The word problem in group theory

Christian Perfect

April 29, 2008

1 Some group theory

Recall that a group G is a set of elements with a multiplicative operation $\star$ such that for each $g_1, g_2 \in G$, $g_1 \star g_2 = g$, with $g \in G$; G has an identity element id (or 1) such that $g \star id = id \star g = g$, $\forall g \in G$ and every g has a unique inverse, g^{-1}, where $gg^{-1} = 1$. Note that it is not necessarily true that a group is commutative, i.e. $g_1 \star g_2 = g_2 \star g_1$.

The trivial group is the group consisting of only one element, the identity. Its group operation is $id \star id = id$.

From now on we use juxtaposition to denote multiplication, rather than $\star$, i.e. $x \star y$ will be instead written xy.

A group can be defined by a *presentation*, which is a set of generators X and a set of relations R on those generators. We write

$$G = \langle X \,;\, R \rangle.$$

Every element of a group presented this way can be written as a product of the generators and their inverses. The relations define which strings of generators are equivalent. In a presentation the set R is almost always a subset of all of the true relations on the group, from which all of the others can be deduced. The relations like $xx^{-1} = 1$ and $1x = x1 = x$, which hold for every group, called here the *free relations*, are taken as read in every presentation, so there is no need to include them in R. Here are some examples of group presentations:

$$\begin{aligned} \langle x \,;\, \emptyset \rangle &\quad \text{the infinite cyclic group,} \\ \langle x \,;\, x^5 = 1 \rangle &\quad \text{the cyclic group of order 5, or } \mathbb{Z}_5, \\ \langle x, y \,;\, xy = yx \rangle &\quad \text{the abelian group.} \end{aligned}$$

A *word* in G is a string $x_1x_2 \dots x_{n-1}x_n$, with each $x_i \in X^{\pm}$, with $X^{\pm} = X \cup X^{-1}$, representing the product $x_1x_2 \dots x_{n-1}x_n \in G$. Note that a single element of G can be represented by more than one word, for example $xx^{-1}x$ and x represent the same element. When the meaning is ambiguous, we will use $w_1 = w_2$ to mean w_1 and w_2 are the same word, and $w_1 =_G w_2$ to mean that w_1 and w_2 represent the same element in the group G. When you have the same generator several times in a row, it is useful for clarity to use superscript notation, eg. $xxyyyx^{-1}x^{-1}$ can be written instead as $x^2y^3x^{-2}$.

If X is a set of letters, we say $X^\star$ is the set of words constructed from X, including the empty word.

It is of course possible to write every relation in the form $w = 1$, where w is a word. You can then write R as simply the list of left-hand-sides of the relations, for example:

$$G = \langle x, y, a \,;\, xa = ay, a^2 = 1 \rangle$$

can be written instead

$$G = \langle x, y, a \,;\, xay^{-1}a^{-1}, a^2 \rangle$$

Note that a group presentation is not unique - there can be other presentations that define the same group.

A group is said to be *finitely generated* if the set of generators is finite. It is said to be *finitely presented* if the set of relations is also finite.

A group is *free* if R is the empty set. It is *virtually free* if it has a free subgroup of finite index.

A group H is a *subgroup* of G if $H \subseteq G$ and $h_1, h_2 \in H \Rightarrow h_1h_2^{-1} \in H$, i.e. the inverse of every element in H is in H and the product of every pair of elements of H is in H.

A subgroup N of G is *normal* if $n \in N, g \in G \Rightarrow gng^{-1} \in N$. The *normal closure* $\{n\}_G$ of an element n in G is the set $\{g^{-1}ng | \forall g \in G\}$.

The *left cosets* of a subgroup H in G are the sets $gH = \{gh \,;\, h \in H\}$, $g \in G$. Clearly, by the definition, every element of G belongs to a unique left coset. Similarly, the *right cosets* of H in G are the sets $Hg = \{hg \,;\, h \in H\}$, $g \in G$. The *index* of H in G, written $[G : H]$, is the number of distinct left (or right) cosets.

Given a subgroup H of G, the *quotient group* G/H of G over H is defined to be the set of all left cosets of H in G, with the group operation defined as $(aH)(bH) = (ab)H$, for each $aH, bH \in G/H$.

If there are t distinct left (or right) cosets in G/H, the *coset representatives* for H in G are a selection of elements $b_1, b_2, \ldots, b_t$ such that each b_i belongs to a different coset.

Lemma 1.1 *If we are given a group G, a finite index subgroup H of G with generating set Y and a word w, then as w is read, one letter at a time, it can be rephrased as a word in the form $w_y b$, where w_y is a word in $Y^\star$ and $b \in B$ where B is the set of the coset representatives of H in G.*

PROOF: Given two different generating sets for G, we can rewrite the generators of one as words from the other, so we are free to choose the generating set we will use. Let $X = Y \cup B$ be the generating set for G. Start with the empty word, ϵ, and add to it the first letter of w. No matter what letter is read, this word is in the desired form.

Now suppose we have already read a portion of w and it has been rephrased as a word $w_y b$, for some $w_y \in Y^\star$, $b \in B$.The next letter we read will either be a member of Y or B.

First suppose it is some $y \in Y$. Then we have the word $w_y by$. by belongs to some right coset Hb, so is equivalent to some $w'_y b'$. So the word we have read is equivalent to $w_y w'_y b'$, which is in the desired form.

Suppose instead that we read some $b' \in B$. Then clearly the resulting word, wbb', is equivalent to some wb'', which is in the desired form. □

If we have an element $g \in G$, the *conjugate of g by $x \in G$* is defined to be xgx^{-1}. Conjugation is a homomorphism, that is, the product of the conjugations of two elements is equal to the conjugation of the product of the elements. This is easy to show: assume we have $g, h \in G$ and conjugate by x: $xgx^{-1}xhx^{-1} = xghx^{-1}$. An important fact to note is that the conjugate of the identity element is the identity element itself, and the conjugate of any nonidentity element must be a nonidentity element.

2 The word problem

Recall that, given a finitely generated group $G = \langle X \,;\, R\rangle$, the set of *words* is the set of strings w of generators and their inverses $x \in X^{\pm}$. We denote by ϵ the empty word.

For example, for the group $G = \langle x, y, a \,;\, xa = ay, a^2 = 1\rangle$, some words are

$$xyx^{-1}, x^2y, a^4, \epsilon$$

We might want to know which words are equivalent to the identity element of G. This is called the word problem on G. 'Word problem' also refers to the set of words equivalent to the identity, written $WP(G)$. In the group G given before, some words in $WP(G)$ are:

$$aa^{-1}, x^2ay^{-1}ax^{-1}, xa^2x^{-1}$$

It would be interesting to be able to say for a given word w whether it is in $WP(G)$ or not, or to be able to construct all of $WP(G)$ for a given group. Clearly the former problem is a computation problem, so we would expect to be able to construct a machine for a given group that would take words as its input and either accept them if they are equivalent to the identity element or reject them if they are not.

3 Automaton Theory

3.1 Finite State Automata

A *deterministic finite state automaton* [11] or FSA is a machine which takes as input a string w of letters from an alphabet Σ. It begins in a starting state i and at each state s reads and deletes the leftmost character x of w and changes to the state $\delta(s, x)$. If the machine is in an *accepting state* when it reaches the last letter of w it accepts w, otherwise it rejects w. An FSA can be thought of as a directed graph with the nodes representing states and the edges labelled by letters. You can then say a word is accepted if it labels a path from the start state to an accepting state.

A *non-deterministic finite state automaton* is like a deterministic FSA but instead of just one starting state there is a set i of starting states, and from each state $\delta(s, x)$ is a set of one or more destination states. So then a word is accepted if it labels one or more paths from some starting state to any accepting state.

Formally, a finite state automaton is a 5-tuple $M = (Q, \Sigma, \delta, i, F)$, where Q is a set of states, Σ is an alphabet, δ is a map $\delta : Q \times \Sigma \to Q$, i is a set of initial states and F is a set of accepting states.

Lemma 3.1 *The word problem of a finite group can be solved on a finite state automaton. [12]*

Proof: For a given finite group G generated by the (necessarily finite) set X, construct an FSA with $Q = G$, $\Sigma = X$, $i = 1$, $F = 1$ and $\delta(g, x) = gx$. □

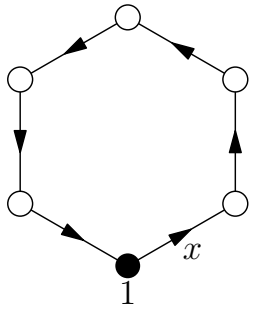


Figure 1: A finite state automaton accepting the word problem of the cyclic group $\mathbb{Z}_6 = \langle x \,;\, x^6 = 1 \rangle$
. The solid dot represents the accepting state.

3.2 Push-down Automata

A *push-down automaton* [11] is a FSA with an additional *stack* which is read from and added to during each move. The stack is a list of letters, which the machine can only read the last letter from or add a letter (or string of letters) to the end of. A PDA can then at each stage make a move like a FSA, reading a letter from the input and depending on that, the current state and the last letter of the stack, move to a different state and/or add a new string to the stack. It can also make moves where nothing is read from the input and what happens depends only on the current state and the stack. In this case we say the empty string ϵ was read from the input, and an *ϵ-move* is made. The PDA accepts the input if it is in an accepting state with the stack empty after it has read the last letter of the input. Otherwise it rejects the input.

So a PDA is a 6-tuple $M = (Q, \Sigma, \Gamma, \delta, i, F)$ where Q, Σ, i and F are as before, but Γ is the alphabet of letters that can be put on the stack, and δ is redefined as $\delta : Q \times \Sigma \cup \{\epsilon\} \times \Gamma \to Q \times \Gamma^\star$.

Lemma 3.2 *The word problem of a free group can be solved on a pushdown automaton.*

PROOF: Given a free group $G = \langle X \,;\, \emptyset \rangle$, every word is of the form $x_1^{a_1} x_2^{a_2} \dots x_n^{a_n}$ with $x_i \in X \cup X^{-1}$ for $i = 1 \dots n$. It is easy to see that a word w represents the identity element of G if and only if w can be reduced to the empty string by successive deletions of subwords of the form xx^{-1}, $x \in X \cup X^{-1}$. Create a PDA with only one state, which is both the initial and accepting state. At each move, read a letter x from the input. If x^{-1} is on the end of the stack, delete it, otherwise add x to the stack. □

Lemma 3.3 *The word problem of a finitely generated virtually free group can be solved on a pushdown automaton. [12, 13]*

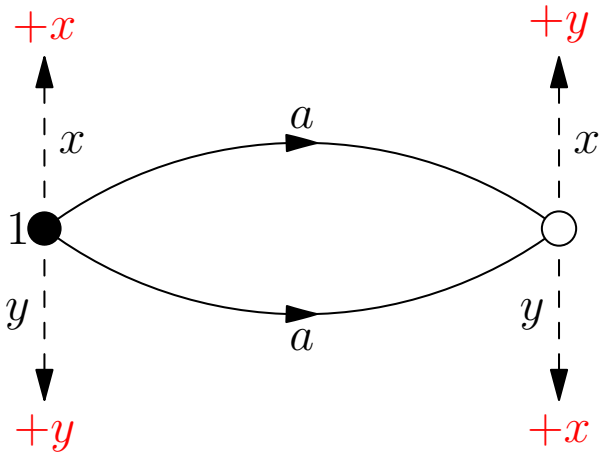


Figure 2: A pushdown automaton accepting the word problem of the virtually free group $G = \langle x, y, a \,;\, xa = ay, a^2 = 1 \rangle$. Dotted lines labelled by a letter z_1 and ending in $+z_2$ indicate a move adding z_2 to the stack after z_1 is read from the input.

PROOF: Let G be a virtually free group with a free subgroup H of finite index. It is well-known that any subgroup of G of finite index contains a subgroup N which is normal in G and itself has finite index. Furthermore, a theorem proved by Schreier says that any subgroup having finite index in a finitely generated group is itself finitely generated, so N is finitely generated by a set Y (which we can express as a set of words over X).

Let $B = G/N$, the quotient group of G over N. Let $y_1, \ldots, y_n$ be the free generators of N, and $b_0, \ldots, b_t$ be the coset representatives for N in G, with b_0 representing the identity element of B. To solve the word problem for a word w, create a PDA with a state q_i for each b_i, to keep track of the image of w in B, and where q_0 is the initial and accepting state.

So at any one stage in the running of our PDA, the word we have read is equivalent to one of the form $w_y b$, where $w_y \in Y^\star$ (w freely reduced, i.e. shortened as much as it can be using only the free relations) and $b \in \{b_0, b_1, \ldots, b_t\}$, meaning at that stage we have w_y on the stack and are in the state labelled by b. Now suppose we read a letter x. The word $w_y b x$ is equal to a word $w_y w'_y b'$, $w'_y \in Y^\star$, $b' \in \{b_0, b_1, \ldots, b_t\}$. Using only the stack we can find the free reduction of $w_y w'_y$ and end up with that on stack. We also move to the state representing b'. If, once we have read all of the input

and reduced the stack, we are in the state q_0 with an empty stack, we can say the input is equivalent to the identity and it is accepted.

□

In fact, the converse of this lemma is also true; if you can solve the word problem of a group on a pushdown automaton then it is necessarily virtually free. The proof of this is considerably longer than the previous lemma, and requires knowledge of *context-free languages*.

4 Languages and grammars

Since we're talking a lot about words, it would be useful to have a more formal definition of what they are and how they can be made. Following Hopcroft and Ullman [11], we formally define a *language* to be a set of words whch are made of letters from a certain *alphabet* and which obey a set of *syntax rules* specifying which words are valid members of the language.

Rather than using the syntax to see which words belong to a language, we could define a *grammar* which describes how to construct all of a language's words. Formally, a grammar is a set of p̂roduction rules describing how strings of symbols can be transformed. To generate a word, start with a *start symbol* and then successively apply production rules to change the string. The symbols in a grammar are either *terminal* or *nonterminal*. Terminal symbols can not be transformed into any other strings.

Formally, a grammar consists of a set N of nonterminal symbols (also called *variables*) including a single start symbol S, a set Σ of terminal symbols disjoint from N, and a set of production rules of the form $(\Sigma \cup N)^\star N(\Sigma \cup N)^\star \to (\Sigma \cup N)^\star$, that is, the left-hand side of every production rule contains at least a nonterminal symbol, but the right hand side is allowed to be the empty string (represented by ϵ).

A *context-free grammar* is one in which the left-hand side of every production rule is a single nonterminal symbol. It is called context-free because every rule can be applied wherever its left-hand side symbol appears, regardless of the rest of the string. A language is said to be context-free if it is generated by a context-free grammar.

A context-free grammar is said to be in *Greibach normal form* if all of the production rules are of the form $A \to \alpha X$ or $S \to \epsilon$, where A is a nonterminal symbol, α is a terminal symbol and X is a (possibly empty) string of nonterminal symbols.

A context-free grammar is said to be in *Chomsky normal form* if all of the production rules are of the form $A \to BC$, $A \to \alpha$ or $S \to \epsilon$, where A,B and C are nonterminal symbols and α is a terminal symbol.

Sometimes these kinds of grammars are defined not to include rules of the form $A \to \epsilon$, so that applications of production rules can only make a string longer or the same length, making some proofs a little easier. However, we need ϵ to be in our languages, so we will not follow this convention.

If $\alpha A \gamma$ is a string in a language produced by a context-free grammar C, and $A \to \beta$ is a production rule in C, we say there is a *derivation* $\alpha A \gamma \Rightarrow \alpha\beta\gamma$. If there is a series of derivations$\eta \Rightarrow \eta_1, \eta_1 \Rightarrow \eta_2, \ldots, \eta_{m-1} \Rightarrow \eta_m, \eta_m \Rightarrow \theta$, we write $\eta \stackrel{\star}{\Rightarrow} \theta$.

A variable A is said to be *useful* if there is a derivation $S \stackrel{\star}{\Rightarrow} \alpha A \beta \stackrel{\star}{\Rightarrow} w$ of a string of terminals which uses A. If A is not useful, it is *useless.* The language $L(A)$ generated by a variable A is defined to be the set of words which are derivable from A, that is, $L(A) = \{w | w \in \Sigma^\star and A \stackrel{\star}{\Rightarrow} w\}$. $L(A)$ is non-empty if A is useful.

A *reduced grammar* is one which contains only useful variables. Every context-free language can be generated by a grammar in Greibach normal form or Chomsky normal form.

Lemma 4.1 *The language generated by a context-free grammar C can also be generated exactly by a context-free grammar C' in Greibach normal form.*

PROOF: Consider a derivation $A \to w$, with $w = w_1 \ldots w_n \in (\Sigma \cup N)^\star$. First of all, if w_1 is a nonterminal symbol, rewrite w as ϵw. Suppose w_n is a nonterminal symbol. Then we can write $w = w_1 \ldots w_m N_1 \ldots N_l$, with $N_i \in N$ for $i = 1 \ldots l$ and $w_m \in \Sigma$. Rewrite the derivation as

$$\begin{aligned} A &\to \epsilon BC \\ B &\to w_1 \ldots w_{m-1} \\ C &\to w_m N_1 \ldots N_l. \end{aligned}$$

Now suppose instead that w_n is a terminal symbol α. Rewrite the derivation as

$$\begin{aligned} A &\to \epsilon BC \\ B &\to w_1 \ldots w_{n-1} \\ C &\to \alpha. \end{aligned}$$

In both these cases, the derivations from A and C are put in Greibach normal form, and we can repeat the procedure on the shorter word derived

from B to produce a set of derivations in Greibach normal form equivalent to the one given. □

Lemma 4.2 *The language generated by a context-free grammar C can also be generated exactly by a context-free grammar C' in Chomsky normal form.*

PROOF: Consider a derivation $A \to w$, with $w = w_1 \dots w_n \in (\Sigma \cup N)^\star$. Suppose that w_n is a terminal symbol. Rewrite the derivation as

$$\begin{aligned} A &\to BC \\ B &\to w_1 \dots w_{n-1} \\ C &\to w_n. \end{aligned}$$

Now suppose instead that w_1 is a terminal symbol. Rewrite the derivation as

$$\begin{aligned} A &\to CB \\ B &\to w_2 \dots w_n \\ C &\to w_1. \end{aligned}$$

Suppose that $w = N_1 \dots N_n$, with $N_i \in N$ for $i = 1 \dots n$. Rewrite the derivation as

$$\begin{aligned} A &\to N_1 A_2 \\ A_2 &\to N_2 A_3 \\ &\dots \\ A_i &\to N_i A_{i+1} \\ A_n &= N_n. \end{aligned}$$

Finally, suppose that $w = Xw'Y$, where X and Y are (possibly empty) strings of nonterminal symbols, and $w' \in (\Sigma \cup N)^\star$. Rewrite the derivation as

$$\begin{aligned} A &\to BC \\ B &\to X \\ C &\to DE \\ D &\to w' \\ E &\to Y. \end{aligned}$$

With $B \to X$ and $E \to Y$ rewritten as in the previous case.

By repeating application of these four procedures, we can produce a set of derivations in Chomsky normal form equivalent to the one given. □

A *regular grammar* is one in which all the left-hand sides of production rules are single nonterminal symbols, like before, but with the additional constraint on the right-hand sides that they must be either a single terminal symbol, a single terminal symbol followed by a nonterminal symbol, or the empty string. We say a language is regular if it is generated by a regular grammar.

A language L is regular if and only if it is accepted by some FSA. It has been shown that a language L is context-free if and only if L is accepted by some PDA. For conciseness, we call the language accepted by an automaton its language. So the language of a FSA is regular, and the language of a PDA is context-free.

We can consider the word problem of a group (in the sense meaning the set of words equivalent to the group's identity) to be a language. Now if we know the grammar generating the language has certain properties, we can show the group has certain properties, and vice versa.

Lemma 4.3 *A group has regular word problem if and only if it is finite. [12, 1]*

Proof: We have already proved in Lemma 3.1 that the word problem of a finite group can be solved on a finite state automaton. The language of accepted words of a FSA is regular, so a finite group has regular word problem. We now need to prove that a group with regular word problem is finite.

Suppose we have a finitely generated *infinite* group $G = \langle X ; R \rangle$. For any integer k, there must be infinitely many elements of G which can not be represented by a word shorter than k, since if there were finitely many, and since X is finite, there would be finitely many elements of G. Let M be a FSA with input alphabet $X^{\pm}$ and n states. Let w be a word with length greater than n and such that no subword of w represents the identity element, that is, w can not be made any shorter. Since there are more letters in w than there are states in M, there must be two initial segments u and uv of w such that M is in the same state after reading either of them. Now $uu^{-1} = 1$ in G, but $uvu^{-1} \neq 1$ in G because uvu^{-1} is a conjugate of v, which is not equivalent to the identity by the definition of w. Since M must either accept or reject both uu^{-1} and uvu^{-1} together, the set of words accepted by M is not exactly equal to the word problem of G. □

Clearly, since for a single group there can be several equivalent but distinct

presentations, the language representing the group's word problem depends on which presentation is used. Thankfully, the following lemma means we don't need to worry too much about which particular presentation is used.

Lemma 4.4 *Suppose we have a group G with a finitely generated subgroup H. If $WP(G)$ is a context-free language in one finitely generated presentation of G, then $WP(H)$ is context-free. Hence, $WP(G)$ is a context-free language in every finitely generated presentation of G.[12]*

PROOF: Let $\langle X\,;\,R\rangle$ be a finitely generated presentation of G with context-free word problem. Let M be a PDA accepting the word problem of the latter presentation of G. Let $\langle Y\,;\,S\rangle$, with $Y = y_1, \ldots, y_n$, be a finitely generated presentation of a subgroup H of G. Then make an embedding $\phi(y_i) = u_i$, $i = 1, \ldots, n$, where each u_i is a word on $X^{\pm}$. Clearly a word w on $Y^{\pm}$ only represents the identity element if $\phi(w)$ represents the identity element in $\langle X\,;\,R\rangle$. So we just need to create a PDA M' for $WP(H)$ which on reading y_i simulates what M would do on reading u_i. In the case that $H = G$ with a different presentation, $WP(H)$ is context-free, so G is context-free no matter how it is presented. □

There is a more general result

Knowing that a group is context-free tells us something about the structure of the words in its word problem. We can represent the construction of words on a graph specific to the group's presentation, called its *Cayley graph*, and consider the graph's geometric properties in order to know more about the group.

5 The Cayley graph of a presentation group, and graph triangulations

If $G = \langle X\,;\,R\rangle$ is a finitely generated group, the *Cayley graph* $\Gamma(G)$ of the presentation is defined like so:

For each element of G there is a vertex in the graph. The origin of the graph is the vertex representing the identity element. For each element $g \in G$ and each generator $x \in X^{\pm}$, there is a directed edge linking the vertex representing g to that representing gx, labelled by x.

From the definition of $\Gamma(G)$, a word $\alpha = x_1 x_2 \ldots x_n$ is equivalent to the identity if and only if it labels a closed path in the Cayley graph. So if

we can find the closed paths in the Cayley graph, we can find the words in $WP(G)$.

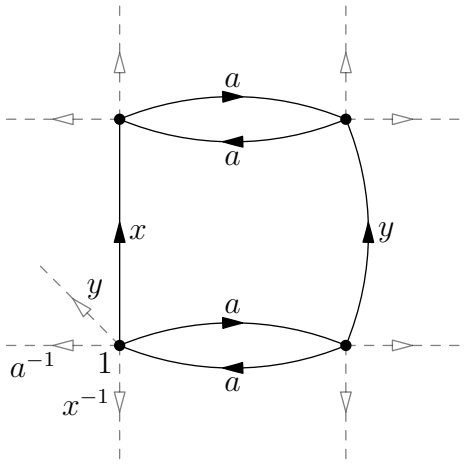


Figure 3: The Cayley graph of the group $G = \langle x, y, a \,;\, xa = ay, a^2 = 1 \rangle$

A polygon P is a set of vertices $\{p_0, \ldots, p_n\}$ with a boundary which is a simple closed curve, that is a series of edges linking p_0 to p_1, p_1 to p_2 and so on, and also p_n back to p_0. A *triangulation* of a polygon P is an addition of edges and possibly vertices so that every polygon contained within P can be broken down into triangles. A *diagonal triangulation* of P is a triangulation which uses only the vertices originally in P. For ease of mind, polygons of just one or two points are also considered to be diagonally triangulated.

Let $G = \langle X \,;\, R \rangle$ be a finitely generated group. Let α be a closed path in $\Gamma(G)$, labelled by $w = y_1 y_2 \ldots y_n$. Write the letters of w clockwise around the boundary of a regular n-gon P, so that the edges of P are labelled by the letters of w. A *K-triangulation of α* is a diagonal triangulation of P with a label from the free group $F = \langle X \rangle$ assigned to each new edge such that reading around the boundary of each triangle gives a true relation in G, and if u is the label on an edge of the triangulation, then $|u| \leq K$.

Lemma 5.1 *Let C be a reduced context-free grammar which generates $WP(G)$ for $G = \langle X \,;\, R \rangle$. If $u, v \in L(A)$ for some variable A of C, then u and v represent the same element of G. [12]*

Proof: Since C is a reduced grammar, A is useful, so there is a derivation $S \stackrel{\star}{\Rightarrow} \alpha A \beta \stackrel{\star}{\Rightarrow} w_1 w_2 w_3$, where $w_1 w_2 w_3 \in WP(G)$, i.e. $w_1 w_2 w_3 = 1$ and w_2 is the part derived from A, i.e. $A \stackrel{\star}{\Rightarrow} w_2$. If we replace the derivation of w_2 by the derivations of u and v, we can get $S \stackrel{\star}{\Rightarrow} w_1 u w_3$ and $S \stackrel{\star}{\Rightarrow} w_1 v w_3$, which

implies $w_1uw_3 = 1 = w_1vw_3$ in G, since C generates the word problem of G. Now since G is a group, we have $u = v$. □

From this result, a theorem about the closed paths in the Cayley graph can be proved.

Theorem 5.2 *If a finitely generated group $G = \langle X ; R \rangle$ is context-free, there exists a constant K such that every closed path in the Cayley graph $\Gamma(G)$ can be K-triangulated. [12]*

PROOF: Let G be context-free, and let C be a reduced grammar in Chomsky normal form which generates $WP(G)$.

Let α be a closed path in $\Gamma(G)$, labelled by $w = y_1, \ldots, y_n$, enclosing an n-gon P. We will create a diagonal triangulation of P by adding edges labelled by words (not letters) in $X^{\pm}$ joining vertices of P so that the label of the boundaries of any resulting polygons are true relations in G. The result is trivial if $n = 3$, so assume $n \geq 4$. If A is a variable of C, let u_A denote the shortest word derivable from A. Consider any derivation $S \stackrel{\star}{\Rightarrow} w$. Since C is in Chomsky normal form, this derivation must be of the form $S \Rightarrow AB \stackrel{\star}{\Rightarrow} w_1w_2$, where $A \stackrel{\star}{\Rightarrow} w_1$ and $B \stackrel{\star}{\Rightarrow} w_2$.

Suppose w_1 and w_2 both have length at least 2. Construct within P an edge labelled by u_B going from the vertex where w_1 ends to the vertex where w_1 begins. By Lemma 5.1 $u_B =_G w_2$. $w_1w_2 =_G 1$, so $u_B =_G w_1^{-1}$. We have now broken P down into two polygons with fewer than n sides, one labelled w_1u_B, the other labelled $u_b^{-1}w_2$.

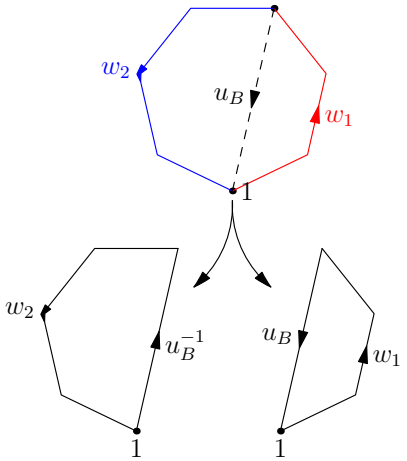


Figure 4: Addition of an edge u_B to triangulate a closed path w_1w_2

Next suppose instead that one of w_1 or w_2 is a terminal symbol a. Say it's w_1. So $S \stackrel{\star}{\Rightarrow} aw_2$ and $|w_2| \geq 3$. Then there is a derivation $S \stackrel{\star}{\Rightarrow} aB \Rightarrow aCD \stackrel{\star}{\Rightarrow} aw_{21}w_{22}$ where at least one of w_{21} or w_{22}, say w_{22}, has length ≥ 2. Add an edge to P labelled by u_D from the vertex where w_{22} begins to the vertex where w_{22} ends. So now we have split P into two polygons, one labelled by $aw_{21}u_D$ and the other labelled by $u_D^{-1}w_{22}$.

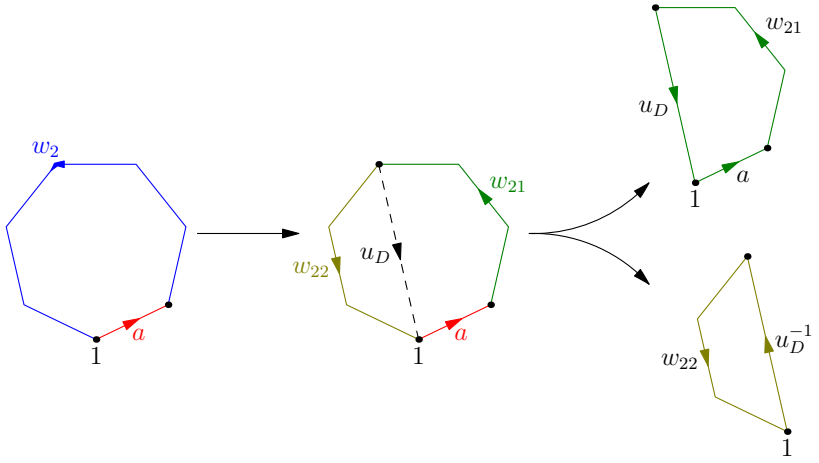


Figure 5: Addition of an edge u_B to triangulate a closed path $aw_2 = aw_{21}w_{22}$

Iterating this procedure on each of the resulting smaller polygons will yield a diagonal triangulation of P. Each edge of the triangulation is labelled by a u_A for some variable A from C, which has finitely many variables. So let $K = max_{A \in C}|u_A|$ and we have a K-triangulation of P. □

Next we will use the *theory of ends* to prove some results about the structure of $\Gamma(G)$, and hence about the construction of G.

6 The Theory of Ends

Let Γ be a graph with origin vertex v_0. Define $\Gamma^{(n)}$ to be the set of points of Γ connected to v_0 by a path of length no more than n.

$\Gamma \backslash \Gamma^{(n)}$ is Γ with all the points no more than n edges away from the origin removed. This may break the graph into several separate components. An *end* is, roughly speaking, a component that is left as n tends to infinity. For example, the infinite cyclic group has two ends, and the free group of rank 2 has infinitely many ends.

Formally define the *number of ends of* Γ, $e(\Gamma)$, by

$$e(\Gamma) = \lim_{n\to\infty} (\text{the number of infinite components of } \Gamma\backslash\Gamma^{(n)}).$$

If $G = \langle X\,;\,R\rangle$ is a finitely generated group, we say that $e(G) = e(\Gamma(G))$ for brevity.

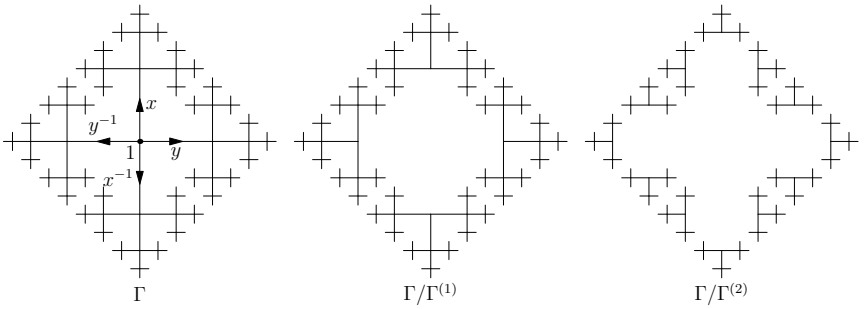


Figure 6: The Cayley graph $\Gamma(G)$ of the free group of order 2, $G = \langle x, y\,;\,\emptyset\rangle$, and with the components $\Gamma^{(1)}$ and $\Gamma^{(2)}$ removed.

Clearly the number of ends of a graph depends on how connected its vertices are, so we shall prove a small result stating that, given a diagonal triangulation of a polygon P whose boundary is divided into three consecutive arcs, there is a triangle with at least one vertex on each arc.

Lemma 6.1 *Let T be a diagonal triangulation of a polygon P with at least 3 edges. If the boundary edges of P are divided into three consecutive nonempty arcs, then some triangle has vertices on all three arcs. Furthermore, this triangle is unique. [12]*

PROOF: Denote the boundary of P by δP. A triangle is said to be on δP if one or more of its edges are in δP. A triangle is called *critical* if it has two edges on δP. Note that if T has more than one triangle, there are at least two critical triangles. If all of the triangles in T with edges in δP are critical, the result holds. So assume there is at least one triangle t with only one edge e in δP and write $\delta P = \eta_1 e \eta_2$, $t = e e_1 e_2$. Then T gives a triangulation T_1 of the polygon P_1 bounded by $\eta_1 e e_1$ and T_2 of P_2 bounded by $\eta_2 e_2 e$. Both T_1 and T_2 have more than one triangle and thus have at least two critical triangles, one of which may be t. The result follows by induction.

Now we will prove that the triangle is unique, by induction on the number k of triangles in T and by 'colouring' the vertices of P. Assign a colour to the vertices lying on each arc, so that three colours are used and the vertices of each colour occur consecutively. If $k = 1$, the result is clearly true. If

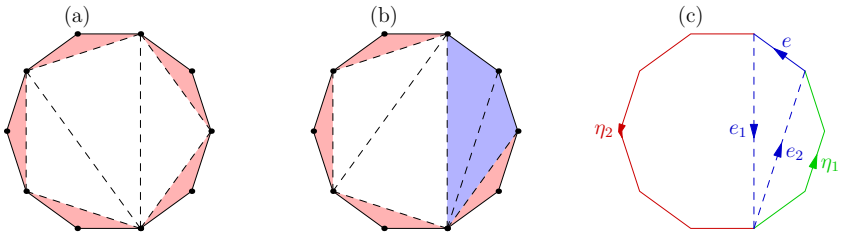


Figure 7: A polygon P (a) triangulated so every triangle on δP is critical, (b) with two triangles on δP not critical, (c) with one non-critical triangle selected to break P into the smaller polygons $\eta_1 e e_1$ and $\eta_2 e_2 e$.

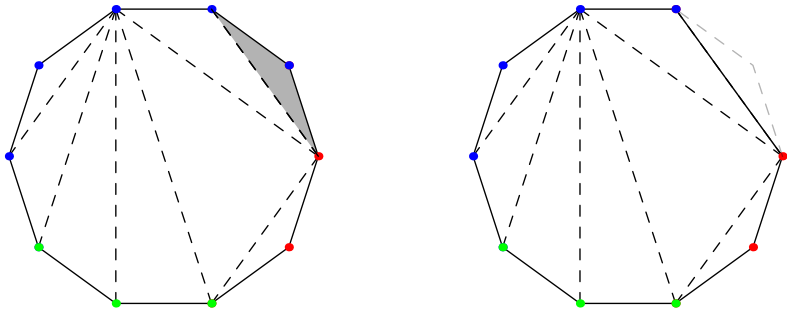


Figure 8: A critical triangle t in the triangulation of a polygon P has vertices of only two colours. The edges of t on δP are removed to create a smaller polygon P' with fewer triangles.

$k > 1$, let t be a critical triangle. If t has a vertex of each colour, then it is the desired triangle and is unique, since if there was another triangle it would have to intersect t. If t has vertices of only one or two colours, create another polygon P' by deleting the edges of t which are on δP. Then P' still has vertices of three colours (since each arc was on at least two vertices) and its triangulation has fewer than k triangles so the result follows. □

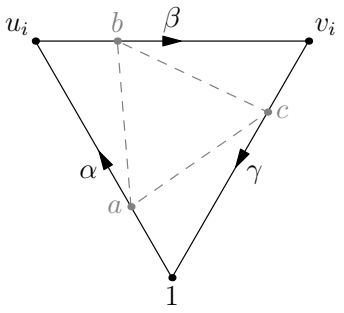
Now we can prove that context-free groups have more than one end, with the aim of eventually showing this implies the group is constructed in a certain way.

Lemma 6.2 *If G is an infinite context-free group, G has more than one end. [12]*

Proof: Let $\Gamma = \Gamma(G)$. If u, v are vertices in Γ then define $d(u, v)$ to be the length of the shortest path between u and v. Choose the identity element 1 of G to be the origin of Γ. G is infinite so there are arbitrarily long words

$y_1 \dots y_j$ such that the shortest word equivalent to them is of length j. We can translate the mid point of the path with label $y_1 \dots y_j$ to the origin of Γ. This means that for any i, $\exists u_i, v_i$ such that $d(u_i, 1) = d(v_i, 1) = i$ and $d(u_i, v_i) = 2i$.

G is context-free so $\exists K$ such that every closed path in Γ can be K-triangulated. Pick $n > \frac{3}{2}K$. Is it true that if $i \geq n$ then u_i and v_i are in different components of $\Gamma \backslash \Gamma^{(n)}$? We will prove this by contradiction. Suppose that u_i and v_i are in the same component of $\Gamma \backslash \Gamma^{(n)}$. Let α be a path of minimal length from 1 to u_i. Let γ be a path of minimal length from v_i to 1. Let β be a path in $\Gamma \backslash \Gamma^{(n)}$ from u_i to v_i. By Lemma 6.1 some triangle t has vertices a, b, c on α, β, γ respectively.



Each edge of t represents a path of length no greater than K. Since $b \in \Gamma \backslash \Gamma^{(n)}$ we have $d(1, a) \geq n - K$, or we would get a path shorter than n to b by going from 1 to a and then from a to b in a path of no longer than K steps. So we now get that $d(a, u_i) \leq i - n + K$ since we have $d(1, u_i) = i$ and a lies on a minimal path from 1 to u_i. Similarly $d(c, v_i) \leq i - n + K$. But now there is a path $u_i \to a \to c \to v_i$ of length $\leq 2i - 2n + 2K + K = 2i + (3K - 2n)$. But this length is less than $2i$ since $2n > 3K$. This contradicts $d(u_i, v_i) = 2i$, so u_i and v_i are not in the same component of $\Gamma \backslash \Gamma^{(n)}$. □

Now that we know a context-free group has more than one end, we can use Stallings' structure theorem to prove it has a certain construction in terms of *free products*.

7 Torsion-free groups and free products

Let G be a group. If an element g of G has finite order, that is, there is some finite n such that $g^n = 1$, then g is a *torsion element* of G. If the only torsion element in G is the identity, then G is *torsion-free*.

If $G = \langle X \,;\, R \rangle$ and $H = \langle Y \,;\, S \rangle$ are groups with X and Y disjoint sets, then

the *free product* of G and H is defined to be $G \star H = \langle X \cup Y \,;\, R \cup S \rangle$. G and H are called the *factors* of the free product. A free product is *nontrivial* if neither of the factors is the trivial group.

If G is a group, the *rank* $r(G)$ of G is the minimal number of generators of G. A theorem of Grusko [6] states that $r(G \star H) = r(G) + r(H)$.

Let $G = \langle X \,;\, R \rangle$ and $H = \langle Y \,;\, S \rangle$ be groups, with A and B respective subgroups. If we have an isomorphism $\phi : A \to B$, then the *free product of G and H amalgamating A and B* is the group $\langle X \cup Y \,;\, R \cup S \cup \{a = \phi(a), a \in A\} \rangle$.

If $G = \langle X \,;\, R \rangle$ is a group, with A and B subgroups of G, and $\phi : A \to B$ an isomorphism, then the *HNN extension* [9] of G with stable letter t and associated subgroups A and B is the group $\langle X \cup \{t\} \,;\, R \cup \{t^{-1}at = \phi(a), a \in A\} \rangle$.

Stallings' structure theorem [14] says that in the case that a group G is finitely generated and torsion free, G has more than one end if and only if G is the infinite cyclic group or is a nontrivial free product.

Theorem 7.1 *A finitely generated torsion-free group is free if and only if it is context-free. [12]*

PROOF: Let G be a context-free group. If $r(G) = 0$ then G is the trivial group, which is free. If $r(G) = 1$ then G is the infinite cyclic group since G is torsion-free. So suppose $r(G) \geq 2$. G is infinite since it is a nontrivial torsion-free group. By Lemma 6.2, G as more than one end so by Stallings' structure theorem G is a nontrivial free product $G = G_1 \star G_2$. By Grushko's theorem G_1 and G_2 have rank less than $r(G)$. By Lemma 4.4 G_1 and G_2 are context-free. Hence, G_1 and G_2 are free by the induction hypothesis. Since the product of free groups is free, G is free. □

Stallings' structure theorem can be stated more generally.

Theorem 7.2 *(Stallings' structure theorem) A finitely generated group G has more than one end if and only if G is eiter a nontrivial free product with amalgamation or an HNN extension, where the amalgamated or associated subgroups are finite. [14]*

If G is a group, we say that $G \supseteq G_0 \supseteq G_1 \supseteq \ldots \supseteq G_n$ is an *accessible series* for G if each G_i is a free product with amalgamation or an HNN extension of G_{i+1} and the amalgamated or associated subgroups are finite.

G is *accessible* if there is an upper bound on the lengths of the accessible series for G, called s, the *accessibility length* of G. Finite groups have accessibility length zero since free products with amalgamation or HNN extensions are infinite. It follows from Grushko's theorem that torsion free groups are accessible. In fact it has been proven by Dunwoody [2] that all finitely presented groups are accessible.

Now we can finally prove our theorem! If the word problem of a group can be solved on a pushdown automaton, then it is context free. So we need to show that if a finitely generated group is context free, then it is virtually free.

Theorem 7.3 *Let G be a finitely generated context-free, accessible group. Then G is virtually free.*

PROOF: We shall prove the result by induction on the accessibility length s of G. If $s = 0$, G is finite and the result holds. So suppose that $G = \langle G_1 \star G_1' \,;\, F = \phi(F)\rangle$ or $G = \langle G_1, t \,;\, tFt^{-1} = \phi(F)\rangle$ with F a finite group. G_1 and G_1' each have accessibility length at most $s-1$. By Lemma 4.4, G_1 and G_1' are context-free since they are subgroups of G, so they are virtually free by the induction hypothesis.

Now we use results from Gregorac, Kerrass *et al* [5] which say that the class of finitely generated virtually free groups is closed under free production with amalgamation or HNN extension, where the associated or amalgamated subgroups are finite. Thus G is virtually free. □

8 One-counter automata

In the special case of a push-down automaton $(Q, \Sigma, \Gamma, \delta, i, F)$ with $\Gamma = \{g\}$, the automaton is called a *one-counter automaton* [7, 10] because, since the stack will always be of the form g^n, it is sufficient just to keep track of the stack height, n.

A language accepted by a one-counter automaton is called a *one-counter language*. It can be quite neatly shown that groups with one-counter word problems are *virtually cyclic*, meaning they have a cyclic subgroup of finite index. This was first proved by Herbst [7], but we shall present here the proof of Holt, Owen and Thomas [10].

First, we define the *growth function* $\gamma_G(n)$ of a group $G = \langle X \,;\, R\rangle$ to be the number of elements of G that are represented by words in X^+ of length at most n.

Theorem 8.1 *If the word problem of a finitely generated group G is a one-counter language, then S has a linear growth function. [10]*

PROOF: Let $G = \langle X ; R \rangle$ be a finitely generated group. Let q be the number of states of a one-counter automaton M accepting $WP(G)$.

For each word $w \in X^+$, choose the shortest path $p(w)$ in M that accepts wxw^{-1}, for some letter x. We need to show that there is a constant K such that immediately after reading x in $p(w)$, the stack height $h(w)$ is at most $K|w|$. Then, after reading x in $p(w)$, for words up to length n there are only $(Kn+1)q$ possibilities for the pair $(h(w), t)$, where t is the state of the machine. This pair cannot be the same for two words w_1 and w_2 that represent different group elements because otherwise there would be a path $w_1 x w_2^{-1}$ accepted by the automaton, implying that w_1 and w_2 do in fact represent the same element.

To show that $h(w) \leq K|w|$, first assume that all moves in M change the stack height by at most one. Allow M, without loss of generality, to only make *reading moves*, where it reads from the input and changes state, or a *non-reading move*, where it changes the stack and changes state. So only non-reading moves can change the stack height.

If $h(w) > q(n+1)$ then, at some point reading w and between two reading moves, there must be an occasion when the stack height increases by at least q. Clearly M must repeat states at this point, so there is a loop in $p(w)$ linking the repeated states during which the stack height is increased by r, with $0 < r \leq q$. Similarly, when reading w^{-1}, there must be a gap when the stack height is reduced by some u, $0 < u \leq q$.

If $h(w) > q^3(n+1)$ then we can find gaps containing q^2 loops of this kind, in which the stack height is increased by at most q when reading w and similarly when reading w^{-1} there are gaps containing q^2 loops in which the stack height is decreased by at most q.

Amongst the increasing loops, at least q of them must increase the stack height by the same number $r \leq q$, and there are similarly at least q decreasing loops which decrease the stack height by some $u \leq q$.

If we remove u loops which increase the stack height by r, and r loops which decrease the stack height by u, we make a shorter path which accepts wxw^{-1}, contradicting the minimality of $p(w)$. We can only do this if doing so does not cause the stack to be empty at any stage.

Assume that $h(w) > q^3(n+2)$. Choose the gap between reading-moves in which we remove the increasing loops to be the latest one in which the stack height increases by at least q^3 at some stage during the gap, and remove

loops as late as possible during that gap. Similarly, choose the earliest possible gap to remove decreasing loops. Between the increasing gap and the end of reading w, the stack cannot be empty because then there would be a later gap which increases the stack height by q^3, and a similar argument follows for the decreasing gaps.

Between the beginning of reading w^{-1} and the place where we removed the decreasing loops, the stack height decreases by less than q^3 during each gap, and hence less than $q^3(n+1)$ in total. Since we are not removing all the loops after this gap, the stack height could decrease by some number less than q^3 during the rest of $p(w)$. But since $h(w) > q^3(n+2)$, this means the stack can never be empty. By this contradiction, we get $h(w) \leq q^3(n+2)$, proving the result. □

Now we need the concept of ShortLex ordering. Suppose X is a finite alphabet with a linear order $<_X$ on it. Then if $\alpha, \beta \in X^\star$, $\alpha <_{SL} \beta$ if either:

- $|\alpha| < |\beta|$ or
- $\alpha \equiv a_1 a_2 \dots a_m$, $\beta \equiv b_1 b_2 \dots b_m$ and $\exists k$ with $1 \leq k \leq m$ such that $a_1 = b_1$, $a_2 = b_2$, $\dots$, $a_{k-1} = b_{k-1}$, $a_k <_X b_k$.

Call the (unique) least representative of a group element under $<_{SL}$ the ShortLex *normal form* of that element.

Note that if uvw is in ShortLex normal form, so is v, since if there is a $v' <_{SL} v$, then $uv'w <_{SL} uvw$. This means that any subword of a word in ShortLex normal form is itself in ShortLex normal form. By convention, the empty word is not in ShortLex normal form.

Theorem 8.2 *If G is a group with linear growth function then there exist elements $a_i, b_j, c_k \in G$, with $1 \leq i, j, k \leq N$ for some N, such that any element of G has the form $a_i b_i^n c_i$ for some i and n with $1 \leq i \leq N$ and $n \geq 0$. [10]*

PROOF: Let $G = \langle X ; R \rangle$ be a finitely generated group with linear growth. Choose a linear order on X and then consider the set N of ShortLex normal forms for G.

Let $L \subseteq N$ denote the set of words $w \in N$ such that w is a prefix of infinitely many $v \in N$, i.e. w can be extended indefinitely.

If G is finite then the result is trivial, so assume G is infinite, and hence L is infinite since every element has a distinct ShortLex normal form.

Form a graph Γ with vertex set L and edges from each w to wx for each $x \in X$ such that $wa \in L$. Add an origin vertex to the graph, with edges labelled by x leading from it to the vertex representing the word x, when $x \in L$. This graph is a subgraph of the Cayley graph of G, with the property that there is a unique path from the origin to each vertex, so Γ is a tree.

Let $K(n)$, $n \geq 0$ be the number of words in L of length n, which is equal to the number of vertices of Γ at a distance n from the origin. Since Γ is a tree with no finite branches, every vertex at a distance n from the origin is connected to at least one vertex at a distance $n+1$, so $K(n)$ is an increasing function of n. But G has linear growth function, and $\gamma_G(n) \geq \sum_{i=1}^{n} K(i)$ implies that $K(n)$ is bounded. So $\exists K$ with $K(n) = K$ for all sufficiently large n.

So, once n is large enough that $K(n) = K$, Γ consists of K disjoint paths, or *strands*. Let $w_1, \ldots, w_K$ be prefixes of the strands such that no w_i is a prefix of another w_j. What this means in effect is that each w_i is picked to be on the strand after the last 'branching' vertex which is connected to more than one longer word. Note that every word in $X^\star$ has a prefix one of the w_i.

We will now split up one particular strand, say that whose prefix is w_1, into segments p_{1n}. Let $p_{11} = w_1$. Then we can read further along the strand to $w\alpha$, where $\alpha \in X^\star$ and α has a prefix p_{12} which is one of the w_i. We can continue in this vein to make an infinite sequence $\{p_{1n}\}_1^\infty$ with each p_{1i} equal to some w_k. Clearly p_{1i} must repeat at some point, say $p_{1i} = p_{1j}$, where $i < j$. Then $p_{1(i+k)} = p_{1(j+k)}$ for all $k \geq 0$.

So the whole strand consists of $p_{11}p_{12}\ldots p_{1(i-1)}$ followed by infinitely many repetitions of $y_1 := p_{1i}\ldots p_{1(j-1)}$. To put it another way, for each i, the infinite strand with prefix w_i consists of a prefix followed by infinitely many repetitions of y_i. Let

$$B = \{b_j : b_j \text{is a cyclic permutation of} y_i, 1 \leq i \leq K\}.$$

Then all words in L are of the form $a_i b_j^n$, where $b_j \in B$ and the a_i are finitely many prefixes of the graph.

Now consider the graph Γ' constructed from N in the same way Γ was constructed from L. Γ' consists of Γ with finite branches added to some vertices. If the lengths of these branches are uniformly bounded, then their vertices represent a finite set of elements of c_k.

Suppose the branches are not uniformly bounded. Then there exist arbitrarily long branches. So there exists a prefix u and some $v \in B$, for which there

exist arbitrarily long words w such that, for some m, $uv^m w \in N$ but $uv^m w_1$ is not infinitely extensible, where w_1 is the first letter of w. In particular, w does not have v as a prefix because otherwise we could pick a different w' and consider $uv^{m+1}w'$.

Suppose that the number of elements of G of length at most n is at most Cn, which we can do since G has linear growth. As $|w|$ increases, so must m, so we can choose m and $|w|$ sufficiently large that $m|w| > C(m|v| + |w|)$.

The $m|w|$ subwords of $v^m w$ of the form $v'w(t)$, for $1 \leq i \leq m$, $1 \leq t \leq |w|$, where $w(t)$ denotes the prefix of w of length t, cannot all represent distinct elements. Since they are all in ShortLex normal form, two of them must be the same word, $v^i w(s) \equiv v^j w(t)$. Clearly $i \neq j$, as that would give $w(s) = w(t)$, which implies that $s = t$. Thus w has v as a prefix. This contradiction proves that the added branches are uniformly bounded, so there are finitely many elements c_k and any element of G has the form $a_i b_i^n c_i$. □

This leads quite quickly to the following theorem:

Theorem 8.3 *A finitely generated group G has one-counter word problem if and only if G is virtually cyclic. [7, 10]*

PROOF: Suppose that G has one-counter word problem. By 8.2 there exist elements $a_1, \ldots, a_n, b_1, \ldots, b_n, c_1, \ldots, c_n \in G$ such that

$$G = \cup_{i=1}^{n} \cup_{r=0}^{\infty} a_i b_i^r c_i$$

Given this, we have,

$$G = \cup_{i=1}^{n} (a_i \langle b_i \rangle a_i^{-1}) a_i c_i.$$

So G is a union of finitely many cosets. At least one of the subgroups $a_i \langle b_i \rangle a_i^{-1}$ has finite index in G, so G is virtually cyclic. □

9 A characterisation of finitely generated groups with soluble word problem

Now we will take a step away from the properties of a group's word problem and instead consider what having a soluble word problem says about the structure of a group.

We will show a famous result of Boone and Higman [3], that a finitely-generated group G has soluble word problem if and only if there exist a simple group H and a finitely-presented group K such that G is a subgroup of H and H is a subgroup of K. First, we will need to define some more terms.

A group G is called *simple* if the only normal subgroups it contains are the trivial group and G itself.

A group $G = \langle X ; R \rangle$ is called *recursively enumerable* if there is a recursive function which produces all of the words in $X^\star$. If a group is recursively enumerable we can order its words, and we will refer to the ith word in this ordering as w_i.

Given a group $G = \langle X ; R \rangle$, we form the group $G^\dagger$ by a sequence of HNN extensions. First, we add a generator t and no relations to obtain the free product $G \star \langle t \rangle$. Then, we add generators u_i and relations of the form $u_i^{-1} t u_i = t w_i$ for each of the words w_i in the canonical ordering of $X^\star$. Finally, we add generators v_i and relations of the form $v_i^{-1} t v_i = t^{-1} w_i^{-1} t w_i$, for each w_i. We call the letters t, u_i and v_i the *stable letters* of $G^\dagger$.

We will denote application of the $\dagger$ extension n times on a group by $G^{(n)}$, and let $G^{(0)} = G$. Let $G^{(\infty)} = \bigcup_{n=1}^{\infty} G^{(n)}$. The *rank* of a word $w \in G^{(\infty)}$ is the smallest n such that $w \in G^{(n)}$.

Given a group $G = \langle X ; R \rangle$ and any word w of G, G^w is the group obtained by adding the relation $w = 1$ to G, that is, $\langle X ; R \cup \{w = 1\} \rangle$.

Lemma 9.1 *Let $G^\star$ be an HNN extension of the group G with stable letters p_v. Then $G \subseteq G^\star$. [4]*

Corollary 9.2 *For any group $G = \langle X ; R \rangle$, G is embedded in $G^\dagger$ by the identity map. Furthermore, given non-negative integers m and n such that $m \leq n$, the group $G^{(m)}$ is embedded in the group $G^{(n)}$ via the identity map. [3]*

Proof: It is clear that $G \subseteq G^\dagger$ by the construction of $G^\dagger$ and Lemma 9.1. It is also clear by induction on $n - m$ that $G^{(m)} \subseteq G^{(n)}$ when $m \leq n$. $\square$

Given two decision problems (for example the word problem) P_1 and P_2, P_1 is said to be *uniformly soluble* in P_2 if a machine which solves P_2 can also solve P_1. Our next step is to show that the word problem of $G^\dagger$ is uniformly soluble in the word problem of G, and hence that the word problem of $G^{(\infty)}$ is uniformly soluble in $WP(G)$ as well.

Lemma 9.3 *For any group G, $WP(G^\dagger)$ is uniformly soluble in $WP(G)$. Furthermore, for any $N \geq 0$, $WP(G^{(N)})$ is uniformly soluble in $WP(G)$. [3]*

We omit the proof, which can be found in [3].

Lemma 9.4 *For any group G, suppose $w = 1$ in $G^{(\infty)}$. Then $w = 1$ in $G^{(n)}$ where w is of rank n. [3]*

PROOF: The fact that w is trivial in $G^{(\infty)}$ is a result of finitely many relations involving finitely many generators. Say these generators all fall in some $G^{(m)}$, so that $w = 1$ in $G^{(m)}$. But $n \leq m$ and so by Lemma 9.2, $w = 1$ in $G^{(n)}$. □

Lemma 9.5 *For any group G and any $n \geq 0$, the group $G^{(n)}$ is embedded in $G^{(\infty)}$ via the identity map. [3]*

PROOF: This is clear, by Lemma 9.2 and Lemma 9.4. □

Proposition 9.6 *For any group G, $WP(G^{(\infty)})$ is uniformly soluble in $WP(G)$. [3]*

PROOF: By Lemmas 9.4 and 9.5, for any word w of $G^{(\infty)}$ of rank N, $w = 1$ in $G^{(\infty)}$ if and only if $w = 1$ in $G^{(N)}$. Now by Lemma 9.3, $WP(G^{(N)})$ is uniformly soluble in $WP(G)$, so the result holds. □

Now we will prove that $G^{(\infty)}$ is simple.

Lemma 9.7 *[3] Let $G = \langle X ; R \rangle$ be a group and $w \in X^\star$ be a word in G such that $w \neq_G 1$. Then $\forall w' \in X^\star$, $w' = 1$ in $G^{\dagger w}$.*

PROOF: Note that $w = w_i$ for some i in the enumeration of $X^\star$, and $G^\dagger$ contains a relation $v_i^{-1} t v_i = t^{-1} w_i^{-1} t w_i$. Then since $w_i = w = 1$ in $G^{\dagger w}$, we get that $t = 1$ in $G^{\dagger w}$. Now, for every word $w' \in X^\star$, firstly $w' = w_j$ for some j, and then there is a relation from the presentation of $G^\dagger$ of the form $u_j^{-1} t u_j = t w_j$, which gives us that $w_j = 1$ in $G^{\dagger w}$ and so $w' = 1$ in $G^{\dagger w}$. □

Lemma 9.8 *Let $G = \langle X ; R \rangle$. G is simple if and only if for each $w \in X^\star$ such that $w \neq_G 1$, G^w is the trivial group.*

PROOF: Suppose there is some w such that G^w is not the trivial group, i.e. it contains some nontrivial element n. Then $\{gng^{-1} : g \in G\}$ is a normal subgroup of G, so G is not simple. □

Proposition 9.9 *For any group G, the group $G^{(\infty)}$ is simple. [3]*

PROOF: Suppose w and v are words of $G^{(\infty)}$, and $w \neq 1$ in $G^{(\infty)}$. Let N be the maximum of the ranks of w and v. By Lemma 9.5, $w \neq 1$ in $G^{(N)}$.

By Lemma 9.7, $v = 1$ in $G^{(N+1)w}$. Since each generator and relation of $G^{(N+1)w}$ is among those of $G^{(\infty)w}$, $v = 1$ in $G^{(\infty)w}$. Now $G^{(\infty)w}$ is the trivial group $\forall w$, so by Lemma 9.8 $G^{(\infty)}$ is simple. □

We just need a couple more results from [8] and [9], respectively, and then we will be able to prove our main theorem.

Lemma 9.10 *A finitely generated group can be embedded in a finitely presented group if and only if it is recursively presented. [8]*

Lemma 9.11 *Any recursively generated group G can be embedded in a group H generated by only two elements. [9]*

Theorem 9.12 *If a finitely generated group G has soluble word problem, then there exist a simple group H and a finitely presented group K such that G is a subgroup of H and H is a subgroup of K. [3]*

PROOF: Let $G = \langle X ; R \rangle$ be a finitely generated group with soluble word problem. We let $H = G^{(\infty)}$. Then by Proposition 9.9, H is simple.

But we can also give a presentation of $G^{(\infty)}$ in terms of generators $t_n, u_{i,n}, v_{i,n}$ and relations of the form $w = 1$ for each $w \in WP(G^{(\infty)})$. By Proposition 9.6, this set of relations is a recursive set.

Then, by Lemma 9.11, $G^{(\infty)}$ can be embedded in a group N consisting of two generators and a resursively enumerable set of relations. Finally, by Lemma 9.10, N can be embedded in a finitely presented group K, as required. □

We can also show the converse.

Theorem 9.13 *Let $G = \langle X ; R \rangle$ be a finitely generated group which is contained in a simple group H, which is itself contained in a finitely presented group K. Then G has soluble word problem. [3]*

PROOF: First note that there exists a recursive functional Ω which, when applied to a finite presentation V of a group, yields a recursive enumeration of all the consequent relations of V.

Suppose we have a finitely generated group G, contained in a simple group H which is contained in a finitely presented group K. Note that if H is trivial then so is G and thus $WP(G)$ is soluble. So assume that H is non-trivial.

Suppose we have a word $w \in X^\star$ such that $w =_G 1$. Then if we check the enumeration $\Omega(K)$ of all the consequent relations of K, we will eventually reach the one of the form $w = 1$. But if we are not sure that w is trivial in G, we will not know when to stop looking, so we can not say for certain that a word is *not* trivial in G. We might just need to look for a bit longer and find a relation that says it is trivial. So we need another process, which we can perform in parallel with this one, which will definitely tell us that a non-trivial word is non-trivial in a finite amount of time.

Let w_{H0} be a word of H such that $w_{H0} \neq_H 1$. Let ϕ be the map embedding G in H, and let ψ be the map embedding H in K. $\psi \circ \phi$ is recursive, meaning it can be computed, since its behaviour depends on the finitely many generators of G. We have, for any word w_H of H, with $w_H \neq_H 1$, that

$$w_{H0} \in H = \{w_H\}_H,$$

since H is simple, and that

$$\psi(\{w_H\}_H) \subseteq \{\psi(w_H)\}_K$$

since $H \subseteq K$.

Thus, for any $w \in X^\star$, if $w \neq_G 1$, then $\phi(w) \neq_H 1$ and $\psi(w_{H0}) \in \{\psi(\phi(w))\}_K$, i.e $\psi(w_{H0}) = 1$ in the group $K^{\psi(\phi(w))}$ which is K with the added relation $psi(\phi(w)) = 1$.

Which is to say, if, where $w \in X^\star$, $\phi(w_{H0}) = 1$ in $K^{\psi(\phi(w))}$, then $w \neq_G 1$.

Now, suppose w is any word of G. First, compute $w' = \psi(\phi(w))$. Next, apply Ω to K and $K^{\psi(\phi(w))}$. Now we can check through the relations in $\Omega(K)$ and $\Omega(K^{\psi(\phi(w))})$ until we decide either that $w' = 1$ in the first set, in which case $w =_G 1$, or that $\psi(w_{H0}) = 1$ in the second set, implying that $w \neq_G 1$. □

References

[1] A. V. Anīsīmov. The group languages. *Kibernetika (Kiev)*, (4):18–24, 1971.

[2] C. Bamford and M. J. Dunwoody. On accessible groups. *J. Pure Appl. Algebra*, 7(3):333–346, 1976.

[3] William W. Boone and Graham Higman. An algebraic characterization of groups with soluble word problem. *J. Austral. Math. Soc.*, 18:41–53, 1974. Collection of articles dedicated to the memory of Hanna Neumann, IX.

[4] John L. Britton. The word problem. *Ann. of Math. (2)*, 77:16–32, 1963.

[5] R. Gregorac. On generalized free products of finite extensions of free groups. *J. London Math. Soc.*, 41:662–666, 1966.

[6] I. Grusko. O bazisah svobodnogo proizvedeniya grupp. *Mat. Sbornik*, 1940.

[7] Thomas Herbst. On a subclass of context-free groups. *RAIRO Inform. Théor. Appl.*, 25(3):255–272, 1991.

[8] G. Higman. Subgroups of finitely presented groups. *Proc. Roy. Soc. Ser. A*, 262:455–475, 1961.

[9] Graham Higman, B. H. Neumann, and Hanna Neumann. Embedding theorems for groups. *J. London Math. Soc.*, 24:247–254, 1949.

[10] Owens Matthew D. Holt, Derek F. and Richard M. Thomas. Groups and semigroups with a one-counter word problem.

[11] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to automata theory, languages, and computation.* Addison-Wesley Publishing Co., Reading, Mass., 1979. Addison-Wesley Series in Computer Science.

[12] David E. Muller and Paul E. Schupp. Groups, the theory of ends, and context-free languages. *J. Comput. System Sci.*, 26(3):295–310, 1983.

[13] Otto Schreier. Über die Erweiterung von Gruppen I. *Monatsh. Math. Phys.*, 34(1):165–180, 1926.

[14] John R. Stallings. On torsion-free groups with infinitely many ends. *Ann. of Math. (2)*, 88:312–334, 1968.