$ is X-regular by Theorem 6.20. Therefore, N has finite index by Theorem 6.12. □

As a side remark, if N is a normal subgroup in a free group F, then the Stallings graph of $\Gamma(N)$ is more commonly known as the *Cayley graph* of the quotient group F/N. The Cayley graph of a group is a very important concept in geometric group theory. In Example 6.9, one can check by examining the graph that the subgroup is normal, and the Stallings graph pictured is the Cayley graph of the quotient group, which is isomorphic to $\mathbb{Z} \oplus \mathbb{Z}$.