THE UNIVERSITY OF CAMBRIDGE.
PART III (*NON-EXAMINABLE*):
DECISION PROBLEMS IN GROUP THEORY.
24 LECTURES, LENT 2016–17

Prepared by Maurice Chiodo

Contents

© Maurice Chiodo, March 6, 2017.

This is a collection of useful references for the course, most of which will have been cited at some point in these notes. They include text books, papers, preprints, and other printed course notes.

As these notes will be released in parts, the list of references will grow dynamically. Thus, they will not be listed alphabetically, but instead in the order in which I add them to the text.

References

[1] R. I. Soare, *Recursively enumerable sets and degrees: a study of computable functions and computably generated sets*, Springer (1987).

[2] F. Stephan, *Recursion Theory*, Online course notes from 2012. See http://www.comp.nus.edu.sg/∼fstephan/recursiontheory-pstopdf.pdf

[3] A. Turing, *On computable numbers, with an application to the entscheidungsproblem*, Proc. London Math. Soc. (2) **42**, 230–265 (1937).

[4] H. Rogers Jr, *Theory of recursive functions and effective computability*, MIT Press (1987).

[5] P.T. Johnstone, *Notes on logic and set theory*, Cambridge University Pres (1987).

[6] A. M. Slobodskoĭ, *Unsolvability of the universal theory of finite groups* (Russian), Algebra i Logika **20** no. 2, 207–230 (1981). English transl., Algebra and Logic, 139–156 (1981).

[7] A. Mal'cev, *Algorithms and recursive function*, Wolters-Noordhoff, Groningen (1970).

[8] J. button, M. Chiodo, *Infinite groups and decision problems*, Online course notes from 2015-16. See https://www.dpmms.cam.ac.uk/∼mcc56/teaching/2015-16

[9] M. Chiodo, *Finding non-trivial elements and splittings in groups*, J. Algebra. **331**, 271–284 (2011).

[10] M. Chiodo, *The subgroup identification problem for finitely presented groups*, Internat. J. Algebra Comput. **23** no. 3, 673–684 (2013).

[11] C. F. Miller III, *Decision problems for groups-survey and reflections*. Algorithms and classification in combinatorial group theory (Berkeley, CA, 1989), Math. Sci. Res. Inst. Publ., **23**, Springer, New York, 1–59 (1992).

[12] J. Rotman, *An introduction to the theory of groups; fourth edition*, Springer-Verlag, New York, (1995).

[13] D. Cohen, *Combinatorial group theory: A topological approach (London Mathematical Society Student Texts)*, Cambridge University Press, Cambridge, (1989).

[14] V. Borisov, *Simple examples of groups with unsolvable word problem*, Math. Zametki **6**, 521–532 (1969); English transl., Math. Notes **6**, 768–775 (1969).

[15] G. Higman, *Subgroups of finitely presented groups*, Proc. Royal Soc. London Ser. A **262**, 455–475 (1961).

[16] S. Aanderaa, D. Cohen, *Modular machines, the word problem for finitely presented groups and Collins' theorem*. Word problems, II (Conf. on Decision Problems in Algebra, Oxford, 1976), pp. 1–16, Stud. Logic Foundations Math., 95, North-Holland, Amsterdam-New York, 1980.

[17] S. Aanderaa, D. Cohen, *Modular machines and the Higman-Clapham-Valiev embedding theorem*. Word problems, II (Conf. on Decision Problems in Algebra, Oxford, 1976), pp. 7–28, Stud. Logic Foundations Math., 95, North-Holland, Amsterdam-New York, 1980.

[18] S. I. Adian, *Finitely presented groups and algorithms*, Dokl. Akad. Nauk SSSR **117**, 9–12 (1957).

[19] M. O. Rabin, *Recursive unsolvability of group theoretic problems*, Annals of Math. **67**, 172–194 (1958).

[20] G. Higman, B. Neumann, H.Neumann, *Embedding Theorems for Groups*, J. London Math. Soc. **24** No. 4, 247–254 (1949).

[21] A. Houcine, *Satisfaction of existential theories in finitely presented groups and some embedding theorems*, Annals of Pure and Applied Logic **142**, 351–365 (2006).

[22] C. F. Miller III, *The word problem in quotients of a group* in *Aspects of Effective Algebra*, ed. J.N. Crossley, Upside Down A Book Company, Steel's Creek, 246–250 (1981).

[23] K. A. Mihailova, *The occurrence problem for direct products of groups*, Dokl. Akad. Nauk SSSR **119**, 1103–1105 (1958).

[24] W.W. Boone, D.J. Collins, *Embeddings into groups with only a few defining relations*, J. Austral. Math. Soc. **18**, 1–7 (1974).
[25] D.J. Collins, *On a group embedding theorem of V. V. Borisov*, Bull. London Math. Soc. **4**, 145–147 (1972).
[26] M. Chiodo, M. Hill, *Preserving torsion orders when embedding into groups with 'small' finite presentations*, arXiv:1610.00977v1 (2016).
[27] W. W. Boone, G. Higman, *An algebraic characterization of the solvability of the word problem*, J. Austral. Math. Soc. **18**, 41–53 (1974).
[28] R. Thompson, *Embeddings into finitely generated simple groups which preserve the word problem*. Word problems, II (Conf. on Decision Problems in Algebra, Oxford, 1976), pp. 401–441, Stud. Logic Foundations Math., 95, North-Holland, Amsterdam-New York, 1980.
[29] W. W. Boone, H. Rogers Jr., *On a problem of J.H.C. Whitehead and a problem of Alonzo Church*, Math. Scand. **19**, 185–192 (1966).
[30] M. Chiodo, *On torsion in finitely presented groups*, Groups Complex. Cryptol. **6**, No. 1, 1–8 (2014).
[31] M. Bridson, H. Wilton, *The triviality problem for profinite completions*, Invent. Math. **202** no. 2, 839–874 (2015).
[32] Z. Sela, *Diophantine geometry over groups. VI. The elementary theory of a free group*, Geom. Funct. Anal. **16** no. 3, 707–730 (2006).
[33] O. Kharlampovich, A. Myasnikov, *Elementary theory of free non-abelian groups*, J. Algebra **302** no. 2 , 451–552 (2006).
[34] M. Bestvina, M. Feighn, *A combination theorem for negatively curved groups*, J. Differential Geom. **35** no. 1, 85–101 (1992).
[35] D. Wise, *The residual finiteness of negatively curved polygons of finite groups*, Invent. Math. **149** no. 3, 579–617 (2002).
[36] M. Burger, S. Mozes, *Finitely presented simple groups and products of trees*, C. R. Acad. Sci. Paris Sér. I Math. **324** no. 7, 747–752 (1997).

0. INTRODUCTION

These notes have been prepared for students/researchers attending the 24-lecture Part III (NON-EXAMINABLE) course *Decision Problems in Group Theory* in Lent term 2016–17. It is highly unlikely that any material will be covered that is *not* contained in these notes; it is even more unlikely that the entirely of these notes will be covered. Everything you need should be contained here.

I have also made available the notes from the Part III course I gave (jointly with Jack Button) last year on Infinite Groups and Decision Problems. Those notes contain all the necessary background in the theory of infinite groups. It is *highly* recommended that you read those, and/or attend the first 4 lectures of the Part III course on Geometric Group Theory this term (lectured by Mark Hagen).

For those of you who are reading this in digital format, you can click on items in the table of contents to go directly to that part in the text. Also, when you click on a definition number, you will go directly to the statement. The same applies for lemmata, theorems, etc.

For those of you reading this in printed format, yet yearning to enter the digital age, the pdf notes can be found by following the teaching link from my homepage at

`https://www.dpmms.cam.ac.uk/~mcc56/`

I have now set a Part III essay on a topic related to this course, titled *SQ-universality and the word problem in groups*, which has been approved by the board of examiners.

0.1. **Shorthand conventions.**

Throughout the lectures I will make use of certain shorthand conventions on the blackboard. Some of these are standard, others are not. For clarity, here is a list of those which I will be using.

b/c = because
c/f = comes from
w/ = with
w/o = without
wts = want to show
wlog = without loss of generality
thm = theorem
defn = definition
lem = lemma
cor = corollary
pf = proof
eg = example
ex = exercise
TM = Turing machine
PC = partial computable
PR = partial recursive
Prim R = primitive recursive
s.t. = such that
rec = recursive
iff = if and only if

1. COMPUTABILITY THEORY

Most of what we do in this section is covered in part A of the book [1] by Soare. Reading this, or at least parts of it, would certainly do you no harm. Another suitable, cheaper, and more easily obtainable option would be the online course notes by Stephan [2].

1.1. **Turing machines.**

To explore what is *incomputable*, we first need a robust definition of what it means for a problem to be *computable*. There are many (equivalent) ways to do this; we present one of them here, first introduced by Turing in 1937 [3]. We begin with some notation about words.

Definition 1.1.
Let X (say $\{a, b, \ldots\}$) be a set of symbols (which we will call letters). A *word* on X is a finite sequence of elements of X. This includes the empty word ϵ. We write X^+ for the set of *all* words on X, so formally $X^+ = \cup_{n=0}^{\infty} X^n$ where n is the word length.
Now let X^{-1} ($= \{a^{-1}, b^{-1}, \ldots\}$ say) be a set with the same cardinality as but disjoint from X, along with a particular bijection $\iota : X \to X^{-1}$ given by $\iota(a) = a^{-1}$, etc. We think of X^{-1} as *formal inverses* for the elements of X. Also let $X^* = (X \cup X^{-1})^+$, so that here our letters are either elements of X or their formal inverses.
We define $X^{\text{red}} \subseteq X^*$ to be the set of *reduced* words on $X \cup X^{-1}$, that is the words which contain no subword xx^{-1} or $x^{-1}x$ for $x \in X$. Later on this could be called *freely reduced* once other notions of reduced are given.

Definition 1.2.
A *Turing machine* (abbreviated to TM) is a finite object which consists of the following:
1. A finite *alphabet* $S = \{s_0, \ldots, s_m\}$.
2. A finite set of *states* $Q = \{q_0, \ldots, q_n\}$ (we distinguish q_1 as the *initial state* and q_0 as the *halting state*.
3. Two formal symbols L, R, different from any symbols in S or Q.
4. A finite set of *instructions*, which are quadruples in $Q \times S \times (S \cup \{L, R\}) \times Q$, one for each ordered pair $(q_i, s_j) \in Q \times S$, each of one of the following three forms:
   a) (q_i, s_j, s_k, q_l)
   b) (q_i, s_j, L, q_l)
   c) (q_i, s_j, R, q_l)
such that each starting pair $(q_i, s_j, *, *)$ occurs precisely once.
5. An *eye* which can read one alphabet symbol at a time.
6. A variable internal state q which takes values in Q.

Definition 1.3.
We define the *action* of a Turing machine on words in its alphabet by the following:
a) Take a word $w = s_{i_1} \ldots s_{i_k} \in S^+$.
b) Write w on a tape, with one symbol per box. That is, $\overline{|s_{i_1}|s_{i_2}|\ldots|s_{i_k}|}$.
c) Place the eye over the leftmost square of the tape.
d) Set the internal state to $q = q_1$.
e) Look for the quadruple $c = (q, s_j, *, *)$ (where s_j is the symbol under the

eye), and implement c according to the following convention:

i) A quadruple of type (q_i, s_j, s_k, q_l) means the machine replaces the read symbol s_j with the symbol s_k, and changes its internal state to $q = q_l$.

ii) A quadruple of type (q_i, s_j, L, q_l) means the eye moves one square to the left, and the machine changes its internal state to $q = q_l$.

iii) A quadruple of type (q_i, s_j, R, q_l) means the eye moves one square to the right, and the machine changes its internal state to $q = q_l$.

Note: If the eye ever moves past the end of the tape, additional squares with symbol s_0 are added as needed, hence the tape is always finite (for this reason we often interpret s_0 as a blank square).

f) If the internal state q is not q_0, then the machine repeats the process from step e) , with the new internal state and the eye at the new position.

g) If the internal state q is q_0, then the machine halts and outputs the (now modified) tape.

To get an algebraically clearer description of the inner working of a Turing machine, we introduce the notion of an *instantaneous description*.

Definition 1.4.
Given a Turing machine T with alphabet $S = \{s_0, \ldots, s_m\}$ and states $Q = \{q_0, \ldots, q_n\}$, we call a word $w \in (S \cup Q)^+$ an *instantaneous description* of T if w takes the form

$$w = s_{i_1} \ldots s_{i_k} q_j s_{i_{k+1}} \ldots s_{i_l}$$

where we may have $k = 0$ (i.e., q_j can be the leftmost letter of w), but we insist that $l > k$ (i.e., q_j cannot be the rightmost letter of w).

The instantaneous description $s_{i_1} \ldots s_{i_k} q_j s_{i_{k+1}} \ldots s_{i_l}$ simply corresponds to "The Turing machine T, in state q_j, reading letter $k+1$ of the l-letter word $s_{i_1} \ldots s_{i_l}$". We then see that we can act on this instantaneous description by choosing the appropriate quadruple $(q_j, s_{i_{k+1}}, *, *)$ from T, to get a new instantaneous description.

Definition 1.5.
If, on input of a word w, the Turing machine T eventually halts in finitely many steps (that is, eventually reaches its halting state q_0), then we say $T(w)$ *halts*, and write $T(w) \downarrow$. We write $T(w) \downarrow = v$ (with $v \in S^+$) to denote that $T(w)$ halts with v written on the tape when it does. If T never reaches its halting state on input w, then we say $T(w)$ *does not halt*, and write $T(w) \uparrow$.

Definition 1.6.
The *halting set* of a Turing machine T with alphabet S, denoted $\Omega(T)$, is the set of all words $w \in S^+$ on which $T(w)$ halts. That is,

$$\Omega(T) := \{w \in S^+ \mid T(w) \downarrow\}$$

Given a finite alphabet S, a set $A \subseteq S^+$ is said to be *recursively enumerable*, abbreviated r.e. (or *computably enumerable*, abbreviated c.e.) if $A = \Omega(T)$ for some Turing machine on alphabet S. $A \subseteq S^+$ is said to be *recursive* (or *computable*) if both A and $S^+ \setminus A$ are recursively enumerable.

1.2. **Partial computable and partial recursive functions.**

Definition 1.7.
Let T be a Turing machine on alphabet $S = \{s_0, \ldots, s_{m-1}\}$. For each integer n, we can consider a base-m expansion of n (with s_0 corresponding to digit

$0, \ldots, s_{m-1}$ corresponding to digit $m-1$). Thus we can consider the inputs and outputs of T as being integers. Thus any Turing machine T on (ordered) alphabet S defines a partial function $f : \mathbb{N} \to \mathbb{N}$. We write $T(x) \downarrow = y$ for $x, y \in \mathbb{N}$ to mean that T, on input of the tape corresponding to x, halts with output the tape corresponding to y.

Definition 1.8.
Define *Cantor's pairing function* $\langle ., . \rangle : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ by

$$\langle x, y \rangle = \frac{1}{2}(x+y)(x+y+1) + y$$

Then this is a bijection from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$, which we can extend inductively to define $\langle x_1, \ldots, x_n \rangle := \langle \langle x_1, \ldots, x_{n-1} \rangle, x_n \rangle : \mathbb{N}^n \to \mathbb{N}$.

Proof. Let $x + y = z$. Then we need only make the following elementary observations:
1. $\langle z, 0 \rangle \leq \langle x, y \rangle \leq \langle 0, z \rangle$
2. $\langle x, y+1 \rangle = \langle x+1, y \rangle + 1$
3. $\langle 0, z \rangle + 1 = \langle z+1, 0 \rangle$

The above imply that a totally ordered listing of $\mathbb{N}$ is given by
$\langle 0,0 \rangle, \langle 1,0 \rangle, \langle 0,1 \rangle, \langle 2,0 \rangle, \langle 1,1 \rangle, \langle 0,2 \rangle, \ldots$ □

Most 'well-defined' problems in mathematics can be reduced to computing membership in a subset of $\mathbb{N}$, via use of Cantor's pairing function. For example, take the set $X := \{\langle x, y \rangle \in \mathbb{N} \mid x \text{ divides } y\}$. Then the problem 'does x divide y?' can be interpreted as 'is $\langle x, y \rangle$ in X?'

Definition 1.9 (Partial computable functions)**.**
A partial function $f : \mathbb{N}^k \to \mathbb{N}$ is said to be *partial computable* by a Turing machine T if, for all $(m_1, \ldots, m_k) \in \mathbb{N}^k$ where $f(m_1, \ldots, m_k)$ is defined, we have that $T(\langle m_1, \ldots, m_k \rangle) \downarrow = f(m_1, \ldots, m_k)$, and $T(\langle m_1, \ldots, m_k \rangle) \uparrow$ otherwise.

We allow our definition to include *partial functions* (those defined on a subset of $\mathbb{N}^k$). When this happens, we require that $T(\langle m_1, \ldots, m_k \rangle) \uparrow$ for the inputs on which $f(m_1, \ldots, m_k)$ is undefined. Thus, every Turing machine T uniquely defines an n-variable partial function for each $n > 0$ (though different machines can define the same function).

We can now give a large class of functions which are partial computable.

Theorem 1.10 (Closure properties of partial computable functions)**.**
a) (Basic functions) *For each $i \leq k$, the* projection function $(n_1, \ldots, n_k) \mapsto n_i$ *is partial computable.*
b) (Basic functions) *The* constant function *with value* 0 *(that is, $n \mapsto 0$), and the* successor function $n \mapsto n+1$*, are partial computable.*
c) (Composition) *If f is a partial computable function on k variables, and $g_1, \ldots, g_k$ are partial computable functions each on l variables, then the function h on l variables given by*

$$h(n_1, \ldots, n_l) := f(g_1(n_1, \ldots, n_l), \ldots, g_k(n_1, \ldots, n_l))$$

is also partial computable. Note that we take h as being defined on $(n_1, \ldots, n_l)$ when each g_i is defined on $(n_1, \ldots, n_l)$ and *f is defined on $(g_1(n_1, \ldots, n_l), \ldots, g_k(n_1, \ldots, n_l))$.*
d) (Recursion) *If f and g are partial computable functions on k and $k+2$*

variables respectively, then the following function h on $k+1$ variables defined inductively by

$$h(n_1, \ldots, n_k, 0) := f(n_1, \ldots, n_k)$$
$$h(n_1, \ldots, n_k, n_{k+1} + 1) := g(n_1, \ldots, n_k, n_{k+1}, h(n_1, \ldots, n_k, n_{k+1}))$$

is partial computable (and defined if and only if f and g are appropriately defined).
e) (Minimisation) *If f is a partial computable function on $k+1$ variables, then the following partial function g on k variables defined by*

$$g(n_1, \ldots, n_k) := \begin{cases} n & \text{if } f(n_1, \ldots, n_k, n) = 0 \text{ and } f(n_1, \ldots, n_k, m) > 0 \ \forall m < n \\ \text{undefined} & \text{otherwise} \end{cases}$$

is partial computable.

Parts a) and b) of this theorem say that the *basic functions* are partial computable, part c) says that partial computable functions are closed under *composition*, part d) says that partial computable functions are closed under *recursion*, and part e) says that partial computable functions are closed under *minimisation*. For a proof of this theorem, see [1, §I Theorem 3.3].

Definition 1.11 (Partial recursive functions)**.**
We define the class of *partial recursive functions* as the smallest class of partial functions from $\mathbb{N}^n \to \mathbb{N}$ (for all n) which is closed under the properties of (1.10). That is, such a function f can be constructed from the basic functions and applications of composition, recursion, and minimisation a finite number of times. If f can be constructed without minimisation, then we say that it is *primitive recursive*.

Lemma 1.12.
Every primitive recursive function $f : \mathbb{N}^n \to \mathbb{N}$ is total *(that is, defined on all of $\mathbb{N}^n$).*

Proof. The projection, constant, and successor functions are obviously total. The composition of total functions is again total. Finally, performing primitive recursion on total functions is again total. □

We point out (without proof) that not every total recursive function is primitive recursive; the *Ackermann function*[1] is one such example.

The key idea of computability theory is the following result, which is a strengthening of (1.10).

Theorem 1.13.
A function f is partial computable if and only if it is partial recursive.

The reverse direction of this is given in (1.10). For a proof of the forward direction, see the discussion after [1, §I Theorem 3.3]. For a full and detailed proof of both directions of this theorem using *Register machines* instead of Turing machines, see [5, §4].

We now make the following definition of 'algorithm', and discuss the merits of this definition in Section 1.4.

[1] A very interesting function, which we do not have time to cover.

Definition 1.14 (Algorithms)**.**
An *algorithm* is any process which takes as input some recursive subset of $\mathbb{N}^k$, and which can be simulated by a Turing machine. A *total algorithm* is one which will always terminate on every element in its input set. A *partial algorithm* is one which may fail to terminate on some elements in its input set.

1.3. **Encodings.**

One problem with our formulation of algorithms and computable functions is that they *only* take in to account computations from $\mathbb{N}^k$ to $\mathbb{N}$. Thus, strictly speaking, we can't consider the process 'take a word in the English language, and compute the number of letters in it' as a computable function; the input is not a k-tuple of integers. Similarly for the process 'take an integer n, and compute the first word in the English dictionary with n letters', as the output is not an integer. As we will soon see, we need ways to *encode* our inputs as k-tuples, and our outputs as integers. We start with ways of encoding *words* as integers.

It helps to have a notion of how to produce an ordered list of the elements of $\mathbb{N}^m$. There are many ways to do this; one such way is called the *shortlex ordering*.

Definition 1.15 (Shortlex ordering)**.**
We define the *shortlex* ordering on $\mathbb{N}^m$ as follows: $(n_1, \ldots, n_m) < (n'_1, \ldots, n'_m)$ if $\Sigma_{i=1}^m n_i < \Sigma_{i=1}^m n'_i$ or $\Sigma_{i=1}^m n_i = \Sigma_{i=1}^m n'_i$ and for some k we have $n_i = n'_i$ for all $1 \leq i \leq k$ but $n_{k+1} < n'_{k+1}$.

We can use shortlex to produce an 'indexed list' of $\mathbb{N}^m$, as an alternative to Cantor's pairing function (1.8): Take all elements $(n_1, \ldots, n_m)$ with sum of entries $\Sigma_{i=1}^m n_i = 0$, and order these by shortlex. Then take all elements $(n_1, \ldots, n_m)$ with sum of entries $\Sigma_{i=1}^m n_i = 1$, and order these by shortlex. And so on. Thus, for each $n \in \mathbb{N}$, we can construct the n^{th} element of $\mathbb{N}^m$ in this list.

We can use this idea to encode words as integers. Consider the set Σ^+ of all words over the finite alphabet Σ. By placing an ordering $\{\sigma_1, \ldots, \sigma_n\}$ on Σ, we can represent each letter σ_i of Σ by the integer i. By restricting the shortlex ordering to $\{1, \ldots, n\}^m$ for each m, we get an induced ordering of Σ^m (words of length m) for each m: given a word $w \in \Sigma^m$, we can associate to it an m-tuple $(i_1, \ldots, i_m)$ representing the sequence of letters in w, and then we use the shortlex ordering on these associated tuples. Now, to produce an indexed list of Σ^+, we first take all words in Σ and order them (via their tuples) by the induced shortlex. Then take all words in Σ^2 and order them by the induced shortlex, and then Σ^3, and so on. Thus, for each $n \in \mathbb{N}$, we can construct the n^{th} element of Σ^+ in this list.

Using this, we may re-interpret our previous question of 'take a word in the English language, and compute the number of letters in it' as 'take the *index* for a word in the English language, and compute the number of letters in it', and thus our function is computable as its input and output are both integers. We will always require our inputs/outputs to be integers. Thus,

From hereon we will take it as a given that the inputs/outputs of our algorithms are given by codes of various objects, either by explicitly giving an encoding, or implicitly without going in to the details.

We now give a ways of encoding *machines* as integers, as later we will want to compute things about machines. There are various ways to uniformly encode Turing machines as integers. Some of these are bijective (one machine $\leftrightarrow$ one integer). We give an encoding here which is not bijective (in particular, there are integers which do not correspond to machines).

Definition 1.16 (Encoding Turing machines as natural numbers)**.**
We suppose that all Turing machines draw states from the infinite well-ordered set $\{q_0, q_1, \ldots\}$, and alphabet from the infinite well-ordered set $\{s_0, s_1, \ldots\}$, with the usual conventions on q_0, q_1, s_0. Fix the well-ordering $Z = \{L, R, q_0, s_0, q_1, s_1, \ldots\}$. Observe that each quadruple can be written as (i, j, k, l) (where i, j, k, l denote positions in Z). So order the quadruples of our Turing machine T lexicographically using the above well-ordering. For the t^{th} quadruple in the list we define

$$s_t := p_t^{p_i p_j p_k p_l}$$

where p_m is the m^{th} prime. If T has r quadruples, then form the integer

$$n(T) = s_1 \cdots s_r$$

which encodes our Turing machine T.

As mentioned, this encoding is not bijective; the integer 2^5 cannot be a code for any Turing machine.

1.4. **Church-Turing thesis.**
It has been proposed by Church and Turing that the following idea, though inherently unprovable because of it's lack of formalisation, is essentially true (and we paraphrase somewhat here):

Any finite written description of a deterministic step-by-step computation is equivalent to some Turing machine. Moreover, there is a construction that, given such a finite description, will give us an explicit Turing machine T that carries out the original computation.

This is colloquially referred to as the *Church-Turing thesis* (sometimes abbreviated to *Church's thesis*). We will make very frequent use of this idea, as at several places in our proofs we will describe, in words, an algorithmic process. This is often referred to as 'Proof by the Church-Turing thesis'. We do not suggest that you formalise all these proofs, as such actions are usually as unenlightening as they are time consuming. We will refrain from explicitly stating that we are appealing to the Church-Turing thesis in proofs, but these instances should be fairly self-evident when we make statements like 'Begin an enumeration of .....', or 'Run the following infinite collection of infinite enumerations as a diagonal process via interleaving....'.

Thesis 1.17 (Church-Turing thesis)**.**

(1) Any abstract theory of finite computation C will give at most *the set of partial recursive functions as its set of C partial computable functions from $\mathbb{N}^k$ to $\mathbb{N}$. Thus the most powerful theory of finite computation is given by Turing machines and their many equivalents.*

(2) Any informal written description of a step-by-step deterministic process with a finite description at every step, a finite set of rules, and finite input/output, starting with some tuple in $\mathbb{N}^k$ and with only integer output, is equivalent to some Turing machine computation.

(3) There is an algorithm that, given a code for such a finite English description as above, will produce a code n for a Turing machine T that carries out the finite process so described.

(Noting that, in the same way that we encoded words over Σ as integers, we can also encode all finite phrases in the English language as integers. We omit the details here; it is very similar to the previous example).

So now we have an algorithm to determine if an integer represents a Turing machine; the reverse of (1.16) where we break down n into products of distinct prime powers, then break down those powers and make sure they only have 4 (possibly repeated) prime factors. Thus, by Church's thesis, the following total function is computable:

$$f(n) := \begin{cases} 1 & \text{if } n \text{ codes a Turing machine} \\ 0 & \text{otherwise} \end{cases}$$

Definition 1.18 (Functions from Turing machines)**.**
Let j_n be the n^{th} code of a Turing machine (j_1 is therefore the smallest code). We write T_n for the Turing machine encoded by j_n, and φ_n for the (1-variable) function $\varphi_n : \mathbb{N} \to \mathbb{N}$ computed by T_n.

So we see that each Turing machine T_n corresponds to a recursively enumerable set of integers, and in particular this set is the domain of definition of φ_n. For simplicity, we will write the domain of definition of φ_n from hereon as

$$W_n := \{x \in \mathbb{N} \mid \varphi_n(x) \downarrow\}$$

We call W_n the n^{th} *recursively enumerable set.*

We can now adapt Cantor's diagonal argument to construct an explicit function which is not partial recursive:

Lemma 1.19 (A function which is not partial recursive)**.**
Define the function $g : \mathbb{N} \to \mathbb{N}$ via

$$g(n) := \begin{cases} \varphi_n(n) + 1 & \text{if } \varphi_n(n) \text{ is defined} \\ 0 & \text{otherwise} \end{cases}$$

Then this is an explicit definition of a function which is not partial recursive.

Proof. We proceed by contradiction. Suppose g were partial recursive. Then there must be some N for which $g = \varphi_N$. Now observe what happens if we try and compute $g(N)$. We see that if $\varphi_N(N)$ were defined then we would have $g(N) = \varphi_N(N) + 1 \neq \varphi_N(N) = g(N)$. Thus $\varphi_N(N)$ is not defined. So by the definition of g we have $g(N) = 0$, thus giving that $\varphi_N(N) = 0$, and so $\varphi_N(N)$ is defined; a contradiction. □

Note that we need the clause '$g(n) = 0$ if $\varphi_N(n)$ is undefined', otherwise g would indeed be partial recursive. To see this, we again appeal to Church's thesis: we have a description of an algorithm to compute the values of $g(n)$ when it is defined (that is, we take the machine T_n and run it with input n). We will elaborate on this idea later.

1.5. **Computable and incomputable sets.** So what does it really mean for $X \subseteq \mathbb{N}$ to be recursively enumerable? The basic idea is that there exists a process which outputs integers in a sequence $a_1, a_2, \ldots$ such that:

1. ONLY elements of X appear in the list.

2. ALL elements of X eventually appear in the list (possibly with repetition). How do we get this list? As X is recursively enumerable, there exists n such that $X = W_n = \Omega(T_n)$. So take one copy of T_n, input 1, and do 'one step' of the computation of $T_n(1)$ (that is, carry out the instructions of one quadruple). Now take another copy of T_n, do one step of $T_n(2)$, and one *more* step of $T_n(1)$. Then take a new copy of T_n, do one step of $T_n(3)$, one more step of $T_n(2)$, and one more step of $T_n(1)$. This is a very long diagonal process. Each time one of the $T_n(k)$ halts, we add k to our sequence. As all these steps happen in a given order, the sequence will have a well-ordering (the order on which the machines halted).

The much stronger notion of being computable comes from this 'listing' process. If $X \subseteq \mathbb{N}$ is computable, then we can construct two processes, one which outputs a sequence $a_1, a_2, \ldots$ for X, and another which outputs a sequence $b_1, b_2, \ldots$ for $\mathbb{N} \setminus X$. We can use this to compute membership in X as follows: given an integer k, simply look for k down the list $a_1, a_2, \ldots$, as well as down the list $b_1, b_2, \ldots$ (do this as a diagonal process: check a_1, then b_1, then a_2, then b_2, ...). Since these lists are disjoint, and their union is $\mathbb{N}$, we know that k will appear in precisely one list, which will tell us if $k \in X$ or $k \notin X$.

Note that being recursive is equivalent to having a characteristic function which is *computable*: there is an algorithm which takes as input an integer n and always eventually halts, outputting 1 or 0 depending on whether n lies in the set or not. Being r.e. is equivalent to having a characteristic function which is *partially computable*: there is an algorithm which takes as input an integer n, which eventually halts with output 1 when n lies in the set, but does not halt if n does not lie in the set.

Theorem 1.20 (Equivalent definitions of recursively enumerable sets)**.**
For a set $E \subseteq \mathbb{N}$, the following are equivalent:
a) $E = \{\varphi_n(m) \mid m \in \mathbb{N}\}$ for some fixed n. That is, E is the range of some partial recursive function φ_n.
b) $E = \{m \in \mathbb{N} \mid \varphi_n(m) \downarrow\}$ for some fixed n. That is, E is the domain of definition of some partial recursive function φ_n.
c) The function ϕ_E defined by

$$\phi_E(n) := \begin{cases} 1 & \text{if } n \in E \\ \uparrow & \text{otherwise} \end{cases}$$

is partial recursive. That is, E is recursively enumerable.
d) The function ψ_E defined by

$$\psi_E(n) := \begin{cases} n & \text{if } n \in E \\ \uparrow & \text{otherwise} \end{cases}$$

is partial recursive.

Proof.
$(b) \Rightarrow (c)$: For a given $m \in \mathbb{N}$, we start computing $\varphi_n(m)$. If this halts, output 1. If not, output nothing. This is a complete description of a way to compute ϕ_E, so by Church's thesis ϕ_E is partial recursive.
$(c) \Rightarrow (d)$: For a given $m \in \mathbb{N}$, we start computing $\phi_E(m)$. If this halts, output m. If not, output nothing. This is a complete description of a way to compute ψ_E, so by Church's thesis ϕ_E is partial recursive.
$(d) \Rightarrow (a)$: This is immediate.

$(a) \Rightarrow (b)$: Given the Turing machine T_n for computing the function $\varphi_n : \mathbb{N} \to \mathbb{N}$ with range E, we describe the following algorithm Q. For each $t \in \mathbb{N}$, we start a diagonal process which starts computing φ_n for all of its inputs. Each time $\varphi_n(m)$ halts in this diagonal process, compare the output to t; if we eventually find one such output is equal to t, then Q terminates on t and outputs 1 (if we never find such an output, then Q is undefined on t). As we have given a full description of the algorithm for Q, then by Church's thesis we can find a Turing machine which computes Q, and thus a partial computable function whose domain is E. □

One of the most important results in computer science is the following theorem, which first appeared in Turing's seminal on computation [3].

Theorem 1.21 (Turing, 1937)**.**
There exists a universal Turing machine*. That is, a Turing machine which can simulate the action of every Turing machine.*

Proof. We define an algorithm as follows: on input of an integer $x = \langle m, n \rangle$, we (begin to) compute $T_m(n)$. Since this is a verbal description of an algorithm, we can conclude that there exists some Turing machine T such that $T(x) = T_m(n)$ (that is, if $T_m(n) \downarrow$ with output k then $T(x) \downarrow$ with output k, and if $T_m(n) \uparrow$ then $T(x) \uparrow$). □

There is a very important set of integers, which forms the basis for most incomputable problems in mathematics.

Definition 1.22.
The Halting Set, denoted $\mathbb{K}$, is given by

$$\mathbb{K} := \{n \in \mathbb{N} \mid T_n(n) \downarrow\} = \{n \in \mathbb{N} \mid n \in W_n\}$$

This 'diagonal' set seems obscure, but has the following useful properties:

Theorem 1.23.
The halting set $\mathbb{K}$ satisfies the following:
1. *$\mathbb{K}$ is recursively enumerable.*
2. *$\mathbb{K}$ is not recursive.*

Proof. Clearly, $\mathbb{K}$ is recursively enumerable; in the same way that we defined a universal Turing machine $T(x) := T_m(n)$ (where $x = \langle m, n \rangle$), we can define a 'restricted' universal Turing machine that computes *one* entry of each Turing machine, by $T'(n) := T_n(n)$. Again, this is a verbal description of an algorithm, so by the Church-Turing thesis we can indeed construct a Turing machine which performs this computation. Thus $\mathbb{K} = \Omega(T')$, and so $\mathbb{K}$ is r.e.

Now, suppose that $\mathbb{K}$ was recursive. Then there exists some m such that $W_m = \mathbb{N} \setminus \mathbb{K}$. Now look at the index m; we want to know if m is in $\mathbb{K}$ or not. So we have:

$$\begin{aligned} m \in \mathbb{K} &\Leftrightarrow T_m(m) \downarrow \\ &\Leftrightarrow m \in W_m \\ &\Leftrightarrow m \in \mathbb{N} \setminus \mathbb{K} \\ &\Leftrightarrow m \notin \mathbb{K} \end{aligned}$$

This is a contradiction, so m cannot possibly exist. That is, there is no m such that $W_m = \mathbb{N} \setminus \mathbb{K}$, and so $\mathbb{N} \setminus \mathbb{K}$ is not recursively enumerable. □

So $\mathbb{K}$ is a set which is *described* by a Turing machine, but for which membership cannot be *computed* by any Turing machine. That is, deciding membership in $\mathbb{K}$ is our first provably *incomputable* problem!

Note that, in general, if we have a recursively enumerable set of Turing machines $X = \{T_{i_1}, T_{i_2}, \ldots\}$ (that is, the set $\{i_1, i_2, \ldots\}$ is r.e.) which take as input values in $\mathbb{N}$, then we can construct a (quasi-universal) Turing machine T which simulates the action of all the machines in X. This is given by $T(\langle m, n\rangle) := T_{i_m}(n)$, and if $|X| < m$ then $T(\langle m, n\rangle)\uparrow$. Note that we do not, *a priori*, need to know the size of X; we just start enumerating these machines $T_{i_1}, T_{i_2}, \ldots$, and if we ever get to T_{i_m} then we input n into T_{i_m} to (try and) compute $T_{i_m}(n)$.

Example 1.24. *Let $\{T_i\}_{i\in I}$ be a collection of Turing machines. Then*
1. *If I is finite then $\bigcap_{i\in I} W_i$ is r.e.*
2. *If I is r.e. then $\bigcup_{i\in I} W_i$ is r.e.*

1.6. **The s-m-n theorem and Kleene's recursion theorem.**

We now introduce a useful idea, which shows that holding some of the variables of a partial recursive function fixed gives us another partial recursive function (which we can construct a Turing machine for). This is often referred to as *currying*, named after Haskell Curry. The theorem itself is called the 's-m-n theorem', named after the notation used in the original proof by Kleene[2].

Theorem 1.25 (The s-m-n theorem)**.**
For all $m, n > 0$, a partial function $h : \mathbb{N}^{m+n} \to \mathbb{N}$ is partial recursive if and only if there is a total recursive function $g : \mathbb{N}^m \to \mathbb{N}$ such that, for all $(e_1, \ldots, e_m, x_1, \ldots, x_n) \in \mathbb{N}^{m+n}$, we have that

$$h(e_1, \ldots, e_m, x_1, \ldots, x_n) = \varphi_{g(e_1,\ldots,e_m)}(\langle x_1, \ldots, x_n\rangle)$$

Here '=' is interpreted to include 'one side is defined iff the other side is'.

Proof. Suppose h satisfies the hypotheses of the theorem; we show that it is partial recursive. Given input $(e_1, \ldots, e_m, x_1, ..., x_n)$, we first compute the total recursive function $g(e_1, \ldots, e_m) = M$, and then start the computation of $\varphi_M(\langle x_1, \ldots, x_n\rangle)$ via the Turing machine T_M. If the computation of $\varphi_M(\langle x_1, \ldots, x_n\rangle)$ ever halts, then we take the output as the value of $h(e_1, \ldots, e_m, x_1, \ldots, x_n)$. Given that we have completely described an algorithm to partially compute h, then by Church's thesis we have that h is partial recursive.

Now, suppose that h is partial recursive. For each $(e_1, \ldots, e_m) \in \mathbb{N}^m$, we describe a function $k_{(e_1,\ldots,e_m)} : \mathbb{N}^n \to \mathbb{N}$ as follows: given input $(x_1, \ldots, x_n)$, start the computation of $h(e_1, \ldots, e_m, x_1, \ldots, x_n)$, and if this halts, take the output as $k_{(e_1,\ldots,e_m)}(x_1, \ldots, x_n)$. Thus we have a complete description of an algorithm which partially computes $k_{(e_1,\ldots,e_m)}$, thus by Church's thesis we can construct, from $(e_1, \ldots, e_m)$, a code (call it $g(e_1, \ldots, e_m)$) for a Turing machine $T_{g(e_1,\ldots,e_m)}$ which partially computes the function $k_{(e_1,\ldots,e_m)}$. That is,

$$k_{(e_1,\ldots,e_m)} = \varphi_{g(e_1,\ldots,e_m)} : \mathbb{N}^n \to \mathbb{N}$$

But this is a total algorithm which describes how to construct the (total) function $g : \mathbb{N}^m \to \mathbb{N}$, and so by another application of Church's thesis we see that g is total recursive. As $h(e_1, \ldots, e_m, x_1, \ldots, x_n) = \varphi_{g(e_1,\ldots,e_m)}(\langle x_1, \ldots, x_n\rangle)$ by definition, we have that h satisfies the required conditions. □

[2]And not for any deeper or more insightful reason.

We can use the s-m-n theorem to show another important result in computability theory: Kleene's recursion theorem.

Theorem 1.26 (Kleene recursion theorem)**.**
Let $f : \mathbb{N} \to \mathbb{N}$ be a recursive function. Then there exists $n \in \mathbb{N}$ with $\varphi_n = \varphi_{f(n)}$.

Proof. Let $h : \mathbb{N} \to \mathbb{N}$ be a total recursive function. Define the partial function $k : \mathbb{N}^2 \to \mathbb{N}$ by

$$k(n,x) = \begin{cases} \varphi_{\varphi_n(n)}(x) & \text{if } \varphi_n(n)\downarrow \text{ and } \varphi_{\varphi_n(n)}(x)\downarrow \\ \uparrow & \text{otherwise} \end{cases}$$

Then by Church's thesis, k is partial recursive. So by the s-m-n theorem (1.25), there is a total computable function $g : \mathbb{N} \to \mathbb{N}$ such that

$$k(n,x) = \varphi_{g(n)}(x)$$

Thus $h \circ g$ is total recursive; say $h \circ g = \varphi_N$. Then, $\forall x \in \mathbb{N}$, we have

$$\begin{aligned} \varphi_{g(N)}(x) &= k(N,x) \quad \text{(definition of } g\text{)} \\ &= \varphi_{\varphi_N(N)}(x) \quad \text{(definition of } k\text{)} \\ &= \varphi_{h\circ g(N)}(x) \quad \text{(definition of } N\text{)} \end{aligned}$$

So $\varphi_{g(N)} = \varphi_{h\circ g(N)}$ as functions (that is, $g(N)$ is our fixed point). □

1.7. **Reductions.**

We now look at another way to show that certain sets are not recursive (or even r.e.), and that is via *reductions*. Intuitively, we look for ways to conclude, in a computational manner, membership in one set X from membership in another set Y. Thus, if we know that we can't decide membership in X, then it means we can't decide membership in Y.

Definition 1.27 (Many-one reductions)**.**
Given two sets $A, B \subseteq \mathbb{N}$, a *many-one reduction* of A to B is a total recursive function $f : \mathbb{N} \to \mathbb{N}$ such that, for all $n \in \mathbb{N}$, we have

$$n \in A \Leftrightarrow f(n) \in B$$

If there is a many-one reduction of A to B, then we say that A *many-one reduces to* B, or A *is many-one reducible to* B, and we write this as $A \leq_m B$.
$A, B \subseteq \mathbb{N}$ are said to be *many-one equivalent* if $A \leq_m B$ and $B \leq_m A$; in such a case we write $A \equiv_m B$.

So in order to compute membership of n in A, we evaluate the function $f(n)$ and then 'ask B one question: Is $f(n)$ in B or not?' If so, $n \in A$, if not, $n \notin A$; in either case, we *cannot* do any computation after this question. The name 'many-one reduction' comes from the fact that membership in A of several elements $n_1, n_2, \ldots$ can reduce to testing if one element lies in B (that is, we might have $f(n_1) = f(n_2) = \ldots$). This differs from the notion of *Turing reductions*, where we are allowed to ask B whether several elements lie inside or outside it, and we can carry out computational steps in between these questions.

It is clear that many-one reducibility is transitive. What is important to realise is that $A \leq_m B$ if we can compute membership in A using as many algorithmic steps as we like, and additionally asking *one* question to membership in B *at the very end*.

Lemma 1.28.
$A \leq_m B \;\Leftrightarrow\; \mathbb{N} \setminus A \leq_m \mathbb{N} \setminus B$.

Proof. This follows immediately from the definition of many-one reductions. □

Many-one reductions help us identify sets which are/aren't recursive or r.e.

Lemma 1.29.
a) If $A \leq_m B$, and B is r.e., then so is A.
b) If $A \leq_m B$, and B is recursive, then so is A.

Proof.
a) Let $A \leq_m B$ via the total recursive function f. As B is r.e., it is the domain of some partial recursive function g (that is, $x \in B \Leftrightarrow g(x)\downarrow$). So A is thus the domain of $g \circ f$, which is partial recursive. Hence A is also r.e.

b) Let $A \leq_m B$ via the total recursive function f. Then the same map f is a many-one reduction of $\mathbb{N} \setminus A$ to $\mathbb{N} \setminus B$ (as $x \notin A \Leftrightarrow f(x) \notin B$). Thus, by a), A and $\mathbb{N} \setminus A$ are both r.e. (as B and $\mathbb{N} \setminus B$ are both r.e.). So A is recursive. □

We use the s-m-n theorem to show that the halting set $\mathbb{K}$ is the strongest r.e. set under many-one reductions, in the following sense.

Theorem 1.30.
A set $X \subseteq \mathbb{N}$ is r.e. if and only if $X \leq_m \mathbb{K}$.

Proof. From (1.29), we see that if $X \leq_m \mathbb{K}$ then X must be r.e. (as $\mathbb{K}$ is). Now, suppose that X is r.e. Define the partial function $f : \mathbb{N}^2 \to \mathbb{N}$ via

$$f(e,n) := \begin{cases} 1 & \text{if } e \in X \\ \uparrow & \text{otherwise} \end{cases}$$

Then f is partial recursive; given an input (e,n) we begin computing $\phi_E(e)$, and this will halt iff $e \in X$. When it does, output 1 for $f(e,n)$. Thus, by Church's thesis, f is partial recursive. So by (1.25), there is a total recursive function $g : \mathbb{N} \to \mathbb{N}$ with $f(e,n) = \varphi_{g(e)}(n)$ for all $(e,n) \in \mathbb{N}^2$. So now we see that

$$\begin{aligned} e \in X &\Leftrightarrow f(e,g(e))\downarrow \\ &\Leftrightarrow \varphi_{g(e)}(g(e))\downarrow \\ &\Leftrightarrow g(e) \in \mathbb{K} \end{aligned}$$

So $e \in X \Leftrightarrow g(e) \in \mathbb{K}$, where g is a total recursive function. Thus $X \leq_m \mathbb{K}$. □

The idea here is that, with absolute knowledge of $\mathbb{K}$, we have absolute knowledge of each r.e. set, in a very nice computable way.

1.8. **Rice's theorem.**

One would, of course, like to compute things *about* r.e. sets. It would be useful, for example, to be able to determine (in a computable way) whether or not W_n is all of $\mathbb{N}$, as this would tell us precisely when φ_n is total. We will soon see that this is not possible; moreover, there is *no* (non-trivial) property of r.e. sets that we can compute!

Definition 1.31.
A *property* of r.e. sets is a map

$$\rho : \{X \subseteq \mathbb{N} \mid X \text{ is r.e.}\} \to \{0,1\}$$

where $0, 1$ represent 'false' and 'true' respectively.

For example, the property of 'being empty' is represented by the map

$$\rho(X) := \begin{cases} 1 & \text{if } X = \emptyset \\ 0 & \text{if } X \neq \emptyset \end{cases}$$

In order to compute whether an r.e. set has a particular property or not, we need a finite way to describe this r.e. set. We can take the integer n for the Turing machine T_n which describes the characteristic function of the r.e. set, but note that it is the *set* which does or doesn't have the property, independent of which Turing machine we pick to describe it (and there may be many). So really, we are computing $\rho(W_n)$ (and actually, we can view this as computing $\rho(n)$). So which properties can we compute?

Example 1.32. *The property 'being non-empty' is r.e. but not recursive. That is, the set $I = \{n \in \mathbb{N} \mid W_n \neq \emptyset\}$ is r.e., but not recursive.*

Proof. Take n and start a diagonal process to begin computing $\varphi_n(1), \varphi_n(2), \ldots$. One of these will terminate iff W_n is non-empty, and so this index set is r.e. by Church's thesis.
On the other hand, take an integer n, and define a partial function g via

$$g(n, x) := \begin{cases} 1 & \text{if } n \in \mathbb{K} \\ \uparrow & \text{otherwise} \end{cases}$$

So by Church's thesis g is partial recursive, and by (1.25) there is a recursive function $h : \mathbb{N} \to \mathbb{N}$ such that $g(n, x) = \varphi_{h(n)}(x)\ \forall (n, x) \in \mathbb{N}^2$. If $n \in \mathbb{K}$ then $W_{h(n)} = \mathbb{N} \neq \emptyset$. If $n \notin \mathbb{K}$ then $W_{h(n)} = \emptyset$. Thus $n \in \mathbb{N} \setminus \mathbb{K} \Leftrightarrow W_{h(n)} = \emptyset \Leftrightarrow h(n) \in \mathbb{N} \setminus I$, and so we have a many-one reduction from a non-r.e. set $\mathbb{N} \setminus \mathbb{K}$ to the set $\mathbb{N} \setminus I$, and so the latter is not r.e. Hence I is not recursive. □

Definition 1.33.
A property of r.e. sets ρ is said to be *nontrivial* if there exist two r.e. sets A, B such that $\rho(A) = 0$ and $\rho(B) = 1$. That is, not all sets have (or do not have) the property described.

It turns out that the only properties of r.e. sets we can algorithmically recognise are the trivial ones[3].

Theorem 1.34 (Rice's theorem)**.**
Let C be a non-trivial class of r.e. sets, and I the set of indices which give r.e. sets in C. That is,

$$I = \{n \in \mathbb{N} \mid W_n \in C\}$$

If $\emptyset \notin C$ then $\mathbb{K} \leq_m I$; if $\emptyset \in C$ then $\mathbb{K} \leq_m (\mathbb{N} \setminus I)$.

Proof. There are two cases to consider here.
Case 1: $\emptyset \notin C$. In this case, fix any r.e. set $\emptyset \neq A \in C$. Now define the following partial recursive function $g : \mathbb{N}^2 \to \mathbb{N}$ by

$$g(n, x) = \begin{cases} 1 & \text{if } n \in \mathbb{K} \text{ and } x \in A \\ \uparrow & \text{otherwise} \end{cases}$$

This is a description of how to compute if g halts on a given input, and so by Church's thesis g is partial recursive. So by (1.25) there is a total recursive function $h : \mathbb{N} \to \mathbb{N}$ such that $g(n, x) = \varphi_{h(n)}(x)\ \forall (n, x) \in \mathbb{N}^2$. Notice that $n \in \mathbb{K} \Rightarrow W_{h(n)} = A \Rightarrow W_{h(n)} \in C$, and $n \notin \mathbb{K} \Rightarrow W_{h(n)} = \emptyset \Rightarrow W_{h(n)} \notin C$.

[3]Hopefully by now this does not come as a surprise to you.

Thus $n \in \mathbb{K} \Leftrightarrow h(n) \in I$, and so we have a many-one reduction $\mathbb{K} \leq_m I$.
Case 2: $\emptyset \in C$ (analogous to the first case). In this case, fix any r.e. set $\emptyset \neq A \notin C$. Now define the following partial recursive function $g : \mathbb{N}^2 \to \mathbb{N}$ by

$$g(n,x) = \begin{cases} 1 & \text{if } n \in \mathbb{K} \text{ and } x \in A \\ \uparrow & \text{otherwise} \end{cases}$$

This is a description of how to compute if g halts on a given input, and so by Church's thesis g is partial recursive. So by (1.25) there is a total recursive function $h : \mathbb{N} \to \mathbb{N}$ such that $g(n,x) = \varphi_{h(n)}(x)\ \forall (n,x) \in \mathbb{N}^2$. Notice that $n \in \mathbb{K} \Rightarrow W_{h(n)} = A \Rightarrow W_{h(n)} \notin C$, and $n \notin \mathbb{K} \Rightarrow W_{h(n)} = \emptyset \Rightarrow W_{h(n)} \in C$. Thus $n \in \mathbb{K} \Leftrightarrow h(n) \in \mathbb{N} \setminus I$, and so we have a many-one reduction $\mathbb{K} \leq_m \mathbb{N} \setminus I$. □

Corollary 1.35.
Every non-trivial property of r.e. sets is undecidable (i.e., nonrecursive). That is, if ρ is a non-trivial property of r.e. sets, then the set

$$\{n \in \mathbb{N} \mid \rho(W_n) = 1\}$$

is not recursive.

Thus, if you are given an r.e. set W_n and asked some non-trivial question about it (i.e., Is it finite? Is it empty? Does it contain more than 55 elements? Does it contain 9 but not 6? Is it recursive? Is it co-finite? Are all its elements even?), then you have no way of answering in an algorithmic manner. You may be able to answer the question for *some* particular cases of n, but not for all cases.

1.9. **Degrees.**

Definition 1.36.
For a set $A \subseteq \mathbb{N}$, the *many-one degree* of A, written $[A]_m$, is the collection of sets which are many-one equivalent to A. That is,

$$[A]_m := \{B \subseteq \mathbb{N} \mid A \equiv_m B\}$$

This defines an equivalence relation on subsets of $\mathbb{N}$.

The set of many-one degrees forms a partially ordered set: we say $[A]_m \leq [B]_m$ if $A \leq_m B$.

Most of the time, we are only interested in the many-one degree of a set.

Definition 1.37.
A set $X \subseteq \mathbb{N}$ is said to be in the class Σ_n^0 if there exists a total recursive function $f : \mathbb{N}^{n+1} \to \mathbb{N}$ such that

$$X = \{x \in \mathbb{N} \mid (\exists y_1)(\forall y_2)(\exists y_3) \dots (\ \cdot\ y_n)(f(y_1, \dots, y_n, x) = 1)\}$$

where $(\exists y_1)(\forall y_2)(\exists y_3) \dots (\ \cdot\ y_n)$ is an alternating sequence of $\exists - \forall$ quantifiers over $\mathbb{N}$.
A set $X \subseteq \mathbb{N}$ is said to be in the class Π_n^0 if there exists a total recursive function $f : \mathbb{N}^{n+1} \to \mathbb{N}$ such that

$$X = \{x \in \mathbb{N} \mid (\forall y_1)(\exists y_2)(\forall y_3) \dots (\ \cdot\ y_n)(f(y_1, \dots, y_n, x) = 1)\}$$

where $(\forall y_1)(\exists y_2)(\forall y_3) \dots (\ \cdot\ y_n)$ is an alternating sequence of $\forall - \exists$ quantifiers over $\mathbb{N}$.
We then define $\Delta_n^0 := \Sigma_n^0 \cap \Pi_n^0$.

Definition 1.38.
A set X is said to be Σ_n^0*-hard* (resp. Π_n^0*-hard*) if every Σ_n^0 set (resp. Π_n^0 set) many-one reduces to X. That is, $A \in \Sigma_n^0$ (resp. $A \in \Pi_n^0$) implies $A \leq_m X$. A set X is said to be Σ_n^0*-complete* (resp. Π_n^0*-complete*) if it is Σ_n^0-hard and is itself Σ_n^0 (resp. Π_n^0-hard and is itself Π_n^0).

Example 1.39. *A set is recursive iff it is Σ_0^0. A set is r.e. iff it is Σ_1^0. The complement of a Σ_n^0 set is a Π_n^0 set, and vice-versa.*

Example 1.40. *By the previous example, we have that $\mathbb{K}$ is Σ_1^0. And by (*1.30*), we have that it is Σ_1^0-hard. Thus $\mathbb{K}$ is Σ_1^0-complete.*

Here are some more examples, which we will make use of later. For a proof of these, see [1].

Example 1.41.
1. *The set* $\text{Tot} := \{n \in \mathbb{N} \mid W_n = \mathbb{N}\}$ *is Π_2^0-complete.*
2. *The set* $\text{Fin} := \{n \in \mathbb{N} \mid |W_n| < \infty\}$ *is Σ_2^0-complete.*
3. *The set* $\text{Rec} := \{n \in \mathbb{N} \mid W_n \text{ is recursive }\}$ *is Σ_3^0-complete.*

The whole purpose of indexing these collections of sets is because they form a hierarchy of increasing 'difficulty', which we specify in the following theorem (found in [4] §14.2 Theorem II).

Theorem 1.42 (Arithmetic hierarchy theorem)**.**
a) $(\Sigma_n^0 \cup \Pi_n^0) \subset \Delta_{n+1}^0$ *for all n.*
b) For any $A \subseteq \mathbb{N}$, we have that $A \in \Sigma_n^0$ if and only if $\mathbb{N} \setminus A \in \Pi_n^0$.

Lemma 1.43.
If A is Σ_n^0 or Π_n^0 for some n, then it is Δ_m^0 (so Σ_n^0 and Π_n^0) for all $m < n$.

Having described many-one degrees in some detail, we now introduce Turing degrees. For a full exposition of this, see [1]; we give only a definition and brief discussion here.

Definition 1.44.
A function $f : \mathbb{N} \to \mathbb{N}$ is a B*-recursive function* if it can be computed by a Turing machine which, along with the standard computational steps described in (1.2), has unlimited access to an oracle for $B \subseteq \mathbb{N}$.

Given that an oracle always returns an answer in constant time, we have (by the Church-Turing thesis) that the above description does indeed describe an algorithm, and this is describable by a Turing machine (modulo the existence of an oracle for B).

Definition 1.45.
A set $A \subseteq \mathbb{N}$ is said to be *Turing reducible* to $B \subseteq \mathbb{N}$ (denoted $A \leq_T B$) if we can compute membership in A with unlimited access to an oracle for B. That is, the characteristic function $\chi_A : \mathbb{N} \to \mathbb{N}$ is B-recursive. Two sets A, B are said to be *Turing equivalent* (denoted $A \equiv_T B$) if both $A \leq_T B$ and $B \leq_T A$. The *Turing degree* of A, written $[A]_T$, is the collection of sets which are Turing equivalent to A. That is,

$$[A]_T := \{B \subseteq \mathbb{N} \mid A \equiv_T B\}$$

The set of Turing degrees forms a partially ordered set: we say $[A]_T \leq [B]_T$ if $A \leq_T B$.

It is clear that Turing reducibility is transitive, and Turing equivalence is an equivalence relation. What is important to realise is that $A \leq_T B$ if we can compute membership in A using as many algorithmic steps as we like, and additionally ask as many questions of membership in B as we like. Compare this to many-one reduction, where we were only able to ask one question to membership in B.

Moreover, $A \leq_m B$ implies $A \leq_T B$, but in general the converse does not hold. In addition, with the exception of the degree of the empty set, every Turing degree splits as the union of an infinite number of many-one degrees. So we see that many-one reducibility is much finer than Turing reducibility.

In the same way as we numbered our original Turing machines, we can number all Turing machines *relative to some oracle*; for any $B \subseteq \mathbb{N}$, we write T_n^B to be the n^{th} Turing machine with oracle for B, φ_n^B for the function it computes, and W_n^B to be its halting set. Note that the machine itself does *not* depend on the set B; only its operation. So T_n^A and T_n^B are always the same *machine*, even if $A \neq B$, but they operate differently and so we may have that $W_n^A \neq W_n^B$.

In general, an oracle Turing machine T_n^- (with no initial reference to the oracle) describes a computational binary tree for each input x. It is only when we specify the oracle B that the machine $T_n^B(x)$ describes a path in this tree.

1.10. **Modular machines.**

We define modular machines as an alternate way of mechanical computing. We will show that they can simulate Turing machines in a very natural way. These will be useful to us later, as it is easier to represent integers in groups than tapes of symbols.

Definition 1.46.
A *modular machine* $\mathcal{M}$ consists of an integer $m > 1$ and a finite set of quadruples each of the form (a, b, c, R) or (a, b, c, L), where $m > a \geq 0$ and $m > b \geq 0$ and $m^2 > c \geq 0$. We require that, for each such pair (a, b), there is at most one quadruple of $\mathcal{M}$ of the form $(a, b, *, *)$.
A *modular machine configuration* is an ordered pair $(\alpha, \beta) \in \mathbb{N}^2$. We write $(\alpha, \beta) \underset{\mathcal{M}}{\longrightarrow} (\alpha_1, \beta_1)$, called a *computational step* of $\mathcal{M}$, if $\alpha = um + a$ and $\beta = vm + b$ (with $0 \leq a, b < m$) and there exists c such that either:

1. $(a, b, c, R) \in \mathcal{M}$ and $\alpha_1 = um^2 + c$ and $\beta_1 = v$, or
2. $(a, b, c, L) \in \mathcal{M}$ and $\alpha_1 = u$ and $\beta_1 = vm^2 + c$.

Note that the action of $\mathcal{M}$ on (α, β) depends only on the class of (α, β) modulo m. This is why we call $\mathcal{M}$ a *modular* machine.
We write $(\alpha, \beta) \xrightarrow[\mathcal{M}]{*} (\alpha', \beta')$ if there exists a finite sequence

$$(\alpha, \beta) = (\alpha_1, \beta_1) \underset{\mathcal{M}}{\longrightarrow} (\alpha_2, \beta_2) \underset{\mathcal{M}}{\longrightarrow} \dots \underset{\mathcal{M}}{\longrightarrow} (\alpha_n, \beta_n) = (\alpha', \beta')$$

Such a sequence is called a *computation* of $\mathcal{M}$.
If, for $\alpha = um + a$, $\beta = vm + b$ $(0 \leq a, b < m)$, no quadruple of $\mathcal{M}$ begins with (a, b), then we say (α, β) is *terminal*. If $(0, 0)$ is terminal in $\mathcal{M}$, then we define the set

$$H_0(\mathcal{M}) := \{(\alpha, \beta) \mid (\alpha, \beta) \xrightarrow[\mathcal{M}]{*} (0, 0)\}$$

We will now see that modular machines can completely simulate the action of Turing machines. First, it helps to slightly re-define Turing machines to have

quintuples (rather than quadruples), all of the form (q_i, s_j, s_k, q_l, Z), where $Z \in \{L, R\}$.

These act on tapes in almost the same way as quadruple-Turing machines. The machine, in state q_i, reading letter s_j, re-writes the letter to s_k, and then moves left or right 1 square (depending on whether $Z = L$ or R). The rest of the functionality of a quintuple-Turing machine is identical to that of a quadruple-Turing machine. Obviously a 'moving' quadruple from our original definition in (1.2) of the form (q_i, s_j, L, q_l) (or (q_i, s_j, R, q_l)) can be re-written in this quintuple form as (q_i, s_j, s_j, q_l, L) (or (q_i, s_j, s_j, q_l, R)); the two act identically on instantaneous descriptions.

To make the two definitions computationally equivalent, we replace each 'non-moving' quadruple (q_i, s_j, s_k, q_l) with the quintuple $(q_i, s_j, s_k, q_{ijl}, R)$ where q_{ijl} is a new auxiliary state, together with quintuples (q_{ijl}, x, x, q_l, L) for all alphabet symbols $x \in S$. That is, we 'write on the tape and move right, and then immediately move left again while leaving the tape unchanged'. Furthermore, by keeping the same alphabet, we have fully constructed a quintuple-Turing machine from a quadruple-Turing machine.

It is then clear that if T is a quadruple-Turing machine, and T' is its associated quintuple-Turing machine as defined above, then the two are computationally equivalent in the sense that if $w \in S^+$ (S being the alphabet of both T and T'), then

$$T(w) \downarrow = v \;\Leftrightarrow\; T'(w) \downarrow = v$$

We may now consider all our Turing machines to be in quintuple form; if we are given one in quadruple form, we convert it (algorithmically, as above) into quintuple form.

We now describe how to convert a quintuple-Turing machine into an equivalent modular machine. So, take a quintuple-Turing machine T with alphabet S and states Q. Set $m = |S| + |Q| + 1$. Now re-write the symbols S as integers $\{0, \ldots, n\}$, and the states Q as integers $\{n+1, \ldots, m-1\}$. We now define, from the quintuples of T, an associated modular machine $\mathcal{M}$ with modulus m.

First, we describe how to interpret an instantaneous description of a Turing machine as a pair of integers. We do this as follows:
Suppose $b_k \cdots b_1 b_0 q a c_0 c_1 \cdots c_l$ (which we shall call C) is an instantaneous description of T. In our re-writing convention above, we can consider b_i's, c_i's, a to all lie in $\{0, \ldots, n\}$, and q to lie in $\{n+1, \ldots, m-1\}$ (recall that we have re-written the symbols and states of T as integers between 0 and $m-1$).

We set $u := \Sigma_{i=0}^{k} b_i m^i$, $v := \Sigma_{i=0}^{l} c_i m^i$, and then to the instantaneous description C we associate *two* modular machine configurations. These are

1. $(um + a, vm + q)$, called the *left associate* of C, and
2. $(um + q, vm + a)$, called the *right associate* of C.

Now we define the modular machine $\mathcal{M}$ associated to T to have modulus m as above, and for each quintuple (q_i, s_j, s_k, q_l, D) in T (where $D \in \{L, R\}$), except those of the form $(q_0, *, *, *, *)$, we include *both* of the following two quadruples in $\mathcal{M}$:

1. $(q_i, s_j, s_k m + q_l, D)$, and
2. $(s_j, q_i, s_k m + q_l, D)$.

Notice that $\mathcal{M}$ always takes associates to associates, but might flip left $\leftrightarrow$ right.

We will now illustrate how the modular machine $\mathcal{M}$ mimics the action of T. Take a word $w \in S^+$ of T, and compute the left associate of $q_1 w$. Now, suppose

we have *any* general instantaneous description C as above, converted to either its left or right associate.

First, suppose T has the quintuple (q, a, a', q', R). Then this quintuple acting on C would yield the instantaneous description $b_k \cdots b_1 b_0 a' q' c_0 c_1 \cdots c_l$. Had we been considering the left (resp. right) associate of C, which is $(um+a, vm+q)$ (resp. $(um+q, vm+a)$), then the *one* quadruple from $\mathcal{M}$ that we could apply here would be $(a, q, a'm+q', R)$ (resp. $(q, a, a'm+q', R)$). Thus, in either case, this one computational step of $\mathcal{M}$ would yield the right associate $(um^2 + a'm + q', v) = \big(\big(\Sigma_{i=0}^{k} b_i m^{i+1} + a'\big)m + q' \ , \ \big(\Sigma_{i=1}^{l} c_i m^{i-1}\big)m + c_0 \big)$. This corresponds to the instantaneous description $b_k \cdots b_1 b_0 a' q' c_0 c_1 \cdots c_l$, as expected.

If instead T has the quintuple (q, a, a', q', L), then this quintuple acting on C would yield the instantaneous description $b_k \cdots b_1 q' b_0 a' c_0 c_1 \cdots c_l$. Had we been considering the left (resp. right) associate of C, which is $(um+a, vm+q)$ (resp. $(um+q, vm+a)$), then the *one* quadruple from $\mathcal{M}$ that we could apply here would be $(a, q, a'm+q', L)$ (resp. $(q, a, a'm+q', L)$). Thus, in either case, this one computational step of $\mathcal{M}$ would yield the left associate $(u, vm^2+a'm+q') = \big(\big(\Sigma_{i=1}^{k} b_i m^{i-1}\big)m + b_0 \ , \ \big(\Sigma_{i=0}^{l} c_i m^{i+1} + a'\big)m + q' \big)$. This corresponds to the instantaneous description $b_k \cdots b_1 q' b_0 a' c_0 c_1 \cdots c_l$, as expected.

Finally, note that $(am+q_0, bm+s)$ and $(am+s, bm+q_0)$ are terminal in $\mathcal{M}$ for all $s \in S$ and all $a, b \geq 0$, and are the *only* associates which are terminal. These correspond to associates of instantaneous descriptions containing the halting state q_0. When we reach such an associate, $\mathcal{M}$ 'terminates', and we convert the associate back to an instantaneous description giving us the output word of $T(w)$.

We summarise the discussion above with the following theorem.

Theorem 1.47.
Given any Turing machine T in quadruple form, we can construct from it a computationally equivalent Turing machine T' in quintuple form. Given any Turing machine T' in quintuple form, we can construct from it a modular machine $\mathcal{M}$ which simulates the action of T'.

We need to take this one step further, and, for a quadruple-Turing machine T, relate its halting set $\Omega(T)$ to the set $H_0(\mathcal{M})$ where $\mathcal{M}$ is its associated modular machine. We do this as follows: after having formed the quintuple-Turing machine T' as described in (1.47), we make a new quintuple-Turing machine T'' by introducing a dummy symbol h into the alphabet, and two dummy states q_L, q_R. When the machine would otherwise enter the halting state q_0, make it instead enter q_L. Then, add some extra instructions so that once it enters q_L it keeps scanning left and re-writing all the squares with s_0. When it moves past the left-end of the tape it adds a new square with h on it. Then it reads that h, re-writes it with s_0 and goes into state q_R and repeats the 's_0-writing' process; this time to the right. When it moves past the right-end of the tape, it adds a new square with h on it. Then it reads h, re-writes it with s_0, and enters the original halting state q_0. To do this formally, the extra instructions needed are (q_L, s, s_0, q_L, L) $\forall s \in S$, (q_L, h, s_0, q_R, R), (q_R, s, s_0, q_R, R) $\forall s \in S$, (q_R, h, s_0, q_0, L), noting that when we move past the end of the tape when in state q_L or q_R, we add a new square with h on it (and not s_0). The point of doing this transformation is that whenever T'' halts, it outputs a tape consisting entirely of s_0's (i.e., it 'wipes the tape' just before halting).

If we construct from this modified quintuple-Turing machine T'' an equivalent modular machine $\mathcal{M}$ as per (1.47) *but where we insist that* s_0 *is assigned the integer* 0 *in the re-labeling and, in a slight deviation from convention, we also assign* 0 *to* q_0 (one can go back and check that this does not ruin the intended function of $\mathcal{M}$), then we can conclude that $T(w)\downarrow$ iff the left associate (α,β) of q_1w in $\mathcal{M}$ satisfies $(\alpha,\beta)\xrightarrow[\mathcal{M}]{*}(0,0)$. That is:

Theorem 1.48.
Let T be a quadruple-Turing machine. Then from it we can construct a modular machine $\mathcal{M}$ such that, if $w\in S^+$ is a word on the alphabet of T, and (α,β) is the left associate of q_1w in $\mathcal{M}$, then

$$T(w)\downarrow \;\Leftrightarrow\; (\alpha,\beta)\in H_0(\mathcal{M})$$

The exposition leading to (1.47) contained several important new ideas and concepts. In contrast, the exposition leading to (1.48) can be described as the analogue in computability theory of *abstract nonsense*; there is nothing deep in (1.48), and in the literature one would just explain (1.48) with the argument 'we can clear the tape before halting', with no further justification.

1.11. **Minsky machines.**

We define Minsky machines as yet another alternate way of mechanical computing. We will state how they simulate Turing machines. These will be useful to us later when studying Slobodskoĭ's Theorem on the undecidability of the universal theory of finite groups [6].

Definition 1.49.
We define the 2-tape *Minsky Machine* as follows: Take two right-infinite tapes such that the leftmost cell of each tape contain a 1, and all other cells contain a 0. The machine has internal states $\{q_0,\dots q_M\}$; q_1 is the initial state and q_0 is the terminal state.

The internal instructions of this machine are of the form

$$q_iab\to q_jT_\alpha T_\beta$$

where $i\in\{1,\dots,n\}$, $j\in\{0,\dots,n\}$, $a,b\in\{0,1\}$, $\alpha,\beta\in\{-1,0,1\}$. Here, α (resp. β) corresponds to moving the first (resp. second) machine head left one square (-1), right one square (1), or leaving it still (0).

If a machine is in state q_i reading position ξ_i on tape i $(i=1,2)$ then we say it occurs in configuration $(\xi_1,\xi_2;q_i)$.

Theorem 1.50 (Minsky)**.**
For each partial recursive function $f:\mathbb{N}\to\mathbb{N}$ there is a corresponding Minsky machine M_f such that, beginning in configuration $(2^k,0;q_1)$, M_f reaches configuration $(2^{f(k)},0;q_0)$ if $f(k)$ is defined (the first such time it reaches state q_0), and never reaches state q_0 if $f(k)$ is undefined. That is, M_f partially computes $2^{f(k)}$.

One can find a proof of this in [7, §15 Theorem 2].

Definition 1.51.
A Minsky machine M is said to *cycle* on an integer k if, beginning in configuration $(2^k,0;q_1)$, it reaches some configuration $(\xi_1,\xi_2;q_i)$ and then later reaches configuration $(\xi_1+v_1,\xi_2+v_2;q_i)$ (where $v_j\geq 0$), and in this transition never touches the left end of either tape.

Definition 1.52.
Two disjoint r.e. sets A, B are called *recursively inseparable* if there is no recursive function $g : \mathbb{N} \to \mathbb{N}$ with $g(x) = 0$ for all $x \in A$ and $g(x) = 1$ for all $x \in B$. That is, seeing g as the characteristic function of some recursive set C, we have that $B \subseteq C$ and $A \cap C = \emptyset$.

Lemma 1.53.
Let $f : \mathbb{N} \to \mathbb{N}$ be a partial recursive function which is not total. Let X be the domain of definition of f. Let Y be the set of k for which $f(k)$ is undefined and for which M_f does not cycle. Then X and Y are recursively inseparable.

Proof. We proceed by contradiction. So let C be a recursive set such that $X \subseteq C$ and $C \cap Y = \emptyset$. We use this to give an algorithm to test membership in X as follows:
Given $n \in \mathbb{N}$, we test whether or not n lies in C. If $n \notin C$ then clearly $n \notin X$. Conversely, if $n \in C$, then, starting in configuration $(2^k, 0; q_1)$, we have that M_f *either* halts *or* cycles; we simply run M_f until the first of these occur (and we can clearly observe each of these occurring). If M_f cycles, then $n \notin X$; if M_f halts, then $n \in X$. Thus the set X is recursive, which is a contradiction as X was taken to be nonrecursive. □

2. GROUP THEORY PRELIMINARIES

Parts §2.1-2.5 of this section are a summary of [8, §1-5]. We will introduce the main definitions and theorems here, but mostly without proof. You are strongly encouraged to read [8, §1-5], or at least have it on-hand when going through this section. [8] is completely self-contained (mathematically).

2.1. **Free groups.**

We begin with some basic notation and definitions.

Definition 2.1. Let $X \subseteq G$. The subgroup $\langle X \rangle$ *generated by* X is $\bigcap H$, where this intersection is over all H with $X \subseteq H \leq G$ (here $H \leq G$ denotes that H is a subgroup of G). It is the smallest subgroup of G containing X. We will write $\langle x_1, \ldots, x_n \rangle$, $\langle x_1, x_2, \ldots \rangle$, $\langle X, Y \rangle$, $\langle X_i : i \in I \rangle$, etc as variants on this notation.

Definition 2.2. A group G is *finitely generated* (f.g.) if we have some $n \in \mathbb{N}$ and elements $g_1, \ldots, g_n$ such that $G = \langle g_1, \ldots, g_n \rangle$. (For $n = 0$ we regard $\{g_1, \ldots, g_n\}$ as $\emptyset$ and $\langle \emptyset \rangle = \{e\}$.)

Definition 2.3. The *normal closure* $\langle\langle X \rangle\rangle^G$ of X in G is $\bigcap N$ over all N with $X \subseteq N \trianglelefteq G$. It is the smallest normal subgroup of G containing X.

Proposition 2.4. *If $X \subseteq G$ then the elements of $\langle\langle X \rangle\rangle^G$ are*

$$g_1 x_1^{n_1} g_1^{-1} g_2 x_2^{n_2} g_2^{-1} \ldots g_k x_k^{n_k} g_k^{-1}$$

for $k \in \mathbb{N}$, $n_1, \ldots, n_k \in \mathbb{Z}$, $x_1, \ldots, x_k \in X$ and $g_1, \ldots, g_k \in G$ (but not necessarily distinct).

Proof. This collection of elements does form a subgroup containing X and it is normal in G: on conjugation by $g \in G$ one just changes the above expression by writing g in front of and g^{-1} behind every subexpression $g_i x_i^{n_i} g_i^{-1}$. But it is also in $\langle\langle X \rangle\rangle^G$. □

Note. This is a relative notion, in that if $X \subseteq H \leq G$ then $\langle\langle X \rangle\rangle^H$ need not equal $\langle\langle X \rangle\rangle^G$; indeed we have $\langle X \rangle \leq \langle\langle X \rangle\rangle^H \leq \langle\langle X \rangle\rangle^G$ because we have more to conjugate with in G than in H. Hence the ^{G} superscript: for the abstract closure $\langle X \rangle$ one could write $\langle X \rangle^G$ but $\langle X \rangle^H = \langle X \rangle^G$ anyway.

We need to understand free groups, as they form the backbone of most of our later constructions.

Definition 2.5.
We define the *free group on* X, $F(X)$, as the set of reduced words X^{red}, where for $w_1, w_2 \in X^{\text{red}}$ we set $w_1 \cdot w_2$ to be the concatenation of these two words, cancelling any inverse pairs that occur.

Note. This is a bad way to define free groups, as the proof of associativity is fiddly because of the cancelling. However it is clear that the empty word ϵ is the identity and inverses of words are formed by writing the word backwards and changing all letters to their inverses. There are other definitions, all of which end up equivalent to this one, so this is what we will use as it is intuitively the easiest to understand.

Theorem 2.6 (Universal property of free groups [8, 2.5])**.**
A free group $F(X)$ has the property that any map $f : X \to H$ (where H is any group) extends uniquely to a homomorphism $f^* : F(X) \to H$.

Moreover if G is a group with generating set S having this universal property then G is free on S*, namely there is an isomorphism from G to some free group $F(X)$ which sends S to X. (We might also say that S* freely generates G*, or is a* free basis *for G.)*

Proposition 2.7 ([8, 2.6])**.**
If $F(X)$ and $F(Y)$ are free groups on X and on Y then $F(X) \cong F(Y) \iff$ the sets X and Y are in bijection.

Consequently we can unambiguously define F_n to be (the) *free group of rank n* where $F_n \cong F(X)$ if and only if $|X| = n$, and also F_∞ for $F(\mathbb{N})$. Then $F_0 = \{e\}$, $F_1 = \mathbb{Z}$, but if $a \neq b \in X$ then $ab \neq ba$ in $F(X)$, so $F(X)$ is non abelian (and of course infinite) whenever $|X| \geq 2$.

Theorem 2.8 ([8, 2.7])**.**
Every (finitely generated) group is a quotient of a (finitely generated) free group.

Definition 2.9. Let $\{G_\lambda : \lambda \in \Lambda\}$ be an indexed family of groups. A *reduced sequence* in $\{G_\lambda\}$ is a finite sequence $g_1 \dots g_r$ of elements in the disjoint union $\bigsqcup_{\lambda\in\Lambda} G_\lambda$ such that:
No g_j is equal to the identity in any G_λ and no successive g_j, g_{j+1} are in the same G_λ.
Let $\mathcal{A}$ be the set of all finite sequences in $\bigsqcup G_\lambda$ and let $\mathcal{R}$ be the set of all reduced sequences, again both including ϵ. We say that a reduced sequence is *P-reduced* (i.e. reduced in the sense of a free Product) if we need to distinguish between different notions of being reduced.

Definition 2.10. The

Definition 2.11. free product $*_{\lambda\in\Lambda} G_\lambda$ of the groups $\{G_\lambda : \lambda \in \Lambda\}$ is the set of P-reduced sequences in $\{G_\lambda\}$, with multiplication defined by concatenation, followed by cancellation in the sequence

Definition 2.12 (Normal form theorem for free products [8, 2.15])**.**
For groups H_1, H_2, any element $g \in H_1 * H_2$ can be written uniquely (the *normal form* for g) as $g_1 \dots g_n$ where $n \geq 1$ (or $n = 0$, but only if g is the identity when we write ϵ), each $g_i \in H_1 \cup H_2$ but no g_i is the identity in H_1 or in H_2, and successive g_i, g_{i+1} are taken from alternating factors.

Lemma 2.13 ([8, 2.19])**.**
Take the group F_2, free on a, b say. Now define elements $a_n := b^n a b^{-n} \in F_2$ and the subset $S = \{a_n \mid n \in \mathbb{N}\}$. Then S freely generates *the subgroup $\langle a_n : n \in \mathbb{N}\rangle \cong F_\infty$ of F_2.*

Proof. Let $*_{n\in\mathbb{N}}\langle x_n\rangle$ be the countably infinite free product of the infinite cyclic groups $\langle x_n\rangle$. As mentioned, this is also the group F_∞ and is free on $\{x_n : n \in \mathbb{N}\}$. Now use (2.6) to obtain a homomorphism $\theta : F_\infty \to F_2$ with image $\theta(F_\infty) = \langle S\rangle$ by extending the map that send(s) x_n to a_n.
So given a P-reduced sequence $g_1 \dots g_k \in F_\infty$ which is non trivial, thus $k \geq 1$, we have

$$\theta(g_1 \dots g_k) = a_{n_1}^{p_1} \dots a_{n_k}^{p_k}$$

where for $1 \leq i \leq k$ we have $g_i = x_{n_i}^{p_i} \in \langle x_{n_i}\rangle \setminus \{e\}$, so $p_1, \dots, p_k \in \mathbb{Z} \setminus \{0\}$ and $n_1, \dots, n_k \in \mathbb{N}$. But successive n_i are distinct because $g_1 \dots g_k$ is P-reduced. However the right hand side is the element

$$b^{n_1} a^{p_1} b^{n_2 - n_1} a^{p_2} \dots b^{n_k - n_{k-1}} a^{p_k} b^{-n_k}$$

which we see is freely reduced in F_2 and not empty, so is not the identity. Thus θ is injective. □

Note. In place of S, one could use any subset T of S with exactly the same proof to conclude that T freely generates the subgroup $\langle T\rangle$ of F_2 and thus $\langle T\rangle \cong F_{|T|}$ by (2.7).

Theorem 2.14 (Nielsen-Schreier Theorem [8, 3.7])**.**
A subgroup of a free group is free.

If X is finite and H has finite index in $F(X)$ then we can easily determine the rank of H.

Theorem 2.15 (Nielsen-Schreier index formula [8, 3.8])**.**
If H has index i in the free group F_n then H is a free group of rank $i(n-1)+1$.

Lemma 2.16 ([8, 10.3])**.**
Let S be a free basis for a free group G. Let $P, Q \subseteq S$ be subsets. Then
1. *$\langle P\rangle$ is free with basis P.*
2. *For any $s \in S$, we have $s \in \langle P\rangle$ iff $s \in P$.*
3. *$\langle P\rangle \cap \langle Q\rangle = \langle P \cap Q\rangle$.*

Proof.
1. Apply the normal form theorem for free products (2.12), as done in (2.13).
2. If $s \neq P$ but $s \in \langle P\rangle$ then S is not a free basis, as otherwise we could express s as a product $s_1 \cdots s_k$ of other elements of S and their inverses, and so s and $s_1 \cdots s_k$ would be two ways of writing the same element of G, contradicting the unique normal form for free products (2.12).
3. If $g \in \langle P \cap Q\rangle$ the g can be written as a product of elements which lie both in P and in Q, and so $g \in \langle P\rangle \cap \langle Q\rangle$. Conversely, if $g \in \langle P\rangle \cap \langle Q\rangle$, then g can be written as a product $p_1 \cdots p_k$ of elements of P and their inverses, as well as a product $q_1 \cdots q_l$ of elements of Q and their inverses. But by the normal form theorem for free products (2.12), as both $P, Q \subseteq S$, we must have that these two (freely reduced) words are the same, and so $g \in \langle P \cap Q\rangle$. □

2.2. **Group presentations.**

On being given any group G and a generating set $S = \{g_i : i \in I\}$ for G, we have by (2.8) that $G \cong F(X)/N$ where X is some set in bijection with S and $N \trianglelefteq F(X)$.

Definition 2.17.
A *presentation* $P = \langle X|R\rangle$ for a group G is a set X and a subset R of reduced words on $X \cup X^{-1}$, which are thus elements of $F(X)$, such that $G \cong F(X)/\langle\langle R\rangle\rangle^{F(X)}$. The elements of X are called *generators*, and the elements of R are *relators*.

Hence every group has a presentation by the comment above. Conversely, on being given some presentation $P = \langle X|R\rangle$ we will write $\overline{P}$ for the group thus defined by this presentation. Furthermore given a word w in $F(X)$, we write $\overline{w} \in \overline{P}$ for the corresponding element of the group $F(X)/\langle\langle R\rangle\rangle^{F(X)}$ under the natural projection map $\pi = \pi_P : F(X) \twoheadrightarrow F(X)/\langle\langle R\rangle\rangle^{F(X)}$. It should be clear that for any group element $g \in \overline{P}$, there will be a reduced word in X^* such that $\overline{w} = g$ in $\overline{P}$. Also we have $\overline{vw} = \overline{v} \cdot \overline{w}$ for $v, w \in F(X)$.

Moreover if $P = \langle X|R\rangle$ and $w \in F(X)$ then by (2.4) we have that $\overline{w} = e$ in $\overline{P}$ if and only if we have $m \in \mathbb{N}$ such that

$$w = \prod_{i=1}^{m} v_i r_{j_i}^{\pm 1} v_i^{-1} \text{ in } F(X)$$

where $r_{j_i} \in R$ and all of w, v_i, r_{j_i} are reduced words in X^* (even though the expression on the right hand side need not be reduced).

Theorem 2.18 (von Dyck: "a quotient = more relators" [8, 4.2])**.**
If $P = \langle X|R\rangle$ is a group presentation with $G = \overline{P}$ then a group H is a quotient of $G \iff$ there is a subset S of $F(X)$ such that $H = \overline{\langle X|R \cup S\rangle}$.

Definition 2.19 ([8, 4.3])**.**
A *finite presentation* is a presentation $P = \langle X|R\rangle$ where both $X = \{x_1, \ldots, x_n\}$ and $R = \{r_1, \ldots, r_m\}$ are finite sets. We write $\langle x_1, \ldots, x_n | r_1, \ldots, r_m\rangle$ for P.
A group G is *finitely presented* (f.p.) if there exists *some* finite presentation $\langle x_1, \ldots, x_n | r_1, \ldots, r_m\rangle$ defining G, i.e. $G = \overline{\langle x_1, \ldots, x_n | r_1, \ldots, r_m\rangle}$ ("finitely many generators and finitely many relators"). If so then G will be generated by $\{\overline{x}_1, \ldots, \overline{x}_n\}$ thus "*f.p. implies f.g.*".

Note. A group G will be defined by many different presentations. Even if G is f.p., it does not mean that all of these will themselves be finite presentations.

Proposition 2.20 (B. Neumann [8, 4.5])**.**
If a group G is defined by the finite presentation $P_1 = \langle X \,|\, r_1, \ldots, r_m\rangle$ for $X = \{x_1, \ldots, x_n\}$ and also by the presentation $P_2 = \langle Y \,|\, s_1, s_2, \ldots\rangle$ for $Y = \{y_1, \ldots, y_k\}$ with finitely many generators but infinitely many relators then there is l such that $\langle\langle s_1, s_2, \ldots\rangle\rangle^{F(Y)}$ is equal to $\langle\langle s_1, \ldots, s_l\rangle\rangle^{F(Y)}$, so that all but finitely many relators in the second presentation are redundant.

Proposition 2.21 ([8, 4.7])**.**
If $G = \overline{\langle X \,|\, R\rangle}$ and $H = \overline{\langle Y \,|\, S\rangle}$ for disjoint sets X and Y then $G * H = \overline{\langle X \cup Y \,|\, R \cup S\rangle}$.

Definition 2.22.
If X is a finite set, and $R \subseteq F(X)$ is an r.e. subset of $F(X)$ (the halting set of a Turing machine on $X \cup X^{-1}$), then we say $\langle X|R\rangle$ is a *recursive presentation*; groups with such a presentation are called *recursively presented*.

We will often consider the 'data' of a recursive presentation to be
1. The finite generating set X, and
2. A Turing machine T whose halting set is R; $\Omega(T) = R$.

This way, a recursive presentation can be described with only finite data. This is important later on when dealing with algorithmic properties of recursively presented groups.

Observe that being finitely presented implies being recursively presented, as all finite sets are r.e. Moreover, one can have a recursive presentation of a finitely presented group. This is a very important distinction; having a recursive presentation of a finitely presented group does not always allow us to algorithmically extract a finite presentation from it.

The reason we write 'recursive presentation' rather that 'recursively enumerable presentation' is because if $P = \langle X|R\rangle$ is a recursive presentation of a group, then $\overline{P}$ has another recursive presentation where the relating set is recursive.

Lemma 2.23 ([8, 8.2])**.**
Let $P = \langle X|R\rangle$ be a recursive presentation of a group. Then there exists a recursive presentation $P' = \langle X'|R'\rangle$ with R' recursive such that $\overline{P'} \cong \overline{P}$.

Proof. First, if R is finite, then it is recursive, so just take the original presentation P.
So let R be an infinite r.e. set, with enumeration $r_1, r_2, \ldots$. Take $X' := X \cup \{t\}$ and $R' := \{t^i r_i \mid r_i \in R\} \cup \{t\}$. It is clear that $\overline{P'} \cong \overline{P}$, as the generator $\bar{t}$ is trivial in $\overline{P'}$, and the other relators are then unchanged at the group level. To see that the set R' is recursive in $F(X \cup \{t\})$, take any word $w \in F(X \cup \{t\})$. If this is not of the form $t^i z$ for some $z \in F(X)$, then $w \notin R'$. Otherwise w is of the form $t^i z$ with $z \in F(X)$. So start enumerating the elements of R. Eventually, as R is infinite, we will reach the word r_i; then $r_i = z$ (in $F(X)$) iff $w \in R'$. This process always halts, so R' is recursive. □

Note that this proof is merely existential. We cannot necessarily construct (a Turing machine enumerating) R' from (a Turing machine enumerating) an arbitrary r.e. set $R \subseteq F(X)$, as we would need *a priori* knowledge of whether R was finite or infinite.

2.3. **Semigroups.**

Semigroups will give us our first examples of algorithmically unsolvable problems in algebra, so we introduce them here.

Definition 2.24.
A *semigroup* S is a set with an associative binary operation and is *commutative* if $ab = ba$ always. A *subsemigroup* is a subset T of S that is closed under multiplication ($a, b \in T$ implies $ab \in T$).
A *semigroup homomorphism* f is a function from one semigroup to another such that $f(ab) = f(a)f(b)$ always. A *semigroup isomorphism* is a bijective semigroup homomorphism, whereupon the inverse map f^{-1} is also a semigroup homomorphism.

Example 2.25.

(1) Any set S with the operation $ab = a$ (not commutative).
(2) The natural numbers $\mathbb{N}$ under addition (either with or without 0).
(3) All integers at least 26 under addition, but all integers at least -2 under addition is not an example.
(4) Any subset of integers with the operation $ab = max(a, b)$.
(5) The main example: let X be a set of symbols. Whereas X^+ was the set of all finite words on X including ϵ, let $X^\circ = X^+ \setminus \{\epsilon\}$ be the set of non empty finite words *on X. Both of these are semigroups under the operation of concatenation and are non commutative if $|X| > 1$.*

As there is no cancellation, free semigroups can be defined directly:

Definition 2.26.
The *free semigroup* on a set X *is* the semigroup X° and clearly satisfies the universal property: if T is a semigroup and $f : X \to T$ any function then f can be extended uniquely to a semigroup homomorphism from X° to T. Moreover if a semigroup S has a subset Y with this universal property then S is semigroup isomorphic to Y°.

In analogy with free groups, we see that every semigroup is the image under some semigroup homomorphism of a free semigroup. Also if $|X| = |Y|$ then the free semigroups X° and Y° are semigroup isomorphic. Moreover if $|X| < |Y|$ then there cannot be a surjective semigroup homomorphism from X to Y (one could use word length and note that words of length more than 1 cannot map to words of length 1).

However subsemigroups of free semigroups need *not* be free: if $|X| > 1$ then X° is the natural numbers with 0 removed under addition and $S = \{2, 3, 4, \ldots\}$ is a subsemigroup of X° and so is also commutative. Thus S would need to be free on a single element n, but on sending $n \in S$ to $1 \in X^\circ$ how would we extend this to a semigroup homomorphism from S to X°?

Also we do not have a direct generalisation of normal subgroups, but we say a *congruence* on the semigroup S is an equivalence relation $\sim$ (or indeed a partition) which is compatible with the multiplication (if $a \sim c$ and $b \sim d$ then $ab \sim cd$), so that the induced multiplication on the set E of equivalence classes is well defined and turns E into a semigroup, the *quotient* semigroup written $S/\sim$, or here even $S/\equiv$ to emphasise that $\equiv$ is a special type of equivalence relation.

The following is the only example of a congruence we will need.

Definition 2.27.
As a congruence on a semigroup S is a special type of relation (a relation being just a subset of $S \times S$) and the intersection of congruences on S is also a congruence, we can talk about the congruence *generated by* any $R \subseteq S \times S$ as the intersection of all the congruences containing R.
In particular let X° be the free semigroup on the set X and suppose that R is a relation on X°. If $\equiv$ is the congruence on X° generated by R then we say that $P = \langle X \mid R \rangle$ is a *semigroup presentation* for the quotient semigroup $X^\circ/\equiv$.
We say that P is a *finite semigroup presentation* if both X and R are finite.

Lemma 2.28.
Let $P = \langle X \mid R \rangle$ be a semigroup presentation where $R = \{(x_i, y_i) : i \in I\} \subseteq X^\circ \times X^\circ$ and let $\overline{P}$ denote the quotient semigroup $X^\circ/\equiv$ given by the congruence $\equiv$ generated by R. Then two elements w, w' of X° are equal in $\overline{P}$ if and only if there is a finite sequence $w = w_0, w_1, \ldots, w_m = w'$ of words $w_j \in X^\circ$ such that w_{j+1} differs from w_j by the replacement of some subword x of w_j with y of w_{j+1}, where either (x, y) or (y, x) is in R.

Proof.
($\Leftarrow$) If (x, y) or (y, x) is in R, so that $x \equiv y$, then for any $u, v \in X^\circ$ we will have $ux \equiv uy$ and $uxv \equiv uyv$, thus we do have $w_j \equiv w_{j+1}$ and $\equiv$ is an equivalence relation.
($\Rightarrow$) If we define the relation $\equiv_R$ on X° by $w \equiv_R w'$ if there is some sequence from w to w' as above then we see that $\equiv_R$ is an equivalence relation on X°, and even a congruence, containing R. As $\equiv$ is the smallest congruence containing R, we will have $w \equiv w' \Rightarrow w \equiv_R w'$. □

2.4. **Amalgamated products and HNN extensions.**

Almost all of our constructions in later sections will be made up entirely of 2 key building blocks: amalgamated products (a way of gluing two groups together along isomorphic subgroups), and HNN extensions (a way of gluing a

group to itself along a pair of its isomorphic subgroups). These constructions will be *indispensable* to us.

Definition 2.29.
Let G, H be groups containing isomorphic subgroups $A \leq G$, $B \leq H$ and let $\varphi : A \to B$ be an isomorphism.
The *free product with amalgamation* is the group

$$(G * H)/\langle\langle a = \varphi(a) \text{ for all } a \in A^{G*H}\rangle\rangle$$

written $G *_\varphi H$. Thus we form the free product $G * H$ and identify A in $G * H$ with its image $\varphi(A) = B$ under φ.

Definition 2.30.
If G is a group and $A, B \leq G$ are isomorphic subgroups of G with $\varphi : A \to B$ some isomorphism between them then the *HNN extension* $G*_\varphi$ is defined by taking some other symbol t (the *stable letter* of the HNN extension), with $\langle t\rangle$ regarded as a copy of $\mathbb{Z}$, and forming the group

$$(G * \langle t\rangle)/\langle\langle tat^{-1} = \varphi(a)\rangle\rangle^{G*\langle t\rangle}.$$

Thus we take the free product $G * \mathbb{Z}$ and then identify A in $G * \mathbb{Z}$ with its image $\varphi(A) = B$ via conjugation by t.
We call G the *base group* of the HNN extension.

Proposition 2.31 ([8, 5.2 and 5.4])**.**
If G and H are both f.g. then so are $G *_\varphi H$ and $G*_\varphi$.
If G and H are both f.p. and A, B are f.g. then $G *_\varphi H$ and $G*_\varphi$ are also f.p.

Definition 2.32.
Given the HNN extension $G*_\varphi$ with $\varphi : A \to B$ an isomorphism and stable letter t, an *H-sequence* (from *H*NN) is a finite sequence of the form $g_0 t^{\epsilon_1} g_1 \dots t^{\epsilon_n} g_n$ where $n \geq 0$, each $g_i \in G$ and each $\epsilon_i \in \{\pm 1\}$. Of course any H-sequence can also be regarded as an element of $G*_\varphi$ by multiplying out the elements.
Choose right transversals T_A and T_B for A and B in G such that they both include e_G. A *normal form* is an H-sequence such that if $\epsilon_i = +1$ then $g_i \in T_A$, if $\epsilon_i = -1$ then $g_i \in T_B$, and there is no subsequence te_Gt^{-1} or $t^{-1}e_Gt$. Let $\mathcal{N}$ be the set of all normal forms.

Note first the slightly different treatment of the identity here from the free product case: any g_i is allowed to be the identity e_G in G, but $n \geq 0$ means the empty word does not occur. Instead $n = 0$ gives us an element $g_0 \in G$, so taking $n = 0$ and $g_0 = e_G$ gives us the identity.

Note also that any element of $G*_\varphi$ can be expressed as an H-sequence (think of the normal form for the free product $G * \langle t\rangle$ and pad it out with various e_G's between the relevant powers of t). But as $ta = \varphi(a)t$ and $t^{-1}b = \varphi^{-1}(b)t^{-1}$, we can move a's (resp. b's) to the left of t (resp. t^{-1}) in $G*_\varphi$. This means that *every* H-sequence can be put into normal form without changing the element in $G*_\varphi$ which it represents by working from right to left: for instance if we see a subsequence tg_i where $g_i \notin T_A$ then set $g_i = a_i\gamma_i$ for some $a_i \in A$ and $\gamma_i \in T_A$, so that we can replace $\dots g_{i-1}tg_i \dots = \dots g_{i-1}ta_i\gamma_i \dots$ with $\dots (g_{i-1}\varphi(a))t\gamma_i \dots$ in our sequence.

Theorem 2.33 (Normal form for HNN extensions [8, 5.6])**.**
Every element in $G*_\varphi$ has a unique normal form.

Corollary 2.34 ("Base group embeds" [8, 5.7])**.**
In any HNN extension $G*_\varphi$ the base group G embeds homomorphically in $G*_\varphi$ by sending $g \in G$ to the H-sequence g.

Proof. If $g \neq e_G$ then the H-sequence g is in normal form, and is not the normal form e_G which is the identity in $G*_\varphi$. □

In fact for many uses we do not need unique representative forms for each element in an HNN extension. We just need an easily verified criterion telling us that our form is *not* the identity.

Definition 2.35.
If $G*_\varphi$ is an HNN extension then a *pinch* of an H-sequence $g_0 t^{\epsilon_1} \dots t^{\epsilon_n} g_n$ is a subsequence of the form $t g_i t^{-1}$ for $g_i \in A$, or $t^{-1} g_i t$ for $g_i \in B$.
An H-sequence is *reduced* (or H-reduced if we need to distinguish) if it contains no pinch, thus in particular all normal sequences are reduced.

Corollary 2.36 ("No pinch", Britton's Lemma [8, 5.9])**.**
If the H-sequence $g_0 t^{\epsilon_1} \dots t^{\epsilon_n} g_n$ is reduced and $n \geq 1$ then it is not the identity in $G*_\varphi$; indeed it is not even in G.

Proof. As mentioned above, we can change this reduced sequence into a normal form without altering the element in $G*_\varphi$ which it represents. The key point is that when doing this for a reduced form, no $t^{\pm\epsilon}$ cancel so we end up with the normal form $g_0' t^{\epsilon_1} \dots t^{\epsilon_n} g_n'$ for $n \geq 1$. Moreover if $g_0 t^{\epsilon_1} \dots t^{\epsilon_n} g_n = g \in G$ then $g_0 t^{\epsilon_1} \dots t^{\epsilon_n} g_n g^{-1}$ is also reduced. □

Here we have similar results in that the factor groups embed and there are equivalent notions of normal and reduced forms. As we will not need unique representatives in this course, we will not here pursue normal forms for amalgamated free products but will prove the other results by reducing to the HNN extension case above.

If $G *_\varphi H$ is a free product with amalgamation where $\varphi : A \to B$ is an isomorphism for $A \leq G$ and $B \leq H$, we say that a P-reduced (meaning free product reduced) sequence $c_1 \dots c_n \in G * H$ is *A-reduced* (for "amalgamated reduced") if whenever $n > 1$ we have that no c_i is in A or B.

We can turn any P-reduced sequence in $G * H$ into an A-reduced sequence by "absorbing" elements of A or B using φ, without changing the element it represents in $G *_\varphi H$. For instance if we see the subsequence $\dots g_{i-1} b_i g_{i+1} \dots$ in our P-reduced sequence, where $g_{i-1}, g_{i+1} \in G$ and $b_i \in B$, then we can replace b_i with $\varphi^{-1}(b_i) \in A$ and then group together $(g_{i-1}\varphi^{-1}(b_i) g_{i+1})$ as an element of G, thus reducing the length of the sequence.

Corollary 2.37 ("Factor groups embed" [8, 5.14])**.**
If $c_1 \dots c_n$ is A-reduced and $n \geq 1$ then the element it represents is not equal to the identity in $G *_\varphi H$. In particular G and H embed in $G *_\varphi H$ by taking $n = 1$.

Given any group $G = G_0$, one is at liberty to create a sequence of HNN extensions in turn, by forming $G_1 = G*_\varphi$, $G_2 = G_1*_{\varphi'}$ etc. However an especially useful version of this is when we form multiple HNN extensions of a group G "simultaneously" (though here we will only ever need to do this for finitely many extensions).

Definition 2.38.
Suppose that G is a group and we have isomorphisms $\varphi_i : A_i \to B_i$ for $i = 1, \ldots, n$, where all A_i and B_i are subgroups of G. Then the *multiple HNN extension* $G*_{\varphi_1,\ldots,\varphi_n}$ with *base group* G and *stable letters* $t_1, \ldots, t_n$, where all $\langle t_i \rangle$ are regarded as copies of $\mathbb{Z}$, is the group

$$(G*\langle t_1\rangle*\ldots*\langle t_n\rangle)/\langle\langle t_1 a_1 t_1^{-1} = \varphi_1(a_1) : a_1 \in A_1, \ldots, t_n a_n t_n^{-1} = \varphi_n(a_n) : a_n \in A_n\rangle\rangle.$$

Thus we form the free product G with a copy of the rank n free group F_n and then identify each A_i with its image $\varphi_i(A_i)$ via conjugation by t_i.

Lemma 2.39 ([8, 5.16])**.**
The multiple HNN extension $G*_{\varphi_1,\ldots,\varphi_n}$ as defined above can be regarded as the final group $G_n = G_{n-1}*_{\varphi_n}$ obtained by the sequence of (single) HNN extensions $G_i = G_{i-1}*_{\varphi_i}$ where we set the base $G = G_0$ and regard $A_i, B_i \leq G_0$ as isomorphic subgroups of G_{i-1}.

We now have two equivalent results for multiple HNN extensions which have already been established for single HNN extensions.

Definition 2.40.
Suppose we have a multiple HNN extension $G*_{\varphi_1,\ldots,\varphi_n}$ with base group G, isomorphisms $\varphi_i : A_i \to B_i$ and stable letters $t_1, \ldots, t_n$. An *MH-sequence* (for multiple HNN extension) is is a finite sequence of the form $g_0 t_{j_1}^{\epsilon_1} g_1 \ldots t_{j_n}^{\epsilon_n} g_n$ where $n \geq 0$, each $g_i \in G$, each $\epsilon_i \in \{\pm 1\}$ and each j_i comes from $\{1, \ldots, n\}$ (again we can think of it as an element of $G*_{\varphi_1,\ldots,\varphi_n}$).
A *t_i-pinch* of an MH-sequence is a subsequence of the form $t_i g t_i^{-1}$ for $g \in A_i$, or $t_i^{-1} g t_i$ for $g \in B_i$.
An MH-sequence is *reduced* if it contains no t_i-pinch for any $i \in \{1, \ldots, n\}$. Any element in $G*_{\varphi_1,\ldots,\varphi_n}$ can be represented by a reduced MH-sequence (replace all pinches until there are no more).

Theorem 2.41 (Multiple Britton's Lemma [8, 5.18])**.**
If the MH-sequence $g_0 t_{j_1}^{\epsilon_1} \ldots t_{j_n}^{\epsilon_n} g_n$ contains no pinch and $n \geq 1$ then it is not the identity in $G*_{\varphi_1,\ldots,\varphi_n}$, nor is it even in G.

Proof. We do this by induction on n. Suppose true up to $n-1$ and that $g_0 t_{j_1}^{\epsilon_1} \ldots t_{j_n}^{\epsilon_n} g_n$ contains no t_i-pinch, but is in G. Regard $G*_{\varphi_1,\ldots,\varphi_n}$ as the single HNN extension with base G_{n-1} and stable letter t_n, as in (2.39). Then on grouping together as subsequences the successive elements that are not equal to t_n or t_n^{-1}, we have that the resulting H-sequence so formed must contain a pinch $t_n a_n t_n^{-1}$ for $a_n \in A_n$ or $t_n^{-1} b_n t_n$ for $b_n \in B_n$.

But this a_n or b_n was obtained by multiplying out a subsequence of our MH-sequence, thus we regard this subsequence as an MH-sequence itself, but one with no appearances of $t_n^{\pm 1}$ so that it is moreover an MH-sequence with respect to the multiple HNN extension $G_{\varphi_1,\ldots,\varphi_{n-1}}$. Now this clearly contains no t_i-pinches as the original sequence did not, so by induction it cannot lie in G, and thus certainly not in A_n or in B_n. □

2.5. **Good subgroup theorems for HNN extensions.**

We give a criterion for when a subgroup of an HNN extension naturally inherits a description as an HNN extension itself. We will appeal to this result (2.45) many many times later on, to show that our constructions have certain properties.

Definition 2.42.
Let $G*_\varphi$ be an HNN extension with base G, stable letter t and A, B subgroups of G where $\varphi : A \to B$ is an isomorphism. We say that a subgroup H of the base G is a *good* subgroup (with respect to this HNN extension) if $\varphi(H \cap A) = H \cap B$.

Lemma 2.43 ([8, 5.12])**.**
If $G*_\varphi$ is as defined in (2.42) *and H is a good subgroup of G then the subgroup $\langle H, t\rangle$ of the HNN extension $G*_\varphi$ is itself naturally an HNN extension $H*_\psi$ with base H, stable letter t, and where $H \cap A, H \cap B$ are subgroups of H with ψ, defined as the restriction of φ to $H \cap A$, an isomorphism from $H \cap A$ to $H \cap B$. Moreover $\langle H, t\rangle \cap G = H$.*

Proof. The condition on H does mean that ψ is an isomorphism. Form the abstract HNN extension $H*_\psi$ with stable letter s and let $\theta : H*_\psi \to G*_\varphi$ be the homomorphism sending s to t and $h \in H$ to its image h in $G*_\varphi$. This is well defined because if $a \in H \cap A$ then $sas^{-1}(\psi(a))^{-1}$ maps to $tat^{-1}(\varphi(a))^{-1} = e$ as $\psi(a) = \varphi(a) \in B$.

The image of θ is clearly $\langle H, t\rangle$ and if $h_0 s^{\epsilon_1} \dots s^{\epsilon_n} h_n$ is an element of $H*_\psi$ represented by a reduced sequence then it maps to $h_0 t^{\epsilon_1} \dots t^{\epsilon_n} h_n$ which also contains no pinch. Thus θ is injective and further $h_0 s^{\epsilon_1} \dots s^{\epsilon_n} h_n$ does not map into G if $n \geq 1$. □

We now have, in exact analogy with (2.42) and (2.43):

Definition 2.44.
Let $G*_{\varphi_1,\dots,\varphi_n}$ be an HNN extension with base G, stable letters $t_1, \dots, t_n$ and A_i, B_i subgroups of G where $\varphi_i : A_i \to B_i$ is an isomorphism. We say that a subgroup H of the base G is a *good* subgroup (with respect to this multiple HNN extension) if $\varphi_i(H \cap A_i) = H \cap B_i$ for each $i = 1, \dots, n$.

Lemma 2.45 ([8, 5.20])**.**
If $G*_{\varphi_1,\dots,\varphi_n}$ is as defined in (2.44) *and H is a good subgroup of G then the subgroup $\langle H, t_1, \dots, t_n\rangle$ of the multiple HNN extension $G*_{\varphi_1,\dots,\varphi_n}$ is itself naturally an HNN extension $H*_{\psi_1,\dots,\psi_n}$ with base H, stable letters t_i, and where $H \cap A_i, H \cap B_i$ are subgroups of H with ψ_i, the restriction of φ_i to $H \cap A_i$, an isomorphism from $H \cap A_i$ to $H \cap B_i$. Moreover $\langle H, t_1, \dots, t_n\rangle \cap G = H$.*

Proof. Again form the abstract HNN extension $H*_{\psi_1,\dots,\psi_n}$. Now use (2.41) instead of (2.36). □

We now study good subgroups of HNN extensions which are normal. This will give us an understanding of when a *quotient* of an HNN extension naturally inherits a description as an HNN extension.

Definition 2.46.
Let $H := G*_{\varphi_1,\dots,\varphi_n}$ be an HNN extension of G with stable letters $t_1, \dots, t_n$. Let $K \trianglelefteq G$ be a good subgroup of G with respect to the HNN extension H. Let $\tilde{\varphi}_i : A_i/(K \cap A_i) \to B_i/(K \cap B_i)$ be the induced isomorphism for each $1 \leq i \leq n$. Define the following HNN extension with stable letters $\tilde{t}_1, \dots, \tilde{t}_n$:

$$H_K := (G/K)*_{\tilde{\varphi}_1,\dots,\tilde{\varphi}_n}$$

There is a surjective homomorphism

$$\phi_K : H \twoheadrightarrow H_K$$

which sends $g \mapsto gK$ for all $g \in G$, and $t_i \mapsto \tilde{t}_i$ for all $1 \leq i \leq n$.

Lemma 2.47.
Let G be a group, and $H := G*_{\varphi_1,\ldots,\varphi_n}$ an HNN extension of G with stable letters $t_1,\ldots,t_n$. Let $K \trianglelefteq G$. Then K is a good subgroup of G with respect to the HNN extension H if and only if $\langle\langle K\rangle\rangle^H \cap G = K$ in H.

Proof.
$\Leftarrow$: Assume that $\langle\langle K\rangle\rangle^H \cap G = K$ in H. Take $1 \leq i \leq n$ and suppose $x \in A_i \cap K$. We know that $\varphi_i(x) \in B_i$: we need to verify that $\varphi_i(x) \in K$. However, it is immediate that $\varphi_i(x) = t_i^{-1}xt_i \in \langle\langle K\rangle\rangle^H$, and thus that $\varphi_i(x) \in \langle\langle K\rangle\rangle^H \cap B_i = \langle\langle K\rangle\rangle^H \cap G \cap B_i = K \cap B_i$; it follows that $\varphi_i(A_i \cap K) \subseteq B_i \cap K$. The inclusion $\varphi_i(A_i \cap K) \supseteq B_i \cap K$ can be proved in a similar fashion.
$\Rightarrow$: Suppose K is a good subgroup of G with respect to the HNN extension H, and take ϕ_K as in (2.46). Then it is clear that $K \leq \langle\langle K\rangle\rangle^H \cap G \leq \mathrm{Ker}(\phi_K) \cap G \leq K$; the last inequality here is a consequence of (2.41). □

Lemma 2.48.
Let H, K and ϕ_K be as in (2.46). *Then* $\mathrm{Ker}(\phi_K) = \langle\langle K\rangle\rangle^H$.

Proof. The containment $\langle\langle K\rangle\rangle^H \subseteq \mathrm{Ker}(\phi_K)$ is immediate.
Let $x \in \mathrm{Ker}(\phi_K)$. We induct on the total number of occurrences of t_i or t_i^{-1} over all the i's, where $1 \leq i \leq n$, in the normal form of x in H: if x has none, then $x \in K$.
Assume that for some i, either t_i or t_i^{-1} appears at least once in x. By (2.41) $\phi_K(x)$ has a subword of the form $\tilde{t}_i^{-1}a\tilde{t}_i$ where $a \in A_i/(A_i \cap K)$ or $\tilde{t}_i b \tilde{t}_i^{-1}$ where $b \in B_i/(B \cap K)$. Thus x has a subword of the form $t_i^{-1}a't_i$ where $a' \in A_iK$ or $t_i b' t^{-1}$ where $b \in B_iK$. Without loss of generality, we assume the former.
This subword $t_i^{-1}a't$ is of the form $t_i^{-1}akt_i$, where $a \in A_i$ and $k \in K$. But $t_i^{-1}at_i = b \in B$, for some $b \in B$. We can therefore write x as $\lambda_1 t_i^{-1}akt_i\lambda_2 = \lambda_1 bt_i^{-1}kt_i\lambda_2$. Observe that $t_i^{-1}kt_i \in \langle\langle K\rangle\rangle^H$, and thus that $t_i^{-1}kt_i\lambda_2 = \lambda_2 y$ where $y \in \langle\langle K\rangle\rangle^H$. We can therefore rewrite $x = \lambda_1 b\lambda_2 y$; from this we see that $\lambda_1 b\lambda_2 \in \mathrm{Ker}(\phi_K)$. By induction, we have that $\lambda_1 b\lambda_2 \in \langle\langle K\rangle\rangle^H$. This tells us that $x \in \langle\langle K\rangle\rangle^H$. □

Corollary 2.49.
Let H, K and H_K be as in (2.46). *Then ϕ_K induces an isomorphism*

$$\tilde{\phi}_K \ : \ H/\langle\langle K\rangle\rangle^H \xrightarrow{\cong} H_K.$$

2.6. **The word problem.**

In this section, we assume that all group presentations with countably many generators draw their generating set from the infinite well-ordered set $\mathcal{A} = \{x_1, x_2, \ldots\}$ (in the same way that we "standardised" our Turing machines when we encoded them). This is to ensure that we can encode and enumerate presentations.

Definition 2.50.
We define the *word problem* (abbreviated to WP) for a recursive presentation $P = \langle X|R\rangle$ of a group as follows:

$$\text{Given two words } u, v \in F(X), \text{ is } \overline{u} = \overline{v} \text{ in } \overline{P}?$$

This can be rephrased in the following equivalent way:

$$\text{Given a word } w \in F(X), \text{ is } \overline{w} = e \text{ in } \overline{P}?$$

Groups which have a recursive presentation for which there exists an algorithm that, on input of a pair of words $u, v \in F(X)$, decides if $\overline{u} = \overline{v}$ in $\overline{P}$, are said to have *solvable word problem* (or *soluble word problem*, or *decidable word problem*), abbreviated to SWP. If there is no such presentation for which such an algorithm exists, then we say the group has *unsolvable word problem* (or *insoluble word problem*, or *undecidable word problem*), abbreviated to IWP.

So saying G has SWP is equivalent to saying that, for some recursive presentation P of G, the set of trivial words in P is recursive.

Lemma 2.51.
Let $P = \langle X|R\rangle$ be a recursive presentation of a group. Then the set of trivial words $\{w \in F(X) \mid \overline{w} = e \text{ in } \overline{P}\}$ is r.e. This is uniform *over all recursive presentations. i.e., there is* one *Turing machine which takes as input pairs (P, w) where P is a recursive presentation and w is a word on the generators of P, and halts iff $\overline{w} = e$ in $\overline{P}$.*

Proof. This follows from (2.4) and the comment before (2.18), since we can form a recursive enumeration of all words of the form $u_1 r_1^{\pm 1} u_1^{-1} \dots u_n r_n^{\pm 1} u_n^{-1}$, as well as their free reductions. Then w will appear in this enumeration iff $\overline{w} = e$ in $\overline{P}$. □

Corollary 2.52.
Let G be a recursively presented group. Then G has solvable word problem iff it has a recursive presentation $P = \langle X|R\rangle$ whose set of non-trivial words $\{w \in F(X) \mid \overline{w} \neq e \text{ in } \overline{P}\}$ is r.e.

It is now immediate that we can enumerate all words equivalent to a given word from a finite presentation of a group.

Corollary 2.53.
Let $P = \langle X|R\rangle$ be an infinite recursive presentation, and $w \in F(X)$. Then the words in $F(X)$ equal to $\overline{w}$ in $\overline{P}$ are recursively enumerable.

Proof. Given two words $w, w' \in F(X)$, we have that $\overline{w} = \overline{w'}$ in $\overline{P}$ if and only if $\overline{w'w^{-1}} = e$ in $\overline{P}$. So take an enumeration w_i of all words in $F(X)$, and use 2.51 to check if $\overline{w'w^{-1}} = e$ in $\overline{P}$ (by proceeding along finite diagonals). This will eventually halt precisely on all words w_i which are equivalent to w in $\overline{P}$. □

Lemma 2.54.
Suppose P_1 and P_2 are recursive presentations defining groups, with $\overline{P_2} \leq \overline{P_1}$. Suppose, moreover, that there is an algorithm to solve the word problem in P_1. Then there is an algorithm to solve the word problem in P_2.

Proof. Let $P_i = \langle X_i|R_i\rangle$, $i = 1, 2$. As $\overline{P_2} \leq \overline{P_1}$, there is some map $\varphi : X_2 \to F(X_1)$ which extends to a map $\varphi' : F(X_2) \to F(X_1)$ which in turn extends to an injective homomorphism $\overline{\varphi'} : \overline{P_2} \to \overline{P_1}$. So, given a word $w \in F(X_2)$, evaluate the word $\varphi'(w)$, and then use the solution to the word problem in P_1 to compute if $\overline{\varphi'(w)} = e$ in $\overline{P_1}$. □

Corollary 2.55.
Suppose P_1 and P_2 are recursive presentations defining the same group. Suppose, moreover, that there is an algorithm to solve the word problem in P_1. Then there is an algorithm to solve the word problem in P_2.

So having solvable word problem is independent of the recursive presentation we are considering. Note that this is *not* the case for infinitely generated groups: the presentation $\langle x_i \ \forall i \in \mathbb{N} \mid - \rangle$ for F_∞ has solvable word problem, but the presentation $\langle x_i \ \forall i \in \mathbb{N} \mid x_i = e \ \forall i \in \mathbb{K} \rangle$ for F_∞ does not.

We now give some examples of groups with solvable word problem.

Example 2.56. *Each of the following classes of groups has (uniformly) solvable word problem amongst their finite presentations: Finitely generated free groups, finitely generated abelian groups, finite groups.*
That is, for each class named, there is an algorithm that on input of a finite presentation P in that class and a word w in P, decides if $\overline{w} = e$ in $\overline{P}$ or not.

We include the following result as we will have some use for it later on.

Proposition 2.57.
The word problem is uniformly solvable on all recursive presentations of non-trivial simple groups.

We prove this by first showing there exists a total algorithm for each recursive presentation of a simple group. We then show that this algorithm can be made uniform over all recursive presentations of *non-trivial* simple groups. Note that our proof does *not* extend to be uniform over all recursive presentations of simple groups if we include the trivial group in that class. Even if we restrict ourselves to the class of *finite* presentations of simple groups including the trivial group, the proof does not extend.

Proof. Take a recursive presentation $\langle X|R\rangle = \langle x_1, \dots x_n|R\rangle$ of a non-trivial simple group and fix a word $s \in F(X)$ representing a non-trivial element. Given an arbitrary word $w \in F(X)$, use (2.51) to begin an enumeration $w_1, w_2, \dots$ of all trivial words in $\langle X|R\rangle$, and at the same time an enumeration $y_1, y_2, \dots$ of all trivial words in $\langle X|R, w\rangle$. Now look for w in the first list, and s in the second list. If w is trivial then it will appear in the first list, if w is non-trivial then s will appear in the second list; it is clear that precisely one of these will occur as $\overline{\langle X|R\rangle}$ is non-trivial and simple. This algorithm can be made uniform over all finite presentations of non-trivial simple groups as follows: search for w in the first list, and search for all $x_i \in X$ in the second list. If the first search halts then w is trivial, if the second halts then w is non-trivial; again, precisely one will halt. □

Notice that our first algorithm works for each individual simple group because in any non-trivial group there exists a non-trivial element. But there is no reason to assume that the process of selecting a non-trivial element can be made uniform; see [9] for a discussion on this. The word problem mentions nothing about being able to recursively construct such an algorithm for the given presentation, only that one must exist. In fact there is no universal algorithm to solve the word problem on all groups with soluble word problem, nor is the class of finitely presented groups with soluble word problem recursively enumerable; see [11].

Note. It is an open problem as to whether (2.57) extends to all finite presentations of simple groups, where this class includes the trivial group. This can immediately be seen to be equivalent to asking if there is an algorithm for deciding triviality on finite presentations of simple groups, which is an open problem.

Definition 2.58.
We define the *subgroup membership problem* (abbreviated to MP) for a recursive presentation $P = \langle X|R\rangle$ of a group and a finite set of words $\{w_1, \ldots, w_m\} \subseteq F(X)$ as follows:

$$\text{Given a word } v \in F(X), \text{ is } \overline{v} \in \langle \overline{w}_1, \ldots, \overline{w}_m \rangle \text{ in } \overline{P}?$$

If there exists an algorithm that, on input of a word $v \in F(X)$, decides if $\overline{v} \in \langle \overline{w}_1, \ldots, \overline{w}_m \rangle$ in $\overline{P}$, then we say that $\langle \overline{w}_1, \ldots, \overline{w}_m \rangle$ has *solvable subgroup membership problem* in $\overline{P}$.

Not only can we enumerate all words equivalent to a given word, we can also enumerate all words equivalent to at least one word in an r.e. set. This eventually allows us to enumerate all words in a finitely generated subgroup.

Lemma 2.59.
Let $P = \langle X|R\rangle$ be an infinite recursive presentation, and $S := \{w_1, w_2, \ldots\}$ a recursive enumeration of words in $F(X)$. Then the words in $F(X)$ representing elements equal (in $\overline{P}$) to at least one such $w_i \in S$ are recursively enumerable. Moreover, this is uniform over all infinite recursive presentations of groups.

Proof. For each $w_i \in S$ (written as a word in $F(X)$), use (2.53) to enumerate all words in $F(X)$ equivalent to it (proceeding over finite diagonals for each i). □

Corollary 2.60.
Let $P = \langle X|R\rangle$ be an infinite recursive presentation, and $S := \{w_1, w_2, \ldots\}$ a recursive enumeration of words in $F(X)$. Then the words in $F(X)$ representing elements in $\langle S\rangle^{\overline{P}}$ are recursively enumerable, as are the words representing elements in $\langle\langle S\rangle\rangle^{\overline{P}}$. Moreover, each of these enumerations is uniform over all recursive presentations of groups.

Proof. In the case of $\langle S\rangle^{\overline{P}}$, we can enumerate $F(S)$ (all reduced words on $S \cup S^{-1}$). Moreover, in this enumeration, we may uniformly re-write each $s_i \in F(S)$ as a word in $F(X)$ (call it s_i'). So we now have a recursive enumeration of words in $F(X)$, such that each element in $\langle S\rangle^{\overline{P}}$ appears at least once as a word in this list. Now use (2.59) to enumerate all words equivalent to something in this r.e. set.
In the case of $\langle\langle S\rangle\rangle^{\overline{P}}$, we need only modify the above proof slightly. Observe that each word in $F(X)$ representing an element in $\langle\langle S\rangle\rangle^{\overline{P}}$ takes the form $\Pi_{i=1}^{n} w_i s'_{j_i} w_i^{-1}$ for some $n \in \mathbb{N}$, some $w_i \in F(X)$, and some $s'_{j_i} \in (F(S))'$. Enumerate all such words for $n = 1, n = 2, \ldots$ by proceeding along finite diagonals, each over all $w_i \in F(X)$. So we now have a recursive enumeration of words in $F(X)$, such that each element in $\langle\langle S\rangle\rangle^{\overline{P}}$ appears at least once as a word in $F(X)$. Now proceed as per the final part of the $\langle S\rangle^{\overline{P}}$ case above. □

We now see that the subgroup membership problem in an amalgamated product or HNN extension influences the word problem there.

Lemma 2.61.
Let G, H have SWP. Let $A \leqslant G$, $B \leqslant H$ be isomorphic finitely generated subgroups of G, H respectively, each having solvable subgroup membership problem (A in G, and B in H). Let $\varphi : A \to B$ be an isomorphism. Then the amalgamated product $G *_\varphi H$ has solvable word problem.

Proof. This follows from the reduced form theorem for amalgamated products (2.37). □

Lemma 2.62.
Let G have SWP. Let $A, B \leqslant G$ be isomorphic finitely generated subgroups of G, each having solvable subgroup membership problem in G. Let $\varphi : A \to B$ be an isomorphism. Then the HNN extension $G*_\varphi$ has solvable word problem.

Proof. This follows from the normal form theorem for HNN extensions (2.33). □

2.7. **Maps and homomorphisms.**

We begin with the most important observation of this section: homomorphisms between finitely presented groups are r.e. Recall our convention that all countably generated group presentations draw their generators from the set $\mathcal{A} = \{x_1, x_2, \ldots\}$.

Lemma 2.63.
There is a partial algorithm on all pairs of finite presentations of groups and set maps between them that, on input of two finite presentations $P = \langle X|R\rangle$ and $Q = \langle Y|S\rangle$ and a set map $\phi : X \to Y^*$, halts if and only if ϕ extends to a homomorphism $\overline{\phi} : \overline{P} \to \overline{Q}$.

Proof. For each $r \in R$, use (2.51) to check if $\overline{\phi}(r) = e$ in $\overline{Q}$. This process will halt if and only if $\overline{\phi}(r) = e$ in $\overline{Q}$ for all $r \in R$, thus if and only if $\overline{\phi}$ is a homomorphism. □

Using this, we can check for surjections between finitely presented groups.

Lemma 2.64.
There is a partial algorithm on all pairs of finite presentations of groups that, on input of two finite presentations $P = \langle X|R\rangle$, $Q = \langle Y|S\rangle$, terminates if and only if there exists a surjective homomorphism $f : \overline{P} \twoheadrightarrow \overline{Q}$. Moreover, whenever this algorithm terminates, it outputs a map $\phi : X \to Y^*$ which extends to a surjective homomorphism $\overline{\phi} : \overline{P} \twoheadrightarrow \overline{Q}$.

Proof. Begin an enumeration $\phi_1, \phi_2, \ldots$ of all set maps $\phi_i : X \to Y^*$. Now begin checking each ϕ_i to see if it extends to a homomorphism by (2.63). For each such ϕ_i identified in this way, use (2.60) to test if $Y \subseteq \langle \overline{\phi}_i(X)\rangle^{\overline{Q}} =: \mathrm{Im}(\overline{\phi}_i)$ (i.e., if $\overline{\phi}_i$ is a surjection). This will halt if and only if there exists a surjection from $\overline{P}$ onto $\overline{Q}$; all ϕ_i on which it halts extend to such a surjection. □

What is most useful for our later work however is the following fact that isomorphisms between finitely presented groups are r.e. In addition, when there is such an isomorphism, we can construct it.

Proposition 2.65.
There is a partial algorithm on all pairs of finite presentations of groups that, on input of two finite presentations $P = \langle X|R\rangle$ and $Q = \langle Y|S\rangle$, halts if and only if $\overline{P} \cong \overline{Q}$, and outputs isomorphisms between them if they are.

Proof. Begin an enumeration of all set maps $\phi_i : X \to Y^*$, and similarly an enumeration of all set maps $\psi_j : Y \to F(X)$. Now, proceeding along finite diagonals, begin checking which of the ϕ_i's and ψ_j's extend to surjective homomorphisms, using (2.64). For each such pair, begin checking if $\overline{\psi_j} \circ \overline{\phi_i}(x) = x$

in $\overline{P}$ for all $x \in X$, and if $\overline{\phi_i} \circ \overline{\psi_j}(y) = y$ in $\overline{Q}$ for all $y \in Y$, using (2.53). Such an overall process will eventually halt if and only if $\overline{P} \cong \overline{Q}$; if it does halt then the maps ϕ_i and ψ_j on which it halts extend to isomorphisms. □

It is important to note that the above proof does not hold if we instead consider recursive presentations. In fact, even if we start with one recursive presentation, and one finite presentation, the proof does not hold. In this case, the proof fails because we cannot algorithmically verify if a map extends to a homomorphism, as we cannot check the (possibly infinite) list of relators to see if they are all trivial. For a discussion of this see [10].

Corollary 2.66.
Let $P = \langle X|R\rangle$ be a finite presentation. Then the set of finite presentations defining groups isomorphic to $\overline{P}$ is recursively enumerable. Moreover, this algorithm is uniform over all finite presentations.

Proof. Begin an enumeration of all finite presentations $P_1, P_2, \ldots$. For each such presentation P_i, use (2.65) to test if $\overline{P_i} \cong \overline{P}$; for all such presentations satisfying this condition, the process will eventually halt. □

Corollary 2.67.
Let $P_1, P_2, \ldots$ be a recursive enumeration of finite presentations. Then the set of all finite presentations isomorphic to at least one P_i is recursively enumerable.

Proof. For each P_i, use (2.66) to form an enumeration of all finite presentations isomorphic to P_i (by proceeding along finite diagonals over each i). □

As is to be expected, the set of finite presentations of finite groups is quite 'well-behaved'. We mention one property in particular because it will be of use to us later on.

Lemma 2.68.
The set of finite presentations of finite groups is r.e.

Proof. For each $n \in \mathbb{N}$, we can list all $n \times n$ multiplication tables on symbols $\{g_1, \ldots, g_n\}$. Since each table is finite, we can check the group axioms for each table, and verify if it is a group or not. We can then generate a finite presentation from each accepted table, with generating set $\{g_1, \ldots, g_n\}$ and relating set given by all multiplications in the table. Since there are only finitely many tables for each n, we can thus enumerate all the presentations we get in this way (for $n = 1$, then $n = 2$, and so on), and every finite group will appear at least once in this list of presentations. (If we really want, we can use (2.66) to begin an enumeration of all finite presentations isomorphic to each such presentation, by proceeding along finite diagonals. This will enumerate all finite presentations of finite groups). □

Not only can we list all finite presentations of finite groups, we can extend the above proof to show that we can determine the size of any given finite group.

Corollary 2.69.
There is a partial algorithm on all finite presentations of groups that, on input of a finite presentation P of a finite group, outputs $|\overline{P}|$.

Proof. Using the above proof, we need only trace back and check to see which finite multiplication table corresponds to P, this process will eventually terminate as $\overline{P}$ is finite, and the (edge) size of the table gives the size of $|\overline{P}|$. □

3. SIMULATING MACHINES DIRECTLY IN GROUPS

3.1. **A finitely presented semigroup with unsolvable word problem.**

Most of what we do in this section is taken from chapter 12 in the book [12] by Rotman.

Definition 3.1.
We define the *word problem* (WP) for a recursive presentation $P = \langle X|R\rangle$ of a semigroup as follows:

$$\text{Given two words } u, v \in X^\circ, \text{ is } \overline{u} = \overline{v} \text{ in } \overline{P}?$$

Semigroups which have a presentation for which there exists an algorithm that, on input of a pair of words $u, v \in X^\circ$, decides if $\overline{u} = \overline{v}$ in $\overline{P}$, are said to have *solvable word problem* (or *soluble word problem*, or *decidable word problem*), abbreviated to SWP. If there is no such presentation for which such an algorithm exists, then we say the semigroup has *unsolvable word problem* (or *insoluble word problem*, *undecidable word problem*), abbreviated to IWP.

Turing machines are not just confined to mathematical logic. We can fully 'realise' the action of a Turing machine in algebraic structures. The following construction is originally due to Markov and Post (independently), and was the first example of a finitely presented semigroup with unsolvable word problem. In essence, they took the construction of a Turing machine and encoded it into a finite presentation of a semigroup in quite a natural way. This construction mimics the inner workings of the Turing machine in the semigroup, without the need for much extra peripheral structure. It is interesting to compare this with the construction we will see later of a finite presentation of a group with unsolvable word problem, which takes a different machine encoding approach (modular machines, which we introduced earlier), but for which the action of the machine is buried much more deeply within the algebraic structure.

As a notational convention, given a semigroup presentation $P = \langle X|R\rangle$, we call an element $(x, y) \in R$ a *semigroup relator* (or just *relator*), and often write this as $x = y$.

Definition 3.2.
Let T be a Turing machine with alphabet $S = \{s_0, \ldots, s_m\}$, states $Q = \{q_0, \ldots, q_n\}$, and halting state q_0. We define the *associated semigroup* $\overline{\Gamma(T)}$ to be the semigroup presented by

$$\Gamma(T) := \langle q, h, s_0, \ldots . s_m, q_0, \ldots, q_n \mid R(T) \rangle$$

where the relators in $R(T)$ are, for all $i, l \in \{1, \ldots, n\}$, all $j, k \in \{0, \ldots, n\}$, and all $\beta \in \{0, \ldots, m\}$:

$$\begin{aligned}
q_i s_j = q_l s_k \quad &\text{if} \quad q_i s_j s_k q_l \in T \\
q_i s_j s_\beta = s_j q_l s_\beta \quad &\text{if} \quad q_i s_j R q_l \in T \\
q_i s_j h = s_j q_l s_0 h \quad &\text{if} \quad q_i s_j R q_l \in T \\
s_\beta q_i s_j = q_l s_\beta s_j \quad &\text{if} \quad q_i s_j L q_l \in T \\
h q_i s_j = h q_l s_0 s_j \quad &\text{if} \quad q_i s_j L q_l \in T \\
q_0 s_\beta = q_0 & \\
s_\beta q_0 h = q_0 h & \\
h q_0 h = q &
\end{aligned}$$

To give some sort of explanation: the first 5 rows of relators in the definition above precisely mimic the action of the Turing machine (1.3) as it computes a given input, by virtue of the fact that their conditions are all quadruples from the Turing machine. The final three rows of relators ensure that, when we reach the halting state q_0, everything collapses down to q. The symbol h can be viewed as an 'end-marker'.

Theorem 3.3 (Markov-Post, 1947)**.**
Let T be a Turing machine, and let $\Gamma(T)$ be the associated semigroup presentation, as in (3.2)*. If $w \in S^+$, then*

$$w \in \Omega(T) \text{ if and only if } \overline{hq_1wh} = \overline{q} \text{ in } \overline{\Gamma(T)}$$

Proof. First, if $w \in \Omega(T)$ then there is a sequence of instantaneous descriptions $q_1w = w_0, w_1, \ldots, w_n = \alpha q_0 \beta$, such that $\alpha, \beta \in S^+$, and each successive pair w_i, w_{i+1} differ by the action of one quadruple from T (so we mirror the action of T on w by the instantaneous descriptions). But by the first 5 relator types of $\Gamma(T)$, if w_i, w_{i+1} differ by the action of one quadruple from T, then $\overline{hw_ih} = \overline{hw_{i+1}h}$ by (2.28). So we get $\overline{hq_1wh} = \overline{h\alpha q_0\beta h}$, and then $\overline{h\alpha q_0\beta h} = \overline{q}$ by repeated application of the last 3 relator types of $\Gamma(T)$, again using (2.28).

For the other direction, notice that by (2.28) if $\overline{hq_1wh} = \overline{q}$ in $\overline{\Gamma(T)}$ then we have a sequence of words $hq_1wh = w_0, w_1, \ldots, w_n = q$, all defining the same element in $\overline{\Gamma(T)}$, each differing by application of one relator. Take a sequence of shortest possible length N. Since the only way to reach q is via the relator $hq_0h = q$, then we must have $w_{N-1} = hq_0h$ (by minimality of N). The only way to reach a word of the form hq_0h (not from q) is to have words of the form $h\alpha q_0\beta h$ in the sequence, where $\alpha, \beta \in S^+$. Let w_M be the first word with q_0 appearing. Again, by the minimality of N, we have that q_0 appears in all words $w_M, w_{M+1}, \ldots, w_{N-1}$. Thus the sequence $w_0, \ldots, w_M$ is obtained by repeated application of relators of the first 5 types in $\Gamma(T)$. Seeing as the first word in the sequence (w_0) is an instantaneous description (buffered either side by h), and all the relators of the first 5 types preserve instantaneous descriptions, then *all* the w_i's are instantaneous descriptions (buffered either side by h), for $i \leq M$.

Note that each relator has corresponding 'time' direction in the Turing machine T. For example, the relator $q_is_j = q_ls_k$ (if $q_is_js_kq_l \in T$) is 'forwards in time' if we replace $q_is_j \to q_ls_k$, and 'backwards in time' if we replace $q_ls_k \to q_is_j$. So each adjacent pair w_i, w_{i+1} corresponds to either a 'forwards in time' or 'backwards in time' application of a relator. Obviously, the pair w_{M-1}, w_M corresponds to a 'forwards in time' relator, as this is the only way to introduce the symbol q_0. Assume that not all pairs are 'forwards in time'. As the sequence ends with a 'forwards in time' pair w_{M-1}, w_M, there must be some $1 \leq j \leq M-1$ such that w_{j-1}, w_j is backwards in time, but w_j, w_{j+1} is forwards in time. But then, by the deterministic nature of a Turing machine, we must have that $w_{j-1} = w_{j+1}$ as words, and so our sequence was not of minimal length. This is a contradiction, so all pairs w_i, w_{i+1} corresponds to a 'forwards in time' application of a relator. This implies that, on input of the word w, the Turing machine eventually reaches the internal state q_0, following these 'forwards in time' pairs w_i by application of quadruples from T. So $T(w)$ halts. □

So we have proved that the action of T is *completely* simulated within the finitely presented semigroup $\overline{\Gamma(T)}$, and moreover we can algorithmically construct $\Gamma(T)$ from T as given in (3.3). Later, we will show that there is an analogous construction which simulates the action of a Turing machine within a finitely presented *group*, and then use it to prove the Higman embedding theorem.

Theorem 3.4.
There is a finitely presented semigroup with unsolvable word problem.

Proof. Let T_n be the Turing machine from (1.22) with halting set $\Omega(T_n) = \mathbb{K}$, and form $\Gamma(T_n)$ as in (3.2). Assume that $\overline{\Gamma(T)}$ has SWP; then there is an algorithm which takes any pair of words in the generators of $\Gamma(T)$ and decides if they give the same element in $\overline{\Gamma(T)}$. But now, given any $w \in S^+$ with S being the alphabet of T, we can use the algorithm for the word problem in $\overline{\Gamma(T)}$ to compute whether $\overline{hq_1wh} = \overline{q}$, and thus whether $w \in \Omega(T_n)$ by (3.3). But $\Omega(T_n) = \mathbb{K}$, which by (1.23) is not recursive. □

3.2. **A finitely presented group with unsolvable word problem.**

There is an excellent survey [11] by Charles F. Miller III on the word problem in groups, and indeed many other group-theoretic decision problems. This resource is freely available.

To begin, we illustrate how easy it is to construct a *recursively* presented group with IWP.

Theorem 3.5.
There exists a recursively presented group with IWP.

Proof. Form the recursive presentation

$$Q = \langle a, b, c, d \mid b^n a b^{-n} = d^n c d^{-n} \ \forall\ n \in \mathbb{K} \rangle$$

This is an amalgamated product $\overline{\langle a, b|-\rangle} *_\varphi \overline{\langle c, d|-\rangle}$ over subgroups $A := \langle \overline{b^n a b^{-n}} \ \forall\ n \in \mathbb{K} \rangle$ and $B := \langle \overline{d^n c d^{-n}} \ \forall\ n \in \mathbb{K} \rangle$, both isomorphic to F_∞ by (2.13) as $\mathbb{K}$ is infinite. The isomorphism $\varphi : A \to B$ is then given by extending the map $\overline{b^n a b^{-n}} \mapsto \overline{d^n c d^{-n}}\ \forall\ n \in \mathbb{K}$. Then, by the reduced form theorem for amalgamated products (2.37), we have that:

$$\overline{b^n a b^{-n} (d^n c d^{-n})^{-1}} = e \text{ in } \overline{Q} \ \Leftrightarrow\ n \in \mathbb{K}$$

So an algorithm solving the word problem for Q will give an algorithm which decides membership in $\mathbb{K}$, which is impossible by (1.23). □

We now show that there is a finitely presented group with unsolvable word problem. We will actually show a much stronger result: that we can take any modular machine, and from it define a finitely presented group which simulates the action of this machine. A quick application of the halting set $\mathbb{K}$ then gives us a finitely presented group with unsolvable word problem. In the next section we will use this finitely presented group that simulates a modular machine to show the Higman embedding theorem: *every recursively presented group embeds in some finitely presented group.*

Our approach will be to outline the entire construction first, and state (without proof) each of the intermediate results that we need. After this initial construction, we prove all the steps that are not obvious (these are denoted with a * in the construction below). This is taken from Cohen's book [13].

Construction 3.6.
The following is a construction for simulating a modular machine within a finitely presented group:

(1) Define the group $K := \mathbb{Z} * (\mathbb{Z} \times \mathbb{Z})$ with presentation $\langle x, y, t \mid [x, y] = e \rangle$.
(2) Define, for all $(r, s) \in \mathbb{Z}^2$, the word $t(r, s) := y^s x^r t x^{-r} y^{-s}$.
(3) Define the subgroup $T := \langle \{\overline{t(r,s)}\}_{(r,s)\in\mathbb{Z}^2} \rangle \leq K$.
(4) *Observe that T is free with basis $\{\overline{t(r,s)}\}_{(r,s)\in\mathbb{Z}^2}$.
(5) For $M > a \geq 0$, $N > b \geq 0$, define

$$T_{a,b}^{M,N} := \langle \{\overline{t(\alpha,\beta)} \mid \alpha \equiv a \mod M,\ \beta \equiv b \mod N\} \rangle \leq T \leq K$$

$$K_{a,b}^{M,N} := \langle \overline{t(a,b)}, \overline{x}^M, \overline{y}^N \rangle \leq K$$

(6) *Observe that $T_{a,b}^{M,N} \cong T$ via extension of the map $\overline{t(uM+a, vN+b)} \mapsto \overline{t(u,v)}$ $\forall u, v \in \mathbb{Z}$.
(7) *Observe that $K_{a,b}^{M,N} \cong K$ via extension of $\overline{t(a,b)} \mapsto \overline{t}$, $\overline{x}^M \mapsto \overline{x}$, $\overline{y}^N \mapsto \overline{y}$.
(8) *Observe that $T \cap K_{a,b}^{M,N} = T_{a,b}^{M,N}$ in K.
(9) Let $\mathcal{M} = \{(a_i, b_i, c_i, R) \mid i \in I\} \cup \{(a_j, b_j, c_j, L) \mid j \in J\}$ be a modular machine with modulus m, in which $(0, 0)$ is terminal.
(10) Define, for each $i \in I$, the map $\phi_i : K_{a_i,b_i}^{m,m} \to K_{c_i,0}^{m^2,1}$ via extension of the map $\overline{t(a_i,b_i)} \mapsto \overline{t(c_i,0)}$, $\overline{x}^m \mapsto \overline{x}^{m^2}$, $\overline{y}^m \mapsto y$.
(11) Define, for each $j \in I$, the map $\varphi_j : K_{a_j,b_j}^{m,m} \to K_{0,c_j}^{1,m^2}$ via extension of the map $\overline{t(a_j,b_j)} \mapsto \overline{t(0,c_j)}$, $\overline{x}^m \mapsto \overline{x}$, $\overline{y}^m \mapsto \overline{y}^{m^2}$.
(12) *Observe that ϕ_i and φ_j are isomorphisms for every $i \in I$, $j \in J$.
(13) Define the following HNN extension with stable letters $\{r_i\}_{i\in I}$, $\{l_j\}_{j\in J}$.

$$K_{\mathcal{M}} := K *_{\{\phi_i\}_{i\in I}, \{\varphi_j\}_{j\in J}}$$

(14) *Observe that $K_{\mathcal{M}}$ is finitely presented.
(15) Define the subgroup $T' := \langle T,\ \overline{r_i}\ \forall i \in I,\ \overline{l_j}\ \forall j \in J \rangle \leq K_{\mathcal{M}}$.
(16) *Observe that T is a good subgroup of K with respect to the HNN extension $K_{\mathcal{M}}$, and thus $T = T' \cap K$.
(17) Define $T_{\mathcal{M}} := \langle \{\overline{t(\alpha,\beta)} \mid (\alpha, \beta) \in H_0(\mathcal{M})\} \rangle \leq T \leq K$.
(18) Define $T'_{\mathcal{M}} := \langle T_{\mathcal{M}},\ \overline{r_i}\ \forall i \in I,\ \overline{l_j}\ \forall j \in J \rangle \leq K_{\mathcal{M}}$.
(19) Define $\langle \overline{t} \rangle' := \langle \overline{t},\ \overline{r_i}\ \forall i \in I,\ \overline{l_j}\ \forall j \in J \rangle \leq K_{\mathcal{M}}$.
(20) *Observe that $T_{\mathcal{M}}$ is a good subgroup of K with respect to the HNN extension $K_{\mathcal{M}}$, and thus $T_{\mathcal{M}} = T'_{\mathcal{M}} \cap K$.
(21) *Observe that $T'_{\mathcal{M}} = \langle \overline{t} \rangle'$ in $K_{\mathcal{M}}$.
(22) *Observe that $T_{\mathcal{M}} = \langle \overline{t} \rangle' \cap K$.
(23) *Observe that $\overline{t(\alpha,\beta)} \in \langle \overline{t} \rangle'$ iff $(\alpha, \beta) \in H_0(\mathcal{M})$.
(24) Define the HNN extension $G_{\mathcal{M}}$ with presentation

$$\langle K_{\mathcal{M}}; k \mid khk^{-1} = h\ \forall h \in \langle t \rangle' \rangle$$

(25) *Observe that $G_{\mathcal{M}}$ is finitely presented.
(26) *Observe that $\overline{kt(\alpha,\beta)k^{-1}} = \overline{t(\alpha,\beta)}$ in $G_{\mathcal{M}}$ iff $(\alpha, \beta) \in H_0(\mathcal{M})$.

We now prove all the nontrivial steps in (3.6), to show that the construction is valid. We will suppress the use of over lines $\overline{w}$ in the proofs, and just use the underlying words as group elements. This is to save on heavy notation; the meaning should be clear in all cases.

Step 4: T is free with basis $\{\overline{t(r,s)}\}_{(r,s)\in\mathbb{Z}^2}$.
By definition, $\{\overline{t(r,s)}\}_{(r,s)\in\mathbb{Z}^2}$ generates T. By applying the normal form theorem for free products (2.12), in the same way that we did in (2.13), we have that these elements *freely* generate T, and thus T is free.

Step 6: $T_{a,b}^{M,N} \cong T$ via extension of the map sending $\overline{t(uM+a, vN+b)} \mapsto \overline{t(u,v)}$ $\forall u, v \in \mathbb{Z}^2$.
By step 4, we have that $\{t(r,s)\}_{(r,s)\in\mathbb{Z}^2}$ is a free basis for T. Thus any subset $S \subseteq \{t(r,s)\}_{(r,s)\in\mathbb{Z}^2}$ is a free basis for the subgroup $\langle S\rangle$ it generates (2.16). In particular, as $\{t(\alpha,\beta) \mid \alpha \equiv a \mod M, \beta \equiv b \mod N\} \subseteq \{t(r,s)\}_{(r,s)\in\mathbb{Z}^2}$, and both sets have the same cardinality (via the map sending $t(uM+a, vN+b) \mapsto t(u,v)$), then they freely generate isomorphic groups with isomorphism given via the map extending $t(uM+a, vN+b) \mapsto t(u,v)$ $\forall u, v \in \mathbb{Z}$; see (2.7) and (2.16).

Step 7: $K_{a,b}^{M,N} \cong K$ via extension of the map $\overline{t(a,b)} \mapsto \bar{t}$, $\overline{x}^M \mapsto \overline{x}$, $\overline{y}^N \mapsto \overline{y}$.
First, observe that $K_{a,b}^{M,N}$ is conjugate to $K_{0,0}^{M,N} = \langle t, x^M, y^N\rangle$, where $t(0,0) = t$ (conjugate by $x^{-a}y^{-b}$; this commutes with both x, y, thus $x^{-a}y^{-b}t(a,b)y^bx^b = t$). Thus the two are isomorphic; see [8, 1.28]. But $\langle t, x^M, y^N\rangle$ is generated by $\langle t\rangle$ and $\langle x^M, y^N\rangle$ which lie in different free factors of K. So $\langle t, x^M, y^N\rangle \cong \langle t\rangle * \langle x^M, y^N\rangle \cong \mathbb{Z} * (\mathbb{Z}\times\mathbb{Z})$.

Step 8: $T \cap K_{a,b}^{M,N} = T_{a,b}^{M,N}$ in K.
For $\supseteq$, note that $t(uM+a, vN+b) = y^{vN}x^{uM}t(a,b)x^{-uM}y^{-vN} \in K_{a,b}^{M,N}$.
For $\subseteq$, note that $x^Mt(\alpha,\beta) = t(\alpha+M,\beta)x^M$ and $y^Nt(\alpha,\beta) = t(\alpha,\beta+N)y^N$, hence any element of $K_{a,b}^{M,N}$ is of the form $gx^{uM}y^{vN}$ where $g \in T_{a,b}^{M,N}$ and $u, v \in \mathbb{Z}$. If this element is in T, then $u = v = 0$ (hence it is in $T_{a,b}^{M,N}$), by the following argument: If $gx^{uM}y^{vN} \in T$ then $x^{uM}y^{vN} \in T$ as $g \in T_{a,b}^{M,N} \leq T$. But as $x^{uM}y^{vN} \in T$ which is free with basis $\{t(r,s)\}_{(r,s)\in\mathbb{Z}^2}$ (by step 4), we have that $x^{uM}y^{vN} = t(c_1,d_1)^{\epsilon_1}\dots t(c_l,d_l)^{\epsilon_l}$ for $\epsilon_i \in \{\pm 1\}$, and thus $e = y^{-vN}x^{-uM}t(c_1,d_1)^{\epsilon_1}\dots t(c_l,d_l)^{\epsilon_l}$. But by the normal form theorem for free products (2.12), as K is a free product, then the word $y^{-vN}x^{-uM}t(c_1,d_1)^{\epsilon_1}\dots t(c_l,d_l)^{\epsilon_l}$ can only be trivial if $t(c_1,d_1)^{\epsilon_1}\dots t(c_l,d_l)^{\epsilon_l}$ is trivial (or there would be a non-cancelling occurrence of t). Thus $e = y^{-vN}x^{-uM}$, and so $v = u = 0$ as x, y generate the $\mathbb{Z}\times\mathbb{Z}$ free factor of K.

Step 12: ϕ_i and φ_j are isomorphisms for every $i \in I$, $j \in J$.
This follows immediately from step 7.

Step 14: $K_{\mathcal{M}}$ is finitely presented.
Note that K is finitely presented, and $K_{\mathcal{M}}$ is a finite tower of HNN extensions of K, each over finitely generated subgroups. It then follows from (2.31) that $K_{\mathcal{M}}$ is finitely presented.

Step 16: T is a good subgroup of K with respect to the HNN extension $K_{\mathcal{M}}$, and thus $T = T' \cap K$.

To verify T is good, we need to show $\phi_i(T\cap K^{m,m}_{a_i,b_i}) = T\cap K^{m^2,1}_{c_i,0}$ for all $i \in I$, and $\varphi_j(T\cap K^{m,m}_{a_j,b_j}) = T\cap K^{1,m^2}_{0,c_j}$ for all $j \in J$. From step 8, we have $T\cap K^{m,m}_{a_i,b_i} = T^{m,m}_{a_i,b_i}$ and $T \cap K^{m^2,1}_{c_i,0} = T^{m^2,1}_{c_i,0}$. But $\phi_i(t(um + a_i, vm + b_i)) = t(um^2 + c_i, v)$, so $\phi_i(T^{m,m}_{a_i,b_i}) = T^{m^2,1}_{c_i,0}$ as ϕ_i is an extension of a map sending a free basis of $T^{m,m}_{a_i,b_i}$ to a free basis of $T^{m^2,1}_{c_i,0}$. So we're done for ϕ_i, and an identical argument works for φ_j. So T is good, and thus by (2.45) $T = T' \cap K$.

Step 20: $T_{\mathcal{M}}$ is a good subgroup of K with respect to the HNN extension $K_{\mathcal{M}}$, and thus $T_{\mathcal{M}} = T'_{\mathcal{M}} \cap K$.
To verify $T_{\mathcal{M}}$ is good, we need to show $\phi_i(T_{\mathcal{M}} \cap K^{m,m}_{a_i,b_i}) = T_{\mathcal{M}} \cap K^{m^2,1}_{c_i,0}$ for all $i \in I$, and $\varphi_j(T_{\mathcal{M}} \cap K^{m,m}_{a_j,b_j}) = T_{\mathcal{M}} \cap K^{1,m^2}_{0,c_j}$ for all $j \in J$. Note that if $\phi_i(t(\alpha,\beta)) = t(\alpha_1,\beta_1)$ or $\varphi_j(t(\alpha,\beta)) = t(\alpha_1,\beta_1)$ then $(\alpha,\beta) \underset{\mathcal{M}}{\longrightarrow} (\alpha_1,\beta_1)$ by definition of ϕ_i, φ_j. Also, $t(\alpha,\beta) \in T_{\mathcal{M}}$ iff $(\alpha,\beta) \in H_0(\mathcal{M})$ by (2.16) since T is free on $\{t(r,s)\}_{(r,s)\in\mathbb{Z}^2}$ (step 4) and $T_{\mathcal{M}}$ is free on a subset of this set. Thus $t(\alpha,\beta) \in T_{\mathcal{M}} \Leftrightarrow (\alpha,\beta) \in H_0(\mathcal{M}) \Leftrightarrow (\alpha_1,\beta_1) \in H_0(\mathcal{M}) \Leftrightarrow t(\alpha_1,\beta_1) \in T_{\mathcal{M}}$. Now observe that $T_{\mathcal{M}} \cap K^{m,m}_{a_i,b_i} = T_{\mathcal{M}} \cap T \cap K^{m,m}_{a_i,b_i}$ as $T_{\mathcal{M}} \leqslant T$, and thus $T_{\mathcal{M}} \cap T \cap K^{m,m}_{a_i,b_i} = T_{\mathcal{M}} \cap T^{m,m}_{a_i,b_i}$ (step 8). By step 4 and (2.16), we see that $T_{\mathcal{M}} \cap T^{m,m}_{a_i,b_i} = \langle\{t(\alpha,\beta) \in T_{\mathcal{M}} \mid \alpha \equiv a_i \mod m,\ \beta \equiv b_i \mod m\}\rangle$ (free on the intersection of their free bases, both of which are subsets of a free basis for T). A similar argument gives that $T_{\mathcal{M}} \cap T^{m^2,1}_{c_i,0} = \langle\{t(\alpha,\beta) \in T_{\mathcal{M}} \mid \alpha \equiv c_i \mod m^2,\ \beta \equiv 0 \mod 1\}\rangle$. By the first part of the argument, if $\phi_i(t(\alpha,\beta)) = t(\alpha',\beta')$ then $t(\alpha,\beta) \in T_{\mathcal{M}} \Leftrightarrow t(\alpha',\beta') \in T_{\mathcal{M}}$. From this it follows that $\phi_i(T_{\mathcal{M}} \cap T^{m,m}_{a_i,b_i}) = T_{\mathcal{M}} \cap T^{m^2,1}_{c_i,0}$ (If $t(\alpha,\beta) \in T_{\mathcal{M}} \cap T^{m,m}_{a_i,b_i}$ then $\phi_i(t(\alpha,\beta)) \in T_{\mathcal{M}}$ as $t(\alpha,\beta) \in T_{\mathcal{M}}$, and $\phi_i(t(\alpha,\beta)) \in T^{m^2,1}_{c_i,0}$ as $t(\alpha,\beta) \in T^{m,m}_{a_i,b_i}$. Thus $\phi_i(t(\alpha,\beta)) \in T_{\mathcal{M}} \cap T^{m^2,1}_{c_i,0}$, and so $\phi_i(T_{\mathcal{M}} \cap T^{m,m}_{a_i,b_i}) \subseteq T_{\mathcal{M}} \cap T^{m^2,1}_{c_i,0}$. Conversely, if $t(\alpha,\beta) \in T_{\mathcal{M}} \cap T^{m^2,1}_{c_i,0}$ then $t(\alpha,\beta) \in T^{m^2,1}_{c_i,0}$, so by step 16 there exists $t(\alpha',\beta') \in T^{m,m}_{a_i,b_i}$ such that $\phi_i(t(\alpha',\beta')) = t(\alpha,\beta)$. But then $t(\alpha',\beta') \in T_{\mathcal{M}}$ as $\phi_i(t(\alpha',\beta')) \in T_{\mathcal{M}}$, so $t(\alpha',\beta') \in T_{\mathcal{M}} \cap T^{m,m}_{a_i,b_i}$, and thus $\phi_i(T_{\mathcal{M}} \cap T^{m,m}_{a_i,b_i}) \supseteq T_{\mathcal{M}} \cap T^{m^2,1}_{c_i,0}$). So we're done for ϕ_i, and an identical argument works for φ_j. Thus $T_{\mathcal{M}}$ is good, and so by (2.45), $T_{\mathcal{M}} = T'_{\mathcal{M}} \cap K$.

Step 21: $T'_{\mathcal{M}} = \langle\bar{t}\rangle'$ in $K_{\mathcal{M}}$.
Clearly $T'_{\mathcal{M}} \supseteq \langle t\rangle'$ in $K_{\mathcal{M}}$, as $t = t(0,0) \in T_{\mathcal{M}} \leq T'_{\mathcal{M}}$ and all r_i, l_j lie in $T'_{\mathcal{M}}$.
To prove the $\subseteq$ direction, it suffices to show that $t(\alpha,\beta) \in \langle t\rangle'$ for all $(\alpha,\beta) \in H_0(\mathcal{M})$. We do this by induction on the length of the computation which takes (α,β) to $(0,0)$. Clearly, for $(\alpha,\beta) = (0,0)$ (computation of length 0) we have $t(0,0) = t \in \langle t\rangle'$.
Now assume that $(\alpha,\beta) \underset{\mathcal{M}}{\longrightarrow} (\alpha_1,\beta_1)$ via the modular machine computation (a_i, b_i, c_i, R), and so $\alpha = um + a_i$, $\beta = vm + b_i$, and $\alpha_1 = um^2 + c_i$, $\beta_1 = v$.

Then we have

$$\begin{aligned} t(\alpha,\beta) &= y^{\beta}x^{\alpha}tx^{-\alpha}y^{-\beta} \\ &= y^{vm+b_i}x^{um+a_i}tx^{-um-a_i}y^{-vm-b_i} \\ &= y^{vm}x^{um}y^{b_i}x^{a_i}tx^{-a_i}y^{-b_i}x^{-um}y^{-vm} \\ &= y^{vm}x^{um}t(a_i,b_i)x^{-um}y^{-vm} \end{aligned}$$

So we thus have

$$\begin{aligned} r_i t(\alpha,\beta)r_i^{-1} &= \phi_i(t(\alpha,\beta)) \\ &= \phi_i(y^{vm}x^{um}t(a_i,b_i)x^{-um}y^{-vm}) \\ &= y^{v}x^{um^2}t(c_i,0)x^{-um^2}y^{-v} \\ &= y^{v}x^{um^2+c_i}tx^{-um^2-c_i}y^{-v} \\ &= t(um^2+c_i,v) \\ &= t(\alpha_1,\beta_1) \end{aligned}$$

So if $(\alpha,\beta) \in H_0(\mathcal{M})$, then $(\alpha_1,\beta_1) \in H_0(\mathcal{M})$ by a shorter computation, and so by induction $t(\alpha_1,\beta_1) \in \langle t\rangle'$. Thus $t(\alpha,\beta) = r_i^{-1}t(\alpha_1,\beta_1)r_i \in \langle t\rangle'$ as both r_i and $t(\alpha_1,\beta_1)$ are in $\langle t\rangle'$.
If $(\alpha,\beta) \underset{\mathcal{M}}{\longrightarrow} (\alpha_1,\beta_1)$ via the modular machine computation (a_j,b_j,c_j,L), then the proof is practically identical. So we are done.

Step 22: $T_{\mathcal{M}} = \langle \bar{t}\rangle' \cap K$.
This is immediate from steps 20 and 21.

Step 23: $\overline{t(\alpha,\beta)} \in \langle \bar{t}\rangle'$ iff $(\alpha,\beta) \in H_0(\mathcal{M})$.
If $(\alpha,\beta) \in H_0(\mathcal{M})$ then $t(\alpha,\beta) \in T_{\mathcal{M}} = \langle t\rangle' \cap K$ by step 22. Conversely, it is clear that $t(\alpha,\beta) \in K$, and thus if $t(\alpha,\beta) \in \langle t\rangle'$ then $t(\alpha,\beta) \in \langle t\rangle' \cap K = T_{\mathcal{M}}$ (step 22) and thus $(\alpha,\beta) \in H_0(\mathcal{M})$ as $T_{\mathcal{M}}$ is free on $\{t(\alpha,\beta) \mid (\alpha,\beta) \in H_0(\mathcal{M})\}$ which is a subset of the free basis $\{t(r,s)\}_{(r,s)\in\mathbb{Z}^2}$ for T; see (2.16).

Step 25: $G_{\mathcal{M}}$ is finitely presented.
Note that $K_{\mathcal{M}}$ is finitely presented by step 14, and $G_{\mathcal{M}}$ is an HNN extension of $K_{\mathcal{M}}$ over a finitely generated subgroup $\langle t\rangle'$. It then follows from (2.31) that $G_{\mathcal{M}}$ is finitely presented.

Step 26: $\overline{k^{-1}t(\alpha,\beta)k} = \overline{t(\alpha,\beta)}$ in $G_{\mathcal{M}}$ iff $(\alpha,\beta) \in H_0(\mathcal{M})$.
By Britton's lemma (2.41), for $g \in K_{\mathcal{M}}$ we have $kgk^{-1} = g \Leftrightarrow g \in \langle t\rangle'$. In particular, $kt(\alpha,\beta)k^{-1} = t(\alpha,\beta) \Leftrightarrow t(\alpha,\beta) \in \langle t\rangle' \Leftrightarrow (\alpha,\beta) \in H_0(\mathcal{M})$ (using step 23).

This concludes the proof that all the steps in the construction are valid. A useful consequence of the construction (3.6) is that we can simulate $H_0(\mathcal{M})$ of *any* modular machine, within a finitely group; step 26 of (3.6) (via the subgroup membership problem: $\overline{t(\alpha,\beta)} \in \langle \bar{t}\rangle'$ iff $(\alpha,\beta) \in H_0(\mathcal{M})$). We use this when proving Higman's embedding theorem.

Definition 3.7.
If $G = \overline{\langle X|R\rangle}$ is an f.g. group, then we write $\mathrm{WP}(G)$ to denote the integers

representing the trivial words in G. That is, for a given encoding $F(X) = \{w_1, w_2, \ldots\}$, we have

$$\mathrm{WP}(G) := \{n \in \mathbb{N} \mid \overline{w}_n = e \text{ in } G\}$$

It should be clear that both $[\mathrm{WP}(G)]_m$ and $[\mathrm{WP}(G)]_T$ are independent of the presentation chosen, so long as it has a finite generating set.

So using (3.6) we immediately get the following result:

Theorem 3.8 (Boone-Britton-Novikov)**.**
Determining membership in $H_0(\mathcal{M})$ can be many-one reduced to the word problem for $G_{\mathcal{M}}$. That is, $H_0(\mathcal{M}) \leq_m \mathrm{WP}(G_{\mathcal{M}})$ A fortiori*, if $\mathcal{M}$ is taken with $H_0(\mathcal{M})$ nonrecursive via* (1.23) *and* (1.48)*, then $G_{\mathcal{M}}$ is a finitely presented group with IWP by* (1.29)*.*

There are some slightly more straightforward examples of groups with IWP, for example Borisov's construction [14] of one with 5 generators and 12 relators. Our construction above would yield a very complicated finite presentation.

3.3. The Higman embedding theorem.

The following standard result is immediate from (2.51), and we leave the proof as an exercise.

Theorem 3.9.
Let G be a finitely presented group, and $H \leqslant G$ a finitely generated subgroup. Then H is recursively presented.

Remarkably, recursively presented groups are *precisely* the set of finitely generated subgroups of finitely presented groups. Graham Higman proved this as his famous embedding theorem of 1961 [15]:

Theorem 3.10 (Higman 1961)**.**
Each recursively presented group embeds into some finitely presented group.

For the moment, we will show that for *each* recursively presented group H there is *some* corresponding finitely presented group G for which $H \hookrightarrow G$. Later, we will see that there is *one* finitely presented group into which *all* recursively presented groups embed[4].

Our approach will be to outline the entire construction first, and state (without proof) each of the intermediate results that we need. After this initial construction, we prove all the steps that are not obvious (these are denoted with a * in the construction below). This is taken from Cohen's book [13].

In a conventional flip[5], we take our Turing machines to start on the *rightmost* letter of an input word and read right to left, rather than the leftmost and read left to right. Functionally, these are the same. Furthermore, as per the paragraph preceding (1.48), we take q_0 as our initial state (as well as our halting state). This is so we can use the result of (1.48).

Construction 3.11.
The following gives a construction for embedding a recursively presented group into a finitely presented group.

(1) Let $C = \langle c_1, \ldots, c_n \mid S\rangle$ be a recursive presentation of a group, where S is the halting set of the Turing machine T.

[4]If you don't already have a headache, then this will give you one.

[5]Because the original source of this construction, Cohen's book [13], presents it as so.

(2) Re-write every word in $\{c_1, \ldots, c_n\}^*$ as being in the free monoid on $\{c_1, \ldots, c_{2n}\}$, where $c_{n+i} = c_i^{-1}$.
(3) *Observe that we can uniformly construct a *new* Turing machine T' whose halting set S' is *all* the trivial words in the group, when written in the free monoid on $\{c_1, \ldots, c_{2n}\}$. We make T' have an extra symbol s_0, different to $\{c_1, \ldots, c_{2n}\}$.
(4) Take the corresponding modular machine $\mathcal{M}$ for T' (1.48), with modulus m. We make sure to assign s_0 to 0, and c_i to i for all $1 \leq i \leq 2n$.
(5) *Observe that $m > 2n$, by construction of the modular machine $\mathcal{M}$.
(6) To each word $w = c_{i_k} c_{i_{k-1}} \cdots c_{i_0}$ we associate an m-ary representation $\alpha = \Sigma_{j=0}^{k} i_j m^j$, as per (1.48).
(7) Define $I := \{\alpha \in \mathbb{N} \mid \alpha \text{ represents a word}\}$. That is, $\alpha = \Sigma_{j=0}^{k} \beta_j m^j$, $1 \leq \beta_j \leq 2n$.
(8) For $\alpha \in I$, define $w_\alpha(c)$ to be the word formed from α.
(9) *Observe that $w_{\alpha m+i}(c) = w_\alpha(c) c_i$ for all $1 \leq i \leq 2n$, $\alpha \in I$.
(10) For $\alpha \in I$, write $w_\alpha(b)$, $w_\alpha(bc)$ for the words obtained from $w_\alpha(c)$ by replacing c_i with b_i and $b_i c_i$ respectively (where the b_i's are a new set of symbols).
(11) *Observe that, for all $\alpha \in I$, we have that $w_\alpha(c) \in S'$ iff $(\alpha, 0) \in H_0(M)$.
(12) Define the group $K_{\mathcal{M}}$ from step 13 in (3.6).
(13) Define $U = \{t, r_i \forall i, l_j \forall j\}$ (t, along with all the stable letters of $K_{\mathcal{M}}$).
(14) Define $t_\alpha := t(\alpha, 0)$.
(15) Define the free product

$$H_1 := K_{\mathcal{M}} * (\overline{C} \times \overline{\langle b_1, \ldots, b_n \mid - \rangle}) * \overline{\langle d | - \rangle}.$$

(16) Let $b_{n+i} = b_i^{-1}$ for $1 \leq i \leq n$.
(17) *Observe that $\{\overline{t_\alpha} | \alpha \in I\}$ and $\{\overline{t_\alpha w_\alpha(b) d} | \alpha \in I\}$ are each a free basis for the subgroup they generate in H_1, and thus generate isomorphic subgroups with isomorphism ψ given via extension of the map $\overline{t_\alpha} \mapsto \overline{t_\alpha w_\alpha(b) d}$ $\forall \alpha \in I$.
(18) With ψ as in step 17, define the following HNN extension with stable letter p:

$$H_2 := H_1 *_\psi$$

(19) Define the subgroup

$$A := \langle \overline{t}, \overline{x}, \overline{d}, \overline{b_j} (1 \leq j \leq n), \overline{p} \rangle \leqslant H_2$$

(20) *Observe that A is an HNN extension of the free group F with free basis $\{\overline{t}, \overline{x}, \overline{d}, \overline{b_j} (1 \leq j \leq n)\}$, with stable letter p sending $p t_\alpha p^{-1} = t_\alpha w_\alpha(b) d$ $\forall \alpha \in I$.
(21) For $1 \leq i \leq 2n$, define the subgroup

$$A_i := \langle \overline{t_i}, \overline{x}^m, \overline{b_i d}, \overline{b_j} (1 \leq j \leq n), \overline{p} \rangle \leqslant H_2$$

(22) *Observe that $\langle \overline{t_i}, \overline{x}^m \rangle \cap \langle \overline{t_\alpha} : \alpha \in I \rangle = \langle \overline{t_\beta} : \beta \in I \text{ with } \beta \equiv i \mod m \rangle$.
(23) *Observe that $\langle \overline{t_i}, \overline{x}^m, \overline{d}, \overline{b_j} (1 \leq j \leq n) \rangle \cap \langle \overline{t_\alpha} : \alpha \in I \rangle = \langle \overline{t_\beta} : \beta \in I \text{ with } \beta \equiv i \mod m \rangle$.
(24) *Observe that $\langle \overline{t_i}, \overline{x}^m, \overline{d}, \overline{b_j} (1 \leq j \leq n) \rangle \cap \langle \overline{t_\alpha w_\alpha(b) d} : \alpha \in I \rangle = \langle \overline{t_\beta w_\beta(b) d} : \beta \in I \text{ with } \beta \equiv i \mod m \rangle$.

(25) *Observe that, for all $1 \leq i \leq 2n$, A_i is an HNN extension of the free group on basis $\{\overline{t_i}, \overline{x}^m, \overline{d}, \overline{b_j}(1 \leq j \leq n)\}$, with stable letter p sending $pt_\alpha p^{-1} = t_\alpha w_\alpha(b)d \ \forall \alpha \in I$ with $\alpha \equiv i \mod m$.

(26) *Observe that $A \cong A_i$ for all $1 \leq i \leq 2n$, via ψ_i obtained by extension of the map $\overline{t} \mapsto \overline{t_i}$, $\overline{x} \mapsto \overline{x}^m$, $\overline{d} \mapsto \overline{b_i d}$, $\overline{b_j} \mapsto \overline{b_j}$ $\forall 1 \leq j \leq n$, $\overline{p} \mapsto \overline{p}$.

(27) Define the subgroup

$$A_+ := \langle \overline{U}, \overline{d}, \overline{b_j}(1 \leq j \leq n), \overline{p} \rangle \leq H_2$$

(28) Define the subgroup

$$A_- := \langle \overline{U}, \overline{d}, \overline{b_j c_j}(1 \leq j \leq n), \overline{p} \rangle \leq H_2$$

(29) *Observe that A_+ is an HNN extension with base group $\langle \overline{U}, \overline{d}, \overline{b_j}(1 \leq j \leq n) \rangle$ (the free product of $\langle \overline{U} \rangle$ and the free group with basis $\{\overline{d}, \overline{b_j}(1 \leq j \leq n)\}$), with stable letter p and HNN relations $pt_\alpha p^{-1} = t_\alpha w_\alpha(b)d \ \forall \alpha \in I$ with $\alpha \equiv i \mod m$.

(30) *Observe that $A_+ \cong A_-$, via ψ_+ which is obtained by extension of the map $\overline{u} \mapsto \overline{u}$ $\forall \overline{u} \in \overline{U}$, $\overline{d} \mapsto \overline{d}$, $\overline{b_j} \mapsto \overline{b_j c_j}$ $\forall 1 \leq j \leq n$, $\overline{p} \mapsto \overline{p}$.

(31) With the isomorphisms defined in steps 26 and 30, define the following HNN extension with stable letters $a_1, \ldots, a_{2n}, k$:

$$H_3 := H_2 *_{\psi_1, \ldots, \psi_{2n}, \psi_+}$$

(32) *Observe H_3 is finitely presented, and $\overline{C} \hookrightarrow H_3$.

We now prove all the nontrivial steps of Construction (3.10).

Step 3: We can uniformly construct a *new* Turing machine T' whose halting set is *all* the trivial words in the group, when written in the monoid on $\{c_1, \ldots, c_{2n}\}$.
First, using (2.51), as C is a recursive presentation, we can (uniformly) construct a new Turing machine T'' which enumerates *all* the trivial words in C. Note that T'' has input alphabet $\{c_1, \ldots, c_n, c_1^{-1}, \ldots, c_n^{-1}\}$ (as formal symbols). Now simply substitute the symbols $\{c_1^{-1}, \ldots, c_n^{-1}\}$ for $\{c_{n+1}, \ldots, c_{2n}\}$ in T'', and add a new symbol s_0 to be used as 'blank'. Call this new machine T'.

Step 5: $m > 2n$, by construction of the modular machine $\mathcal{M}$.
This follows immediately from (1.47) and (1.48), where we assign s_0 to 0, and c_i to i for all $1 \leq i \leq 2n$.

Step 9: $w_{\alpha m+i}(c) = w_\alpha(c)c_i$ for all $1 \leq i \leq 2n$, $\alpha \in I$.
This becomes obvious when we write out the words in full.

Step 11: For all $\alpha \in I$, we have that $w_\alpha(c) \in S$ iff $(\alpha, 0) \in H_0(M)$.
Note that $w_\alpha(c) \in S'$ iff T' halts on input of a tape with $w_\alpha(c)$ written on it (T' starts reading from the rightmost letter of $w_\alpha(c)$, in start state q_0 by our strange convention from (1.48)). The left associate of the instantaneous description $w_\alpha(c)q_0$ is $(\alpha, 0)$. So by the construction of $\mathcal{M}$ in (1.47) and (1.48), $w_\alpha(c) \in S'$ iff $\mathcal{M}$, on input $(\alpha, 0)$, eventually reaches $(0, 0)$; that is, iff $(\alpha, 0) \in H_0(M)$.

Step 17: $\{\overline{t_\alpha} | \alpha \in I\}$ and $\{\overline{t_\alpha w_\alpha(b)d} | \alpha \in I\}$ are each a free basis for the subgroup they generate in H_1, and thus generate isomorphic subgroups with isomorphism ψ given via extension of the map $\overline{t_\alpha} \mapsto \overline{t_\alpha w_\alpha(b)d}$ $\forall \alpha \in I$..

We showed in step 4 of (3.6) that $\{t_\alpha | \alpha \in I\}$ is a free basis. By the normal form theorem for free products (2.12), it follows that $\{t_\alpha w_\alpha(b)d | \alpha \in I\}$ is also a free basis, as $w_\alpha(b)d$ is from a different free factor to t_α in H_1. The two sets generate isomorphic (free) groups by (2.7), as they both have the same cardinality $|I|$.

Step 20: A is an HNN extension of the free group F with basis $\{\overline{t}, \overline{x}, \overline{d}, \overline{b_j}(1 \leq j \leq n)\}$, with stable letter p sending $pt_\alpha p^{-1} = t_\alpha w_\alpha(b)d \ \forall \alpha \in I$.
Clearly, F is free on the given basis, as it is the union of free basis elements in distinct factor groups of the free product H_1. To use the good subgroup theorem (2.45), we need to show that $\psi(\langle t_\alpha : \alpha \in I\rangle \cap F) = \langle t_\alpha w_\alpha(b)d : \alpha \in I\rangle \cap F$. But recall that $t_\alpha := t(\alpha, 0) = x^\alpha t x^{-\alpha}$. Thus it is clear that $\langle t_\alpha : \alpha \in I\rangle \subseteq F$, and similarly $\langle t_\alpha w_\alpha(b)d : \alpha \in I\rangle \subseteq F$. Hence $\langle t_\alpha : \alpha \in I\rangle \cap F = \langle t_\alpha : \alpha \in I\rangle$ and $\langle t_\alpha w_\alpha(b)d : \alpha \in I\rangle \cap F = \langle t_\alpha w_\alpha(b)d : \alpha \in I\rangle$, and so $\psi(\langle t_\alpha : \alpha \in I\rangle \cap F) = \psi(\langle t_\alpha : \alpha \in I\rangle) = \langle t_\alpha w_\alpha(b)d : \alpha \in I\rangle = \langle t_\alpha w_\alpha(b)d : \alpha \in I\rangle \cap F$. The rest now follows by the good subgroup theorem (2.45).

Step 22: $\langle \overline{t_i}, \overline{x}^m\rangle \cap \langle \overline{t_\alpha} : \alpha \in I\rangle = \langle \overline{t_\beta} : \beta \in I \text{ with } \beta \equiv i \mod m\rangle$.
From step 8 of (3.6) we have that $T \cap K_{a,b}^{M,N} = T_{a,b}^{M,N}$ in K. Written out, this is $\langle \{t(r,s)\}_{(r,s)\in\mathbb{Z}^2}\rangle \cap \langle t(a,b), x^M, y^N\rangle = \langle \{t(\alpha,\beta) \mid \alpha \equiv a \mod M,\ \beta \equiv b \mod N\}\rangle$. In particular, we have $\langle \{t(r,s)\}_{(r,s)\in\mathbb{Z}^2}\rangle \cap \langle t(i,0), x^m, y^0\rangle = \langle \{t(\alpha,\beta) \mid \alpha \equiv i \mod m,\ \beta \equiv 0 \mod 0\}\rangle = \langle \{t(\alpha,0) \mid \alpha \equiv i \mod m\}\rangle = \langle \{t_\alpha \mid \alpha \equiv i \mod m\}\rangle$. So now we have shown that $\langle t_i, x^m\rangle \cap \langle \{t(r,s)\}_{(r,s)\in\mathbb{Z}^2}\rangle = \langle t_\alpha : \alpha \equiv i \mod m\rangle$. Intersecting both sides with $\langle t_\alpha : \alpha \in I\rangle$ gives $\langle t_i, x^m\rangle \cap \langle \{t(r,s)\}_{(r,s)\in\mathbb{Z}^2}\rangle \cap \langle t_\alpha : \alpha \in I\rangle = \langle t_\alpha : \alpha \equiv i \mod m\rangle \cap \langle t_\alpha : \alpha \in I\rangle$. Observing that $\{t_\alpha : \alpha \in I\}$ is a subset of a free basis for T, we have that $\langle \{t(r,s)\}_{(r,s)\in\mathbb{Z}^2}\rangle \cap \langle t_\alpha : \alpha \in I\rangle = \langle t_\alpha : \alpha \in I\rangle$ by (2.16), and $\langle t_\beta : \beta \equiv i \mod m\rangle \cap \langle t_\alpha : \alpha \in I\rangle = \langle t_\alpha : \alpha \in I \text{ with } \alpha \equiv i \mod m\rangle$ by (2.16).

Step 23: $\langle \overline{t_i}, \overline{x}^m, \overline{d}, \overline{b_j}(1 \leq j \leq n)\rangle \cap \langle \overline{t_\alpha} : \alpha \in I\rangle = \langle \overline{t_\beta} : \beta \in I \text{ with } \beta \equiv i \mod m\rangle$.
This follows since $\langle t_\alpha : \alpha \in I\rangle$ is contained in the free factor $K_{\mathcal{M}}$ of H_1, and so $\langle t_i, x^m, d, b_j(1 \leq j \leq n)\rangle \cap \langle t_\alpha : \alpha \in I\rangle = \langle t_i, x^m\rangle \cap \langle t_\alpha : \alpha \in I\rangle = \langle t_\beta : \beta \in I \text{ with } \beta \equiv i \mod m\rangle$ (the final part coming from step 22).

Step 24: $\langle \overline{t_i}, \overline{x}^m, \overline{d}, \overline{b_j}(1 \leq j \leq n)\rangle \cap \langle \overline{t_\alpha w_\alpha(b)d} : \alpha \in I\rangle = \langle \overline{t_\beta w_\beta(b)d} : \beta \in I \text{ with } \beta \equiv i \mod m\rangle$.
This is similar to step 23. It is immediate that the containment $\supseteq$ holds. To show $\subseteq$, observe that $\langle t_i, x^m, d, b_j(1 \leq j \leq n)\rangle \cap \langle t_\alpha w_\alpha(b)d : \alpha \in I\rangle$ is contained in H_1. Now project onto the $K_{\mathcal{M}}$ factor of H_1; call this map $f : H_1 \twoheadrightarrow K_{\mathcal{M}}$. But now observe that $f(\langle t_i, x^m, d, b_j(1 \leq j \leq n)\rangle \cap \langle t_\alpha w_\alpha(b)d : \alpha \in I\rangle) = \langle t_i, x^m\rangle \cap \langle t_\alpha : \alpha \in I\rangle$, which by step 23 is equal to $\langle t_\beta : \beta \in I \text{ with } \beta \equiv i \mod m\rangle$. Thus, by (2.16), as the t_α's are a subset of a free basis of T, the only values of α we can have are $\alpha \equiv i \mod m$. So $\langle t_i, x^m, d, b_j(1 \leq j \leq n)\rangle \cap \langle t_\alpha w_\alpha(b)d : \alpha \in I\rangle = \langle t_i, x^m, d, b_j(1 \leq j \leq n)\rangle \cap \langle t_\alpha w_\alpha(b)d : \alpha \in I \text{ with } \alpha \equiv i \mod m\rangle$. But it is then immediate that $\langle t_\alpha w_\alpha(b)d : \alpha \in I \text{ with } \alpha \equiv i \mod m\rangle \subseteq \langle t_i, x^m, d, b_j(1 \leq j \leq n)\rangle$, and so this intersection is simply $\langle t_\alpha w_\alpha(b)d : \alpha \in I \text{ with } \alpha \equiv i \mod m\rangle$.

Step 25: A_i is an HNN extension of the free group on basis $\{\overline{t_i}, \overline{x}^m, \overline{d}, \overline{b_j}(1 \leq j \leq n)\}$, with stable letter p sending $pt_\alpha p^{-1} = t_\alpha w_\alpha(b)d \ \forall \alpha \in I$ with $\alpha \equiv i$

mod m.
We need to show, for each i, that $\psi(\langle t_i, x^m, d, b_j(1 \leq j \leq n)\rangle \cap \langle t_\alpha : \alpha \in I\rangle) = \langle t_i, x^m, d, b_j(1 \leq j \leq n)\rangle \cap \langle t_\alpha w_\alpha(b)d : \alpha \in I\rangle$. By steps 23 and 24, this equates to showing that $\psi(\langle t_\beta : \beta \in I \text{ with } \beta \equiv i \mod m\rangle) = \langle t_\beta w_\alpha(b)d : \beta \in I \text{ with } \beta \equiv i \mod m\rangle$, which is immediate from the definition of ψ. Now use (2.45).

Step 26: $A \cong A_i$ for all $1 \leq i \leq 2n$, via the map ψ_i which is an extension of the map sending $\bar{t} \mapsto \bar{t}$, $\overline{x} \mapsto \overline{x}^m$, $\overline{d} \mapsto \overline{b_i d}$, $\overline{b_j} \mapsto \overline{b_j}$ $\forall 1 \leq j \leq n$, $\overline{p} \mapsto \overline{p}$.
Clearly ψ_i is surjective, as it is an extension of a map from a generating set to a generating set (of the same cardinality). To see that ψ_i is a homomorphism, we need to verify that it preserves relations. The only relations of A are $pt_\alpha p^{-1} = t_\alpha w_\alpha(b)d$ for all $\alpha \in I$. But

$$\begin{aligned}\psi_i(pt_\alpha p^{-1}) &= p\psi_i(x^\alpha t x^{-\alpha})p^{-1} \\ &= px^{\alpha m}x^i t x^{-i}x^{-\alpha m}p^{-1} \\ &= pt_{\alpha m+i}p^{-1} \\ &= t_{\alpha m+i}w_{\alpha m+i}(b)d \\ &= t_{\alpha m+i}w_{\alpha m}(b)b_i d \quad \text{(by step 9)}\end{aligned}$$

and similarly $\psi_i(t_\alpha w_\alpha(b)d) = \psi_i(x^\alpha t x^{-\alpha})w_\alpha(b)b_i d = t_{\alpha m+i}w_\alpha(b)b_i d$.
To see that ψ_i is injective, take a nontrivial element $g \in A$ with $\psi_i(g) = e$. Since A is an HNN extension, then g has a reduced form; a freely reduced word w on $\{t, x, d, b_j(1 \leq j \leq n), p\}$ representing g which contains no p-pinch. But now consider the image word $\psi_i(w)$. If w contains no occurrence of p, then neither does $\psi_i(w)$; as w is freely reduced then so is $\psi_i(w)$ which is a contradiction as $\psi_i(w)$ is trivial in the base group $\langle\{t_i, x^m, d, b_j(1 \leq j \leq n)\}\rangle$ which is free on the given generators. If instead w contains an occurrence of p, then so does $\psi_i(w)$, which must thus contain a p-pinch as it is also freely reduced. But then the p-pinch in $\psi_i(w)$ can be 'pulled back' to a p-pinch in w in A as follows: Take w and evaluate ψ_i on the occurrences of $p^{\pm 1}$ only. So we are considering a word $\psi_i(v_1)p^{\epsilon_1}\psi_i(v_2)\dots p^{\epsilon_l}\psi_i(v_{l+1})$. So the pinch is of the form (without loss of generality) $p\psi_i(v_j)p^{-1}$ with $\psi_i(v_j) \in \langle t_\alpha w_\alpha(b)d : \alpha \in I\rangle$. Thus $v_j \in \langle t_\alpha : \alpha \in I\rangle$ (as we can consider p-pinches in terms of the HNN extension H_2). Thus pv_jp^{-1} is a subword of w, and is a p-pinch in A; a contradiction.

Step 29: A_+ is an HNN extension with base group $\langle \overline{U}, \overline{d}, \overline{b_j}(1 \leq j \leq n)\rangle$ (the free product of $\langle\overline{U}\rangle$ and the free group with basis $\{\overline{d}, \overline{b_j}(1 \leq j \leq n)\}$), with stable letter p.
First, observe that $\langle t\rangle' := \langle U\rangle$, from step 19 of (3.6). So, using step 22 of (3.6), we have that $\langle U\rangle \cap K = \langle t\rangle' \cap K = T_{\mathcal{M}} = \langle t(\alpha, \beta) : (\alpha, \beta) \in H_0(\mathcal{M})\rangle$. As $\langle t_\alpha : \alpha \in I\rangle \leq K$, it follows that $\langle U\rangle \cap \langle t_\alpha : \alpha \in I\rangle = \langle U\rangle \cap K \cap \langle t_\alpha : \alpha \in I\rangle = \langle t(\alpha, \beta) : (\alpha, \beta) \in H_0(\mathcal{M})\rangle \cap \langle t_\alpha : \alpha \in I\rangle = \langle t_\alpha : (\alpha, 0) \in H_0(\mathcal{M})\rangle$ (we are dealing with subsets of free bases, which are again free bases by (2.16)). We then apply this to obtain the following 2 facts:
1. By a very similar argument to that used in step 23, we have that $\langle U, d, b_j(1 \leq j \leq n)\rangle \cap \langle t_\alpha : \alpha \in I\rangle = \langle t_\alpha : (\alpha, 0) \in H_0(\mathcal{M})\rangle$.
2. By a very similar argument to that used in step 24, we have that $\langle U, d, b_j(1 \leq j \leq n)\rangle \cap \langle t_\alpha w_\alpha(b)d : \alpha \in I\rangle = \langle t_\alpha w_\alpha(b)d : (\alpha, 0) \in H_0(\mathcal{M})\rangle$.

Thus, by the definition of ψ from step 17, we have that $\psi(\langle U, d, b_j(1 \leq j \leq n)\rangle \cap \langle t_\alpha : \alpha \in I\rangle) = \psi(\langle t_\alpha : (\alpha, 0) \in H_0(\mathcal{M})\rangle) = \langle t_\alpha w_\alpha(b)d : (\alpha, 0) \in H_0(\mathcal{M})\rangle = \langle U, d, b_j(1 \leq j \leq n)\rangle \cap \langle t_\alpha w_\alpha(b)d : \alpha \in I\rangle$. So $\langle U, d, b_j(1 \leq j \leq n)\rangle$ is a good subgroup of H_2, and so by the good subgroup theorem (2.45), A_+ is an HNN extension with base group $\langle U, d, b_j(1 \leq j \leq n)\rangle$ (the free product of $\langle U\rangle$ and the free group with basis $\{d, b_j(1 \leq j \leq n)\}$), with stable letter p, and with relations $pt_\alpha p^{-1} = t_\alpha w_\alpha(b)d$ $\forall \alpha \in I$ with $(\alpha, 0) \in H_0(\mathcal{M})$.

Step 30: $A_+ \cong A_-$ via ψ_+, which is obtained by extension of the map $\overline{u} \mapsto \overline{u}$ $\forall \overline{u} \in \overline{U}$, $\overline{d} \mapsto \overline{d}$, $\overline{b_j} \mapsto \overline{b_j c_j}$ $\forall 1 \leq j \leq n$, $\overline{p} \mapsto \overline{p}$.
First, take the map $f : H_2 \to H_2$ where f sends each c_j to e, and maps each other generator of H_2 to itself. Then $f(A_-) = A_+$, and so this induces a homomorphism $\psi_- : A_- \to A_+$ which is actually a surjection since f (and thus ψ_-) send the given generators of A_- to the given generators of A_+ ($U \mapsto U$, $d \mapsto d$, $b_j c_j \mapsto b_j$, $p \mapsto p$). Now define the map $\psi_+ : A_+ \to A_-$ to be the 'reverse' of ψ_-, by extending the map $U \mapsto U$, $d \mapsto d$, $b_j \mapsto b_j c_j$, $p \mapsto p$. We show ψ_+ is a homomorphism; in doing so we will have proved that ψ_+ is an isomorphism as then ψ_+ and ψ_+ will be mutual inverses.
So, to show ψ_+ is a homomorphism, we must show that $\psi_+(pt_\alpha p^{-1}) = \psi_+(t_\alpha w_\alpha(b)d)$ for all $\alpha \in I$ with $(\alpha, 0) \in H_0(\mathcal{M})$. First, observe that $\psi_+(pt_\alpha p^{-1}) = pt_\alpha p^{-1}$ by definition of ψ_+. Next, observe that $\psi_+(t_\alpha w_\alpha(b)d) = t_\alpha w_\alpha(bc)d$, again by definition of ψ_+. Now, since b_j and c_l commute for all j, l, we have $w_\alpha(bc) = w_\alpha(c)w_\alpha(c)$. But, by step 11, we have that $w_\alpha(c) = e$ in C whenever $(\alpha, 0) \in H_0(\mathcal{M})$. Thus $\psi_+(pt_\alpha p^{-1}) = pt_\alpha p^{-1} = t_\alpha w_\alpha(b)d = t_\alpha w_\alpha(bc)d = \psi_+(t_\alpha w_\alpha(b)d)$ for all $\alpha \in I$ with $(\alpha, 0) \in H_0(\mathcal{M})$, and so ψ_+ is a homomorphism.

Step 32: H_3 is finitely presented, and $\overline{C} \hookrightarrow H_3$.
First note that H_3 is formed by starting with a free product H_1 of $K_\mathcal{M}$, $C \times F_n$, and $\mathbb{Z}$. Then we HNN extend 2 times, forming H_2 and H_3, with finitely many stable letters each. Thus H_3 is finitely generated, and contains an embedded copy of C.
The relations of H_3 consist of the following collections of words:
1. The finitely many relations of $K_\mathcal{M}$, the relations of the form $b_i c_j = c_j b_i$ for $1 \leq i, j \leq n$, the relation $ptp^{-1} = td$, and the finitely many relations involving the stable letters of the HNN extension H_3.
2. The relations $pt_\alpha p^{-1} = t_\alpha w_\alpha(b)d$ for all $\alpha \in I$ with $\alpha \neq 0$ (from the HNN extension H_2).
3. The set S' of all trivial words from our original group C. That is, $w_\alpha(c) = e$ for all $\alpha \in I$ with $(\alpha, 0) \in H_0(\mathcal{M})$ (by step 11).
We now proceed to show that all the relations of type 3 are consequences of those of types 1 and 2, and then to show that all the relations of type 2 are consequences of those of type 1.
First, by step 23 of (3.6), we see that if $(\alpha, 0) \in H_0(\mathcal{M})$ then t_α $(= t(\alpha, 0))$ lies in $\langle t\rangle'$, and thus can be written as a word over U. Moreover, this relationship comes as a consequence of the relations of $K_\mathcal{M}$ (type 1). Then, as a consequence of the relations involving k in H_3 (type 1), we have $kt_\alpha k^{-1} = t_\alpha$ if $(\alpha, 0) \in H_0(\mathcal{M})$ (as t_α can be written as a word over U precisely when $(\alpha, 0) \in H_0(\mathcal{M})$). We have, as a consequence of the relations of type 2, the relations $kw_\alpha(b)k^{-1} = w_\alpha(bc)$ for all $\alpha \in I$ (as we have $kb_j k^{-1} = b_j c_j$ for all j; type 1 relations, from the HNN extension H_3). Moreover, taking $\alpha \in I$ with

$(\alpha, 0) \in H_0(\mathcal{M})$, we have that $pt_\alpha p^{-1} = t_\alpha w_\alpha(b)d$ (type 2 relation). Observing that $kpk^{-1} = p$ and $kdk^{-1} = d$ are type 1 relations, and that we already have $kt_\alpha k^{-1} = t_\alpha$ as a consequence of relations of type 1, we see that, as a consequence of type 1 relations,

$$\begin{aligned} t_\alpha w_\alpha(b)d &= pt_\alpha p^{-1} \\ &= kpt_\alpha p^{-1}k^{-1} \\ &= kt_\alpha w_\alpha(b)dk^{-1} \\ &= t_\alpha k w_\alpha(b)k^{-1}d \\ &= t_\alpha w_\alpha(bc)d \\ &= t_\alpha w_\alpha(b)w_\alpha(c)d \end{aligned}$$

Thus $w_\alpha(c) = e$ is a consequence of the relations of types 1 and 2, for all $\alpha \in I$ with $(\alpha, 0) \in H_0(\mathcal{M})$. So all relations of type 3 are consequences of relations of types 1 and 2.

It remains to show that all the relations of type 2 are consequences of those of type 1. To begin, let $w_\alpha(a)$ be the word $w_\alpha(b)$ where we substitute $b_j \mapsto a_j$ $(1 \leq j \leq 2n)$. We induct on the length of $w_\alpha^R(a)$, the *reverse* of the word $w_\alpha(a)$, to show that $w_\alpha^R(a)tw_\alpha^R(a)^{-1} = t_\alpha$ and $w_\alpha^R(a)dw_\alpha^R(a)^{-1} = w_\alpha(b)d$ are consequences of the type 1 relations. For $|w_\alpha(a)| = 1$ we have $w_\alpha^R(a) = a_i$ (so $\alpha = i$), some $1 \leq i \leq 2n$. Thus:

$$\begin{aligned} w_\alpha^R(a)tw_\alpha^R(a)^{-1} &= a_i t a_i^{-1} = t_i = t_\alpha \\ w_\alpha^R(a)dw_\alpha^R(a)^{-1} &= a_i d a_i^{-1} = b_i d = w_\alpha(b)d \end{aligned}$$

are already type 1 relations (for $\alpha \in I$ with $\alpha \neq 0$).

Now, for the inductive step, we observe that a word of length 'one more that $w_\alpha^R(a)$' will be of the form $w_{\alpha m+i}^R(a)$, some $1 \leq i \leq 2n$ (step 9). Thus,

$$\begin{aligned} w_{\alpha m+i}^R(a)tw_{\alpha m+i}^R(a)^{-1} &= a_i w_\alpha^R(a)tw_\alpha^R(a)^{-1}a_i^{-1} \quad \text{(by step 5)} \\ &= a_i t_\alpha a_i^{-1} \quad \text{(a consequence of type 1 relations, by induction)} \\ &= a_i x^\alpha t x^{-\alpha} a_i^{-1} \\ &= x^{\alpha m} a_i t a_i^{-1} x^{-\alpha m} \quad (\text{as } a_i x a_i^{-1} = x^m\text{; a type 1 relation}) \\ &= x^{\alpha m} t_i x^{-\alpha m} \quad \text{(from above; a type 1 relation)} \\ &= t_{\alpha m+i} \end{aligned}$$

are all consequences of type 1 relations. Similarly,

$$\begin{aligned} w_{\alpha m+i}^R(a)dw_{\alpha m+i}^R(a)^{-1} &= a_i w_\alpha^R(a)dw_\alpha^R(a)^{-1}a_i^{-1} \quad \text{(by step 5)} \\ &= a_i w_\alpha(b)da_i^{-1} \quad \text{(a consequence of type 1 relations, by induction)} \\ &= a_i w_\alpha(b)a_i^{-1}b_i d \quad (\text{as } a_i d a_i^{-1} = b_i d\text{; a type 1 relation}) \\ &= w_\alpha(b)b_i d \quad (\text{as } a_i b_j a_i^{-1} = b_j \ \forall i, j\text{; a type 1 relation}) \\ &= w_{\alpha m+i}(b)d \quad \text{(by step 9)} \end{aligned}$$

are also all consequences of type 1 relations.

Now, it remains to show that the type 2 relations $pt_\alpha p^{-1} = t_\alpha w_\alpha(b)d$ for all $\alpha \in I$ with $\alpha \neq 0$ are consequences of the type 1 relations. We have $ptp^{-1} = td$ is a type 1 relation. Conjugating both sides by $w_\alpha^R(a)$, we consider the relation

$w_\alpha^R(a)ptp^{-1}w_\alpha^R(a)^{-1} = w_\alpha^R(a)tdw_\alpha^R(a)^{-1}$; a consequence of a type 1 relation. Then

$$\begin{aligned} w_\alpha^R(a)ptp^{-1}w_\alpha^R(a)^{-1} &= pw_\alpha^R(a)tw_\alpha^R(a)^{-1}p^{-1} \quad (\text{as } a_ipa_i^{-1} = p\ \forall 1 \le i \le 2n) \\ &= pt_\alpha p^{-1} \end{aligned}$$

using only consequences of type 1 relations. Similarly, we get

$$\begin{aligned} w_\alpha^R(a)tdw_\alpha^R(a)^{-1} &= w_\alpha^R(a)tw_\alpha^R(a)^{-1}w_\alpha^R(a)dw_\alpha^R(a)^{-1} \\ &= t_\alpha w_\alpha(b)d \end{aligned}$$

using only consequences of relations of type 1. Thus, as $ptp^{-1} = t$ is a type 1 relation, we have that $pt_\alpha p^{-1} = t_\alpha w_\alpha(b)d$ for all $\alpha \in I$ with $\alpha \neq 0$, as consequences of type 1 relations.
So we can discard all relations of H_3 of types 2 and 3, and are left with the (finite set of) type 1 relations. So H_3 is finitely presented.

This concludes the proof that all the steps in the construction are valid. By combining (3.9) and (3.10), we now state the complete classification of finitely generated subgroups of finitely presented groups.

Theorem 3.12.
Let H be a finitely generated group. Then H embeds into some finitely presented group iff H is recursively presented.

We will make extensive use of this theorem, and the ideas behind it, in later sections.

3.4. **Strictly preserving degrees.**
Our exposition of Sections §3.2 and §3.3, where we simulate a modular machine in a finitely presented group to make one with IWP, and then extend this to give Higman's Embedding Theorem, come from Cohen's book [13]. However, Cohen draws this from his original two papers with Aanderaa [16], [17]. In these two papers, Aanderaa and Cohen prove stronger results. They show that these groups which simulate modular machines do so without introducing any additional complexity. That is, they preserve the Turing degree of the machine they are simulating.
The following result is from [16].

Theorem 3.13 ([16, Theorem 2])**.**
i) Let T be a Turing machine, and $\mathcal{M}(T)$ the modular machine associated to it from (1.48)*. Then $\Omega(T) \equiv_m H_0(\mathcal{M}(T))$.*
ii) Let $\boldsymbol{d}$ be an r.e. many-one degree with $\boldsymbol{d} \neq [\mathbb{N}]_m$. Then there is a modular machine $\mathcal{M}_{\boldsymbol{d}}$ with $[H_0(\mathcal{M}_{\boldsymbol{d}})]_m = \boldsymbol{d}$.

Note. The numerical nature of modular machines makes it easy to show that the functions they compute are partial recursive (1.11). But by (1.48) we have that every Turing machine cam be simulated by a modular machine. Thus we obtain a fairly straightforward proof of (1.13); that every partial computable function is partial recursive. This is mentioned in [16, §1].

It turns out that the word problem in the group constructed in (3.6) is no harder than that of the halting set of the original modular machine that we started with. That is,

Theorem 3.14 ([16, Theorem 2])**.**
Given a Turing machine T, and successively constructing from it a modular machine $\mathcal{M}(T)$ as per (1.48) *and then a finitely presented group $G_{\mathcal{M}(T)}$ as per* (3.6)*, we have that*

$$\Omega(T) \equiv_m H_0(\mathcal{M}(T)) \equiv_T G_{\mathcal{M}(T)}$$

So this preserves the Turing degree of the halting set. In particular, for any r.e. Turing degree $\boldsymbol{d}$, we have that there is a finitely presented group G with $[\mathrm{WP}(G)]_T = \boldsymbol{d}$.

Aanderaa and Cohen are able to extend this result to their construction of the Higman embedding theorem, to show the following:

Theorem 3.15 ([17, Theorem B])**.**
Let C be a finitely generated recursively presented group, and H_3 the finitely presented group constructed from it in (3.11) *with $\overline{C} \hookrightarrow H_3$. Then*

$$\mathrm{WP}(\overline{C}) \equiv_T \mathrm{WP}(H_3)$$

The key idea in proving both (3.14) and (3.15) is the reverse of (2.62). It is possible to show (and Aanderaa and Cohen do show this in [16] and [17]) that at every stage of the HNN extensions constructed, the membership problems for the subgroups over which we are doing the HNN extensions are Turing reducible to the word problem in the base group (and we really do need Turing reduction here; we need to 'ask the word problem' several questions to deduce membership). That is, to decide if a word is trivial, we just go through and detect all the pinches, then reduce them, and continue until there are no more. So the word problem in the final group constructed in each of (3.6) and (3.11) can be 'wound back' incrementally (going back through every HNN extension) to the original modular machine / group that we started with

4. MORE EMBEDDING THEOREMS

4.1. **The Adian-Rabin construction.**

Now that we have *one* algorithmically undecidable problem relating to finitely presented groups, we can use some simple algebraic tricks to propagate this out and show that *many* other problems are algorithmically undecidable.

Several decision problems (like the word problem) are solvable in particular classes of finitely presented groups. For example, the word problem is solvable in finitely presented free groups, and finitely presented abelian groups. However, to use such algorithms, we need to *know* that we have a presentation of a group which is free, abelian, etc. So, can we recognise these properties? Well, no. And to show that we can't, we use a very clever result by Adian and Rabin[6].

Theorem 4.1 (Adian-Rabin)**.**
There is a construction that, on input of a presentation $P = \langle X|R\rangle$ with X countable, and a word $w \in F(X)$, produces a new presentation $P(w)$ and an explicit homomorphism $\phi : \overline{P} \to \overline{P(w)}$ such that:
1. *If $\overline{w} \neq e$ in $\overline{P}$, then ϕ is an embedding, and hence $\overline{P(w)} \neq \{e\}$.*
2. *If $\overline{w} = e$ in $\overline{P}$, then $\overline{P(w)} \cong \{e\}$.*
3. *$\overline{P(w)}$ can be generated by 2 elements.*
4. *If X, R are both r.e. sets, then $P(w)$ is a recursive presentation.*
5. *If X, R are both finite sets, then $P(w)$ is finite presentation.*
6. *In cases 4. and 5. above, the construction is algorithmic; we can obtain such a recursive/finite presentation directly from P and w.*

Proof. Take a presentation $P = \langle X|R\rangle = \langle x_1, x_2, \dots |R\rangle$ and form the free product $\overline{P} * F_2$ with presentation $Q := P * \langle a, b|-\rangle$. Now take another copy of F_2 with presentation $S := \langle c, d|-\rangle$.

Form the subgroup $A \leq \overline{P} * F_2$ by

$$A := \langle \overline{b}, \overline{aba^{-1}}, \overline{a^2bab^{-1}a^{-2}}, \overline{a^3[w,b]a^{-3}}, \overline{a^{(3+i)}x_iba^{-(3+i)}} \ \forall x_i \in X\rangle$$

If $\overline{w} \neq e$ in $\overline{P}$ then, by applying the normal form theorem for free products (2.12) in the same way as we did in (2.13), A is free on its given generating set, and so $A \cong F_\infty$ if $|X| = \infty$ or $A \cong F_{n+4}$ if $|X| = n < \infty$.
Similarly, form the subgroup $B \leq F_2$ by

$$B := \langle \overline{d}, \overline{cdcd^{-1}c^{-1}}, \overline{c^2dcd^{-1}c^{-2}}, \overline{c^3dc^{-3}}, \overline{c^{(3+i)}dc^{-(3+i)}} \ \forall x_i \in X\rangle$$

Just like in the case of A, we have that B is free on its given generating set, and so $B \cong F_\infty$ if $|X| = \infty$ or $B \cong F_{n+4}$ if $|X| = n < \infty$. Thus $A \cong B$.

Take the map $\varphi : A \to B$ given by extending the following bijection between generating sets (this is an isomorphism when $\overline{w} \neq e$ in $\overline{P}$):

$$\begin{aligned}
\overline{b} &\mapsto \overline{d} \\
\overline{aba^{-1}} &\mapsto \overline{cdcd^{-1}c^{-1}} \\
\overline{a^2bab^{-1}a^{-2}} &\mapsto \overline{c^2dcd^{-1}c^{-2}} \\
\overline{a^3[w,b]a^{-3}} &\mapsto \overline{c^3dc^{-3}} \\
\overline{a^{(3+i)}x_iba^{-(3+i)}} &\mapsto \overline{c^{(3+i)}dc^{-(3+i)}} \ \forall x_i \in X
\end{aligned}$$

[6]Done independently, and in a *much* more complicated way than what we present here.

Now form the amalgamated product $(\overline{P} * \overline{\langle a, b|-\rangle}) *_\varphi (\overline{\langle c, d|-\rangle})$. This has presentation $P(w)$ given by (after replacing d with b since $\varphi(\bar{b}) = \bar{d}$):
Generators: $X \cup \{a, b, c\}$.
Relations: R along with

$$\begin{aligned} aba^{-1} &= cbcb^{-1}c^{-1} \\ a^2bab^{-1}a^{-2} &= c^2bcb^{-1}c^{-2} \\ a^3[w, b]a^{-3} &= c^3bc^{-3} \\ a^{(3+i)}x_iba^{-(3+i)} &= c^{(3+i)}bc^{-(3+i)} \ \forall x_i \in X \end{aligned}$$

We now prove all the properties of $P(w)$ as stated in the theorem.

When $\overline{w} \neq e$ in $\overline{P}$, φ is an isomorphism, and so $\overline{P(w)}$ is an amalgamated product. So take ϕ to be the natural embedding $\overline{P} \hookrightarrow (\overline{P} * \overline{\langle a, b|-\rangle}) *_\varphi (\overline{\langle c, d|-\rangle}) = \overline{P(w)}$ by (2.37). Thus 1. is proved.

If $\overline{w} = e$ in $\overline{P}$, then A is not free on its generating set, so $\overline{P(w)}$ is not an amalgamated product. Moreover, one can see from the relators of $P(w)$ that, in the case $\overline{w} = e$ in $\overline{P}$, we have that $\overline{P(w)} \cong \{e\}$ (first see that $\bar{b} = e$, then $\bar{c} = e$, then $\bar{a} = e$, then $\overline{x_i} = e$ for all $x_i \in X$). Thus 2. is proved.

We only need $\bar{b}$ and $\overline{a^{-1}c}$ to generate $\overline{P(w)}$; this follows immediately from the relations of $P(w)$ as we have $c = b^{-1}(a^{-1}c)^{-1}b(a^{-1}c)b$, and then $a = c(a^{-1}c)^{-1}$, and then $x_i = a^{-(3+i)}c^{(3+i)}bc^{-(3+i)}a^{(3+i)}$. Thus 3. is proved

Suppose both X and R are r.e. The relations of $P(w)$ are R (which is r.e.), along with the finite set $\{aba^{-1} = cbcb^{-1}c^{-1}, a^2bab^{-1}a^{-2} = c^2bcb^{-1}c^{-2}, a^3[w, b]a^{-3} = c^3bc^{-3}\}$, and the infinite set $\{a^{(3+i)}x_iba^{-(3+i)} = c^{(3+i)}bc^{-(3+i)}\}_{x_i \in X}$ (which is r.e. since X is r.e.). By (1.24), as the relations of $P(w)$ are the union of finitely many r.e. sets, then they form an r.e. set. We add a generator t, and add the relation $t = a^{-1}c$. Using 3., we can (algorithmically) re-write the r.e. set of relations of $P(w)$ using just b and t, as these generate the group. We then discard all other generators apart from b and t. This re-written presentation has only 2 generators. Thus $\overline{P(w)}$ is recursively presented, and so 4. is proved.

Similarly, if X, R are finite, then the re-writing done to prove 4. gives a 2-generator finite presentation for $\overline{P(w)}$. Thus 5. is proved.

In the proofs of cases 4. and 5., the constructions and re-writings given are clearly algorithmic. Thus 6. is proved. □

The following consequence, originally proved by Higman, Neumann and Neumann in 1949 [20] but now a corollary of (4.1), is a predecessor to the Higman embedding theorem. Their original proof was the first use of HNN extensions in mathematics.

Theorem 4.2 (Higman, Neumann, Neumann, 1949)**.**
Every countable group C embeds into some 2-generator countable group G_C; if C is recursively (resp. finitely) presented, then G_C is also.

Proof. Take a presentation P for C, and form the free product presentation $Q := P * \langle t|-\rangle$ of the group $C * \mathbb{Z}$. Now form the presentation $Q(t)$ as per (4.1); the theorem now follows immediately as $\bar{t} \neq e$ in $\overline{Q}$. □

A group G is said to be *SQ-universal* if every countable group C embeds in some quotient of G. The above shows that F_2 is SQ-universal, and was in fact the first example of a finitely generated SQ-universal group.

The above construction has many additional interesting properties[7]. Miller [22] uses it to show that there is a finitely presented group G with IWP for which *every* non-trivial quotient of G has IWP (so G is not simple). Houcine [21] shows, amongst other things, that the above construction preserves the Turing degree of the word problem (in the case $\overline{w} \neq e$).

4.2. **Markov properties.**

The first way that we apply (4.1) is to show that the *triviality problem*, of recognising if a finite presentation P defines the trivial group or not, is unsolvable.

Theorem 4.3.
There is no algorithm that, on input of a finite presentation P, decides if $\overline{P} \cong \{e\}$ or not.

Proof. Take a finite presentation $P = \langle X|R\rangle$ of a finitely presented group with IWP (3.8); noting that the proof there allows us to construct an *explicit* such presentation. Now, for each word $w \in F(X)$, we can construct the finite presentation $P(w)$ as per (4.1). Now simply observe that $\overline{w} = e$ in $\overline{P}$ iff $\overline{P(w)} \cong \{e\}$, so an algorithm to determine if an arbitrary finite presentation defined the trivial group or not would allow us to solve the word problem for $\overline{P}$; a contradiction. □

This immediately implies the following (known as the *isomorphism problem* for finitely presented groups).

Corollary 4.4.
There is no algorithm that, on input of two finite presentations P_1, P_2, decides if $\overline{P_1} \cong \overline{P_2}$ or not.

So we can't tell if a finite presentation P gives the trivial group or not, and we can't tell if two finite presentations P_1, P_2 give the same group or not. But can we recognise other properties?

Definition 4.5.
A property of groups ρ is an *algebraic property* if it is invariant under isomorphism.

Definition 4.6.
We call an algebraic property ρ of finitely presented groups a *Markov property* if there exist two finitely presented groups G_+, G_- such that:
1. G_+ has the property ρ.
2. G_- does not have the property ρ, nor does it embed in any finitely presented group with the property ρ.

We can tie this in with the Adian-Rabin construction (4.1) to show that many properties of finitely presented groups are algorithmically unrecognisable.

Theorem 4.7 (Markov)**.**
It is impossible to algorithmically recognise any Markov property amongst finitely presented groups. That is, for any fixed Markov property ρ, there is no algorithm that, on input of a finite presentation P, determines if $\overline{P}$ has that property or not. In particular, every Markov property is Σ_1^0-hard.

[7]Which you will learn about in great detail if you take the Part III essay that I set.

Proof. Let ρ be a Markov property, with G_+, G_- as above (with finite presentations P_+, P_- respectively). Let $Q = \langle X|R\rangle$ be a finite presentation of a group with unsolvable word problem from (3.8) ($\mathrm{WP}(\overline{Q})$ can be taken to be Σ_1^0-complete, by (3.14)). Given any word $w \in F(X)$, we can form the finite presentation $P_+ * \big((P_- * Q)(w)\big)$ as per (4.1). Then we have that:
1. If $\overline{w} = e$ in $\overline{Q}$, then $\overline{w} = e$ in $\overline{P_- * Q}$, and so $\overline{(P_- * Q)(w)} \cong \{e\}$. That is, the Adian-Rabin construction applied to $P_- * Q$ with word w gives the trivial group. Hence $\overline{P_+ * \big((P_- * Q)(w)\big)} \cong \overline{P}_+$, and so has property ρ.
2. If $\overline{w} \neq e$ in $\overline{Q}$, then $\overline{w} \neq e$ in $\overline{P_- * Q}$, and so $\overline{P}_- \hookrightarrow \overline{P_- * Q} \hookrightarrow \overline{(P_- * Q)(w)} \hookrightarrow \overline{P_+ * \big((P_- * Q)(w)\big)}$. So $\overline{P_+ * \big((P_- * Q)(w)\big)}$ does not have property ρ, as $\overline{P}_-$ embeds into it.
So in summary, $\overline{P_+ * \big((P_- * Q)(w)\big)}$ has property ρ if and only if $\overline{w} = e$ in $\overline{Q}$. But $\overline{Q}$ is a group with IWP, so the problem of recognising ρ groups amongst finitely presented groups is algorithmically unsolvable. Moreover, we have a many-one reduction

$$\mathrm{WP}(\overline{Q}) \leq_m \{P \mid \overline{P} \in \rho\}$$

So $\{P \mid \overline{P} \in \rho\}$ must be Σ_1^0-hard, as $\mathrm{WP}(\overline{Q})$ is. □

Lemma 4.8.
The following is a (non-exhaustive) list of Markov properties, and hence all are algorithmically unrecognisable amongst finitely presented groups. We give (G_+, G_-) in each case:
Being trivial: $(\{e\}, \mathbb{Z})$
Being finite: $(\{e\}, \mathbb{Z})$
Being free: (F_2, C_2)
Being abelian: $(\mathbb{Z}, F_2)$
Being cyclic: $(\mathbb{Z}, F_2)$
Being solvable: $(\mathbb{Z}, F_2)$
Being nilpotent: $(\mathbb{Z}, F_2)$
Being torsion free: $(\mathbb{Z}, C_2)$
Being torsion: $(C_2, \mathbb{Z})$
Having solvable word problem: $(C_2,$ Group with unsolvable word problem $)$
Being linear: $(C_2,$ Group with unsolvable word problem $)$
Being hyperbolic: $(C_2, \mathbb{Z} \times \mathbb{Z})$
Being simple: $(C_2,$ Group with unsolvable word problem $)$
Being residually finite: $(C_2,$ Group with unsolvable word problem $)$
Being a 3-manifold group: $(\mathbb{Z},$ Group with unsolvable word problem $)$
Having polynomial growth: $(\{e\}, F_2)$

4.3. **Subgroup membership problem.**

In the previous section we constructed an f.p. group with IWP by first constructing an f.p. group with a subgroup with IMP, and then performing an HNN extension. Of course, the new group also has a subgroup with IMP (the original subgroup, but also $\{e\}$). We will now see that there exists a (very straightforward) f.p. group, namely $F_2 \times F_2$, with SWP which posesses a finitely generated subgroup with IMP. This was first done by Mihailova [23].

Theorem 4.9.
Let M be any group with a given set of generators $S = \{s_1, \ldots, s_n\}$ having quotient group $\overline{H}$ with presentation $H := \langle s_1, \ldots, s_n \mid r_1, \ldots, r_m\rangle$ on the images

of the given generators of M. Let $G = M \times M$ be the direct product of two copies of M, and let L_H be the subgroup of G generated by the elements

$$L_H := \langle (\overline{s}_1, \overline{s}_1), \ldots, (\overline{s}_n, \overline{s}_n), (\overline{r}_1, e), \ldots, (\overline{r}_m, e) \rangle$$

Then for any pair of words $u, v \in F(S)$ in the given generators, we have

$$(\overline{u}, \overline{v}) \in L_H \;\Leftrightarrow\; \overline{u} = \overline{v} \text{ in } \overline{H}$$

Proof. If $(\overline{u}, \overline{v}) \in L_H$ then clearly $\overline{u} = \overline{v}$ in $\overline{H}$ since this is true for each of the generators of L_H and H is a quotient group of M (recall that $\overline{r_i} = e$ in $\overline{H}$ for all i).

For the converse, first suppose $\overline{w} = e$ in $\overline{H}$ for some $w \in F(S)$. Then

$$w = \prod_{k=1}^{r} z_k(s_i) r_{i_k}^{\pm 1} z_k(s_i)^{-1} \text{ in } F(S)$$

for suitable words $z_k(s_i) \in F(S)$ (by (2.18) and the comment before it). But then we have

$$\begin{aligned}(w, e) &= \big(\prod_{k=1}^{r} z_k(s_i) r_{i_k}^{\pm 1} z_k(s_i)^{-1}, e \big) \text{ in } F(S) \times F(S) \\ &= \prod_{k=1}^{r} z_k((s_i, s_i))(r_{i_k}, e)^{\pm 1} z_k((s_i, s_i))^{-1} \text{ in } F(S) \times F(S) \\ &\in L_H \text{ in } G = M \times M\end{aligned}$$

as desired, where $z_k((s_i, s_i))$ is obtained from the word $z_k(s_i)$ by replacing all instances of s_i with (s_i, s_i). So if $\overline{w} = e$ in M, then $(\overline{w}, e) \in L_H$ in G.

Now consider the general case of $\overline{u} = \overline{v}$ in $\overline{H}$. Then $\overline{uv^{-1}} = e$ in $\overline{H}$ and so by what we just shown we have that $(\overline{uv^{-1}}, e) \in L_H$. Since L_H contains the diagonal of $G = M \times M$, it contains $(\overline{v}, \overline{v})$ and hence we also have $(\overline{uv^{-1}}, e) \cdot (\overline{v}, \overline{v}) = (\overline{u}, \overline{v}) \in L_H$. So we're done. □

Theorem 4.10.
Let M be a finitely presented group having a quotient group H with IWP. Then the group $G := M \times M$ has a finitely generated subgroup L_H with IMP in G.

Proof. By (4.9), and using the same notation from there, we have that

$$(\overline{w}, 1) \in L_H \text{ in } G \;\Leftrightarrow\; \overline{w} = e \text{ in } \overline{H}$$

As H has IWP, then L_H has IMP in G. □

Corollary 4.11 (Mihailova [23])**.**
Let G be any f.p. group with IWP (i.e., from (3.8)), and suppose that it has rank m. Then the group $G := F_m \times F_m$ has a finitely generated subgroup L with IWP in G.

As every finitely presented group embeds into a 2-generator finitely presented group (4.1), we have that there exists a 2-generator group with IWP, and so by the above we have that $F_2 \times F_2$ has a finitely generated subgroup L with IWP.

Lemma 4.12.
With $M = F_n \times F_n$ and H as in (4.9), we have that

$$\overline{H} = \{e\} \Leftrightarrow L_H = F_n \times F_n$$

Proof. This follows immediately from (4.9), as we have $(\overline{u}, \overline{v}) \in L_H \Leftrightarrow \overline{u} = \overline{v}$ in $\overline{H}$. □

Definition 4.13.
We define the *generating problem* (abbreviated to GP) for a recursive presentation $P = \langle X|R\rangle$ of a group as follows:

$$\text{Given a set of words } w_1, \ldots, w_n \in F(X), \text{ is } \langle \overline{w}_1, \ldots, \overline{w}_n\rangle = \overline{P}?$$

That is, do $\overline{w}_1, \ldots, \overline{w}_n$ generate $\overline{P}$? Groups which have a recursive presentation for which there exists an algorithm to solve the generating problem are said to have *solvable generating problem*, abbreviated to SGP.

Now we make use of several of our earlier results to show the following:

Theorem 4.14.
For any $n \geq 2$, the group $F_n \times F_n$ has generating problem which is Σ_1^0-complete, and thus $F_n \times F_n$ has IGP.

Proof. Take any f.p. group with IWP (3.8). By (4.1), this embeds in a 2-generator group, which thus has IWP by (2.54). By adding $n-2$ superfluous generators and relators, we can thus form a finite presentation with n generators, of a group with IWP. Call this presentation H.

Now observe by (4.12) that $\overline{H} = \{e\} \Leftrightarrow L_H = F_n \times F_n$. But $L_H = \langle (\overline{s}_1, \overline{s}_1), \ldots, (\overline{s}_n, \overline{s}_n), (\overline{r}_1, e), \ldots, (\overline{r}_m, e)\rangle$. Thus we have a many-one reduction from the triviality problem for finitely presented groups to the generating problem for $F_n \times F_n$, the former of which is Σ_1^0-hard by (4.7).

To see that the generating problem for $F_n \times F_n$ is in Σ_1^0, take an arbitrary collection of words $w_1, \ldots, w_n$ in $F_n \times F_n$, and use (2.60) to start testing if all of the standard generators of $F_n \times F_n$ lies in $\langle \overline{w}_1, \ldots, \overline{w}_n\rangle$; this will halt iff $w_1, \ldots, w_n$ generates $F_n \times F_n$. □

The problem of determining generating sets of $F_n \times F_n$ relates closely to the (now-resolved) Poincare conjecture.

4.4. **Universality.**

We now state the following remarkable consequence of the HNN embedding theorem (4.2) and Higman's embedding theorem (3.10).

Theorem 4.15.
There is a universal finitely presented group*. That is, a finitely presented group which contains an embedded copy of every finitely presented group.*

Proof. Fix a countably infinite alphabet $X = \{x_1, x_2, \ldots\}$. Now, for each n, we can construct all finite presentations of 'length n' (i.e., have precisely n letters appearing as generators/relations), and we can order these lexicographically (take the ordering on X and then read across the presentation from left to right as if it was a string of letters only; adopt the convention $x_i < x_i^{-1} < x_{i+1}$). Thus, by ordering all the presentations of length 1, then of length 2, and so on, we have an enumeration of finite presentations $P_1, P_2, \ldots$. Observe that, from j, we can algorithmically construct the presentation P_j. Moreover, every finitely presented group is isomorphic to the group given by one such presentation (perhaps many).
So now form the presentation $P = P_1 * P_2 * \ldots$. As we can construct P_j from j, then we have that P has an r.e. set of generators, and an r.e. set of relations. So

by (4.2), we have that $\overline{P}$ embeds into a recursively presented group G. Finally, we use (3.10) to embed G into a finitely presented group H. Now, as every finitely presented group embeds into $\overline{P}$, then they all embed into H also. □

A result by Boone and Collins [24] shows that *every* finitely presented group can be (uniformly) embedded into a finitely presented group with 13 generators and 33 relations (and later improved by Collins [25] to 8 generators and 26 relations). Thus there is a universal f.p. group with 8 generators and 26 relators, though we don't have a "complete presentation" for it yet.

Here is the 13 generator 33 relation example; everything is pre-determined except for the integers r, t, and the two words $\tau(M)^{\#}, \tau(N)$ which (ultimately) depend on our original finite presentation P. For a full exposition of this, see [26].

Theorem 4.16.
There is a uniform algorithm for embedding a finitely presented group A into another finitely presented group K_4 with 13 generators and 33 relations, given by the presentation

$$s_1, s_2, q, t, k, a, d, c_1, c_2, b_1, b_2, f, h;$$

$$s_j^{-1} a s_j = a \qquad s_j^{-1} d s_j = d^6 a d^6$$

$$q^{-1}(dads_2^{-1}s_1^{-2})q = s_1^2 s_2 s_1 s_2 dad$$

$$q^{-1}(d^2ad^2s_2^{-1}s_1^{-2})q = s_1^2 s_2^2 d^2 a d^2$$

$$q^{-1}(d^3ad^3\tau(M)^{\#})q = \tau(N) d^3 a d^3$$

$$q^{-1}(d^{3+j}ad^{3+j}s_j^{-1})q = s_j d^{3+j} a d^{3+j}$$

$$ta = at \qquad td = dt$$

$$ka = ak \qquad kd = dk$$

$$k^{-1}(\Phi_0^{-1} q^{-1} t q \Phi_0) k = (\Phi_0^{-1} q^{-1} t q \Phi_0)$$

$$b_i^{-1} s_j b_i = s_j \qquad b_i^{-1} c_j b_i = c_j \qquad b_i^{-1} k_0 b_i = k_0 c_i^{-1}$$

$$f^{-1}\theta(a_i)^{\epsilon} b_i^{\epsilon} f = \theta(a_i)^{\epsilon} \qquad f^{-1} k_0 f = k_0$$

$$h^{-1} t h = tf \qquad h^{-1} k_0 h = k_0 \qquad h^{-1} s_j h = s_j$$

for all $i = 1, 2$, $j = 1, 2$, $\epsilon = \pm 1$ and with the shorthand

$$\Phi_0 := \chi(p) s_2 s_1^{2t} \ (= s_2 s_1 s_2 s_1 s_2^6 s_1 s_2^{r-3} s_1^{2t})$$
$$k_0 := (q\Phi_0) k (q\Phi_0)^{-1}$$

where the integers r, t, and the words $\tau(M)^{\#}, \tau(N)$ are determined by our original input finite presentation for A.

We also have the following observation, which we leave as an exercise.

Theorem 4.17.
The property of 'not being a universal finitely presented group' is a Markov property. Thus, a finitely presented group G is a universal finitely presented group iff G posesses no Markov property.

4.5. **The algebraic characterisation of the word problem.**

Our current definition of a group with solvable word problem (2.50) makes references to algorithms, and thus to mathematical logic. However, there are completely *algebraic* ways to define such groups, with no mention of algorithms whatsoever. This was first done by Boone and Higman, and is often referred to as the Boone-Higman theorem [27].

Theorem 4.18 (Boone, Higman)**.**
A finitely generated group G has solvable word problem iff G embeds into some simple group S, which in turn embeds into some finitely presented group H. That is,

$$G \text{ has SWP } \Leftrightarrow\ G \hookrightarrow S^{(\text{simple})} \hookrightarrow H^{(\text{f.p.})}$$

Proof.
($\Leftarrow$) :
We begin by following the proof of (2.57). If $G = \{e\}$, then the result holds trivially. So, suppose $G \neq \{e\}$ embeds in a simple subgroup S of a finitely presented group H. As G is f.g. and embeds in an f.p. group, then by (3.9) it must be recursively presented. Thus by (2.51) the set of trivial words in G is r.e.; by (2.52) it remains to show that the set of non-trivial words of G is also r.e. Let $P = \langle X|R\rangle$ be a finite presentation of G, and $Q = \langle Y|T\rangle$ a finite presentation of H. Let $\theta : S \hookrightarrow H$ be an embedding of S into H, and $\psi : G \hookrightarrow H$ an embedding of G into H with $\psi(G) \leq \theta(S)$. Now fix a word $s \in F(Y)$ such that $\overline{s}$ is non-trivial and lies $\theta(S)$. For $w \in F(X)$, define the finite presentation $Q_w := \langle Y|T, \psi(w)\rangle$ and group $H_w := \overline{Q}_w$. Then H_w is the quotient of H by the normal closure of $\psi(\overline{w})$. As $\theta(S)$ is simple, if $\overline{\psi(w)} \neq e$ in H then the image of $\theta(S)$ is trivial in H_w. In particular, the image of s will be trivial in H_w. Thus $\overline{w} \neq e$ in G iff $\overline{s} = e$ in H_w. But H_w is f.p., so the trivial words in H_w will be r.e., by (2.51). Thus the non-trivial words in G are r.e.
($\Rightarrow$) :
Suppose G is f.g. with SWP, say with finitely generated presentation $P = \langle X|R\rangle$. Consider the set

$$\{(u_1, v_1), (u_2, v_2), \ldots\}$$

of all pairs of words $(u_i, v_i) \in F(X) \times F(X)$ which are non-trivial ($\overline{u}_i, \overline{v}_i \neq e$ in $\overline{P}$). Then this set is recursive, as it is the Cantor pairing (1.8) of two recursive sets; the set of non-trivial words in G is recursive by (2.52). So now we define the (infinite) HNN extension

$$G_1 := \langle G; x_1, t_i (i \geq 1) \mid t_i u_i x_1 u_i x_1^{-1} t_i^{-1} = v_i x_i u_i x_i^{-1} \rangle$$

By the normal form theorem for free products (2.12), the elements $\overline{u_i x_1 u_i x_1^{-1}}$ and $\overline{v_i x_i u_i x_i^{-1}}$ each have infinite order in $G * \overline{\langle x_1|-\rangle}$, as the normal form for $(\overline{u_i x_1 u_i x_1^{-1}})^n$ is $(u_i x_1 u_i x_1^{-1})^n$, and similarly for $\overline{v_i x_i u_i x_i^{-1}}$. Moreover, each of these cyclic subgroups has SMP in $G * \overline{\langle x_1|-\rangle}$ (To see this, take any word and convert it to normal form in the free product; this is possible as both G and $\overline{\langle x_1|-\rangle}$ have SWP. Then see if that normal form is exactly a power of the generator of the cyclic subgroup). So G_1 is an HNN extension of $G * \overline{\langle x_1|-\rangle}$. Moreover, by (2.61) we have that $G * \overline{\langle x_1|-\rangle}$ has SWP, and combining what we have just shown with (2.62) gives that G_1 has SWP.

We define G_{k+1} inductively from G_k as in the previous paragraph: replace G with G_k and G_1 with G_{k+1}. Observe that $G_k \leq G_{k+1}$ for all k. Now set

$$S := \bigcup_{k=1}^{\infty} G_k$$

Our construction of G_{k+1} from G_k was completely uniform. Thus we can algorithmically construct a countably generated recursive presentation for S, and moreover S has SWP (Note that S is infinitely generated, but is also recursively presented as each G_k is).

To see that S is simple, take any two non-trivial elements $u, v \in S$. Then there is some $k-1$ such that $u, v \in G_{k-1}$. Thus there is a stable letter p such that

$$pux_kux_k^{-1}p^{-1} = vx_kux_k^{-1}$$

Thus $v = pux_kux_k^{-1}p^{-1}x_ku^{-1}x_k^{-1}$, and so v lies in the normal closure of u. So the normal closure of any element in S contains all of S, and thus S is simple.

Finally, we can use (4.1) to embed S into a 2-generator recursively presented group K, and then (3.10) to embed K into a finitely presented group H. □

This result was strengthened by Thompson [28]; the proof of which is highly non-trivial and uses group actions on trees.

Theorem 4.19 (Thompson)**.**
A finitely generated group G has solvable word problem iff G embeds into some finitely generated *simple group S, which in turn embeds into some finitely presented group H. That is,*

$$G \text{ has SWP } \Leftrightarrow\ G \hookrightarrow S^{(\text{f.g. simple})} \hookrightarrow H^{(\text{f.p.})}$$

The following is a longstanding open problem:

Question 4.20.
Is it true that a finitely generated group G has solvable word problem iff G embeds into some *finitely presented* simple group S. That is,

$$G \text{ has SWP } \Leftrightarrow\ G \hookrightarrow S^{(\text{f.p. simple})}$$

There is another algebraic characterisation of finitely generated groups with solvable word problem, proved by Houcine[8] in [21]:

Theorem 4.21 (Houcine)**.**
A finitely generated group G has SWP if and only if there exists a finitely presented group H such that G is embeddable in every non-trivial quotient of H.

4.6. **Degrees of various incomputable problems.**

There are *very few* group properties which are recursive amongst finite presentations of groups. One of these is the property of being *perfect* (i.e., having trivial abelianisation).

Lemma 4.22.
The word problem is uniformly solvable amongst all finite presentations of abelian groups.

From this, we can immediately deduce the following result.

[8]Again, to be covered in the Part III essay I have set.

Corollary 4.23.
There set of finite presentations of perfect groups is recursive.

By (4.7), every Markov property is Σ_1^0-hard. We now try and investigate precisely where some common group properties sit on the arithmetic hierarchy.

Theorem 4.24.
The following group properties are all r.e. (i.e., in Σ_1^0) amongst finite presentations, and by (4.7) *are thus Σ_1^0-complete as they are all Markov properties: Trivial, free, finite, abelian, nilpotent, polycyclic.*

Proof. These all follow from the definitions of these groups; to verify the property in question requires carrying out an algorithmic test which is guaranteed to halt if the group posesses that property. We leave the details as a (long and involved) exercise. □

Here are two results by Boone[9] and Rogers, which show how intractable the word problem in groups really is.

Theorem 4.25 ([29, Theorem 1])**.**
The set of finite presentations of groups with SWP is Σ_3^0-complete.

Theorem 4.26 ([29, Theorem 2])**.**
There is no uniform algorithm to solve the word problem in set of finite presentations of groups with SWP.

There are very few other group theoretic properties which have been *precisely* located on the arithmetic hierarchy. Torsion-freeness is one of these.

Theorem 4.27 ([30, Theorem 4.2])**.**
The set of finite presentations of torsion-free groups is Π_2^0-complete.

Question 4.28.
The following properties of finitely presented groups have not been precisely located on the arithmetic hierarchy:
Solvable ($\in \Sigma_3^0$).
Residually finite ($\in \Pi_2^0$).
Simple ($\in \Pi_2^0$).

Later we will address the property of 'having a non-trivial finite quotient'. This turns out to be Σ_1^0-complete, but the proof of this is very difficult.

By studying torsion, and in particular the way that the construction (3.11) of the Higman embedding theorem preserves torsion, one can prove many interesting incomputability results in group theory, as was done in [30]. Here are some of them.

Theorem 4.29 ([30, Theorem 3.10])**.**
There is a universal finitely presented torsion-free group G. That is, G is torsion-free, and every finitely presented torsion-free group embeds in G. Thus by (4.27)*, the finitely presented subgroups of this finitely presented group G form a Π_2^0-complete set. That is, $\{P \mid \overline{P} \hookrightarrow G\}$ is Π_2^0-complete.*

[9]As you have probably noticed, William Boone features a lot in these notes. He did a great deal of work in this area from 1950-1980. His PhD student was Charles Miller III, who was my PhD supervisor.

Theorem 4.30 ([30, Proposition 4.4])**.**
For any fixed prime p, the set of finite presentations into which C_p embeds is Σ_2^0-complete. That is, $\{P \mid C_p \hookrightarrow \overline{P}\}$ is Σ_2^0-complete.

With the above results, it is not hard to deduce the following:

Theorem 4.31 ([30, Theorem 4.5])**.**
Take an enumeration $P_1, P_2, \ldots$ of all finite presentations of groups. Then the set $K = \{(i,j) \in \mathbb{N}^2 \mid \overline{P}_i \hookrightarrow \overline{P}_j\}$ is Σ_2^0-hard, Π_2^0-hard, and has a Σ_3^0 description (and thus is neither Σ_2^0 nor Π_2^0).

We write $\mathrm{Tord}(G)$ to denote the orders of non-trivial torsion elements of a group G, and say a set $A \subseteq \mathbb{N}$ is *factor-complete* if it is closed under taking multiplicative factors (excluding 1). With this, we give a set-theoretic description of precisely which sets can appear as $\mathrm{Tord}(G)$ for G finitely (or recursively) presented.

Theorem 4.32 ([30, Theorem 5.2])**.**
For a set of natural numbers A the following are equivalent:
(1) *$A = \mathrm{Tord}(G)$ for some finitely presented group G;*
(2) *$A = \mathrm{Tord}(G)$ for some countably generated recursively presented group G;*
(3) *A is a factor-complete Σ_2^0 set.*

Given any set $A \subseteq N$, the set $A_{\mathrm{prime}} := \{p_i \mid i \in A\}$ is factor-complete and many-one equivalent to A. Thus, taking A to be Σ_2^0-complete, the set A_{prime} is Σ_2^0-complete and can be realised as the set of orders of torsion elements of some finitely presented group G.

This page has been intentionally left blank.

5. FINITE QUOTIENTS

5.1. **Controlling finite quotients.**

Recall that every group G has an abelian universal quotient. This is known as the abelianisation G^{ab} (In this case, and at other times later on, we will use a suitable abbreviation for ρ when writing the ρ universal quotient G^{ρ} for a group G. For example, writing G^{ab} for G^{abelian}). We investigate another such property which admits universal quotients.

Definition 5.1.
Given a group G, we define the *finite residual* of G to be the normal subgroup

$$R_G := \bigcap_{i \in I} N_i$$

where $\{N_i\}_{i \in I}$ is the collection of all (not necessarily proper) finite index normal subgroups of G. A group G is said to be *residually finite* if $R_G = \{e\}$.

Lemma 5.2.
A group G is residually finite iff for each $e \neq g \in G$ there exists a finite group H_g and a homomorphism $\phi_g : G \to H_g$ such that $\phi_g(g) \neq e$ in H_g. That is, every non-trivial element of G survives in some finite quotient of G.

So G being residually finite (i.e., $R_G = \{e\}$) means G is 'as far as possible' from not having any finite quotients. Moreover, for an element $g \in G$, we have that $g \in R_G$ iff g has trivial image in every finite quotient of G, and $g \notin R_G$ iff G has some finite quotient in which the image of g is non-trivial. Thus we have proved the following:

Lemma 5.3.
G has no *finite quotients iff $R_G = G$.*

Residually finite groups have been studied extensively in the literature. They exhibit many useful properties. In particular, every f.p. residually finite group has SWP, and this is uniform over all finite presentations of residually finite groups.

Proposition 5.4.
Every group G has a residually finite universal quotient. That is, a quotient G^{res} for which every map from G to a residually finite group factors through G^{res}.

Proof. It is immediate that the *residual quotient*, $G^{\text{res}} := G/R_G$, has all the desired properties of a residually finite universal quotient for G. □

Unfortunately, the residual quotient of a finitely presented group can be very poorly behaved indeed, which we shall expand on in later sections.

Theorem 5.5 (Bridson-Wilton [31])**.**
There exists a finitely presentable group G for which G^{res} is not recursively presentable.

Proof. The work of Slobodskoĭ [6] has been be used by Bridson and Wilton [31] to construct a finite presentation $P = \langle X | R \rangle$ for which the finite residual $R_{\overline{P}}$ is not recursively enumerable (as a subset of $F(X)$). Hence $\overline{P}^{\text{res}}$ is not recursively presentable. □

Understanding finite quotients of a finitely presented group is quite complicated. In particular, there is a finitely presented group with *no* (non-trivial) finite quotients. We describe such a group here. See [8, §6] for more detail on this section.

If a subgroup H of a group G has finite index in G, namely a finite number of left (equivalently right) cosets, we will write here $H \leq_f G$ (and $H \trianglelefteq_f G$ if H is also normal in G, $H <_f G$ if H is also proper etc), whereas $[G : H]$ will denote the the number of these cosets: the *index* of H in G.

Theorem 5.6 ([8, 6.5])**.**
If $H \leq_f G$ with $[G : H] = n$ then there exists $N \trianglelefteq_f G$ with $N \leq H$ and $[G : N] \mid n!$.

So a group G has a non-trivial finite index subgroup iff it has a non-trivial finite index *normal* subgroup, and thus iff it has a non-trivial finite quotient. We will make extensive use of this fact.

Theorem 5.7 (Higman 1951)**.**
The group $\overline{Q}$ defined by the presentation

$$Q = \langle a_1, a_2, a_3, a_4 \mid a_1 a_2 a_1^{-1} = a_2^2,\ a_2 a_3 a_2^{-1} = a_3^2,\ a_3 a_4 a_3^{-1} = a_4^2,\ a_4 a_1 a_4^{-1} = a_1^2 \rangle$$

has no proper finite index subgroups.

Proof. If $H <_f \overline{Q}$ then (5.6) gives a non-trivial finite quotient $\overline{Q}/N$. Note that for $n > 1$ and a prime p dividing $2^n - 1$, the least prime factor of n is less than p.

Take r the order of $2 \bmod p$, then $r \mid n$ and $r \mid p - 1$ (by Fermat's Little Theorem). Now say n_i is the order of the image of $\overline{a_i}$ in $\overline{Q}/N$. Then $\overline{a_1}^n \overline{a_2}\, \overline{a_1}^{-n} = \overline{a_2}^{2^n}$, so $n_2 \mid 2^{n_1} - 1$, and so on.

Let p be the smallest prime dividing $n_1 n_2 n_3 n_4$, with (wlog) $p \mid n_2$. Then n_1 has a smaller prime factor. This is a contradiction unless $n_1 n_2 n_3 n_4 = 1$. □

Theorem 5.8.
The group $\overline{Q}$ from (5.7) *is infinite.*

Proof. The group H defined by the presentation $\langle x, y \mid yxy^{-1} = x^2 \rangle$ is an HNN extension in which we see that x and y have infinite order, by (2.34). Let H' be an isomorphic copy of H with presentation $\langle x', y' \mid y'x'y'^{-1} = x'^2 \rangle$. We form $H *_\varphi H'$, where $\varphi(\overline{x}) = \overline{y'}$ is infinite and will be defined by the presentation (via $x = y'$, $z = x'$)

$$\langle x, y, z \mid yxy^{-1} = x^2, xzx^{-1} = z^2 \rangle$$

Note that $\overline{y}, \overline{z}$ freely generate F_2 by (2.37) as a freely reduced word $w(y, x')$ with powers gathered becomes an A-reduced sequence with respect to this free product with amalgamation $H *_\varphi H'$.

Now take four copies of H, say H_i, each defined by the presentation $\langle a_i, b_i \mid b_i a_i b_i^{-1} = a_i^2 \rangle$ for $i = 1, 2, 3, 4$. Form

$$K = H_1 *_\varphi H_2 = \langle a_1 (= b_2), b_1, a_2 \mid b_1 a_1 b_1^{-1} = a_1^2,\ a_1 a_2 a_1^{-1} = a_2^2 \rangle$$

which is infinite with $\langle \overline{b_1}, \overline{a_2} \rangle$ free as before, and $L = H_3 *_\varphi H_4$ (send $1 \mapsto 3$, $2 \mapsto 4$).

Finally, make $K *_\theta L$ for $\theta(\overline{b_1}) = \overline{a_4}$, $\theta(\overline{a_2}) = \overline{b_3}$. This is $\overline{Q}$. □

We can use this group to show that having non-trivial finite quotients 'avoids' Markov properties, in the following sense:

Lemma 5.9.
Any finitely presented group $\overline{P}$ can be uniformly embedded into a 2-generator finitely presented group with no finite index subgroups.

Proof. Let $P = \langle X|R\rangle$ be any finitely presented group. We show that $\overline{P}$ embeds into some finitely presented group with no proper, finite index subgroups. Take the free product of $\overline{P}$ with the Higman group $\overline{Q}$, which has presentation

$$P * Q = \left\langle X, a, b, c, d \mid R, aba^{-1} = b^2, bcb^{-1} = c^2, cdc^{-1} = d^2, dad^{-1} = a^2 \right\rangle$$

Now form the 2-generator finite Adian-Rabin presentation $(P * Q)(a)$ over the word a via (4.1). This group has no proper, finite index subgroups. For suppose so, then it would have a proper, normal, finite index subgroup K by (5.6). As $\overline{(P * Q)(a)}/K$ is finite, then by (5.7) the image of a in this quotient is trivial. But by the Adian-Rabin relations (4.1), this means that the entire quotient is trivial. Hence $K = \overline{(P * Q)(a)}$. Finally, since $\overline{a} \neq e$ in $\overline{P}$, we observe that $\overline{P} \hookrightarrow \overline{P * Q} \hookrightarrow \overline{(P * Q)(a)}$, where $\overline{(P * Q)(a)}$ is a group without any proper subgroups of finite index. □

Theorem 5.10.
The group property $\rho :=$ 'having a non-trivial finite-index subgroup' is neither a Markov property nor a co-Markov property.

Proof.
ρ is not a Markov property, as there is no G_-; each finitely presented group G embeds into the group $G \times C_2$ which has ρ.
Conversely, the complement of ρ (having no non-trivial finite-index subgroup) is not a Markov property, as there is no G_-; each finitely presented group G embeds into a finitely presented group with no non-trivial finite index subgroups by (5.9). □

Thus we cannot appeal to (4.7) to show that 'having a non-trivial finite index subgroup' (equivalently, having a non-trivial finite quotient) is undecidable. We do have a starting point though: this property is r.e.

Lemma 5.11.
The set of finite presentations of groups which posess a non-trivial finite quotient is r.e.

Proof. Enumerate all finite groups by (2.68), and enumerate all maps to these groups by (2.67) and (2.63). Look for one which is non-trivial by (2.69). □

Lemma 5.12.
Let G be a finitely presented group. Then R_G is r.e.

Proof. This is similar to the above: enumerate all finite groups, and enumerate all maps to these groups. Look for one which the image of g is non-trivial. □

5.2. **Slobodskoĭ's theorem: the first order theory of finite groups.**

We won't formally define a first-order sentence in group theory here. Intuitively, it is a statement using the standard group-theoretic syntax (group operation, inverse, identity), along with quantifiers $(\forall, \exists)$, brackets $(,)$, the logical connectives conjunction $(\wedge)$, disjunction $(\vee)$, negation $(\neg)$, implication $(\Rightarrow)$, biconditional $(\Leftrightarrow)$. It can also use variables (x_0, x_1), and equality $(=)$. Note that we *cannot* quantify over *sets*, only over variables in the group itself.

Definition 5.13.
The *universal theory* of a class of groups C is the set of universally-quantified first-order sentences which are true on all all groups in C.
The *existential theory* of a class of groups C is the set of existentially-quantified first-order sentences which are true on all all groups in C.
The *first-order theory* of a class of groups C is the set of first-order sentences which are true on all groups in C.

Clearly, the universal and existential theories of a class C lie in the first order theory of that class. The following is a result of Sela [32]:

Theorem 5.14.
The first-order theory of the free group F_n is the same for every finite $n \geq 2$.

This was extended by Kharlampovich-Miasnikov [33] to the following:

Theorem 5.15.
The first-order theory of the free group F_n is the same for every finite $n \geq 2$, and moreover is recursive.

We begin by showing Slobodskoĭ's theorem for semigroups: the universal (and hence first-order) theory of finite semigroups is *not* recursive. More specifically, it is Π_1^0 (i.e., its complement is r.e., but it is not Σ_1^0).

Definition 5.16.
Take any fixed Minsky machine M. We define the following sentence $\Phi(k)$ in the language of semigroups, depending only on $k \in \mathbb{N}$:

$$\Phi(k) := \forall \theta, q_0, \ldots, q_n, a_1, a_2, A_1, A_2, A_1', A_2', h_1, h_2, e_1, e_2, c$$
$$(\Phi^* \to h_1 A_1 A_2 a_1^{2^k} c q_1 = \theta \vee h_2 A_1 A_2 a_1^{2^k} c q_1 = \theta)$$

where Φ^* is the conjunction of the formulas 1–4 below:

1. (Corresponding to the instructions of the Minsky machine M)
$a_1 a_2 c q_i = a_1^{1+\alpha} a_2^{1+\beta} c^2 q_j$ (where $q_i 00 \to q_j T_\alpha T_\beta$),
$A_1 a_2 c q_i = A_1^2 a_2^{1+\beta} a_1^\alpha c^2 q_j$ (where $q_i 10 \to q_j T_\alpha T_\beta$),
$A_2 a_1 c q_i = A_2^2 a_1^{1+\alpha} a_2^\beta c^2 q_j$ (where $q_i 01 \to q_j T_\alpha T_\beta$),
$A_1 A_2 c q_i = A_1^2 A_2^2 a_1^\alpha a_2^\beta c^2 q_j$ (where $q_i 11 \to q_j T_\alpha T_\beta$).

2.
$a_i c = c a_i,\ A_i c = c A_i,\ h_i c = c h_i,\ a_i A_i = A_i,$
$e_i A_i = A_i e_i = A_i,\ A_i A_i' = A_i' A_i = e_i,$
$A_i \theta = a_i \theta = c\theta = \theta,$
$e_i A_i' = A_i'$ where $i = 1, 2$.

3.
$a_i a_j = a_j a_i,\ A_i A_j = A_j A_i, h_i h_j = h_j h_i$, where $i, j = 1, 2$.
$a_i A_j = A_j a_i,\ h_i a_j = a_j h_i,\ A_i h_j = h_j A_i$, where $i, j = 1, 2$, $i \neq j$.

4.
$h_i e_i q_j = \theta$ where $i = 1, 2$, $j = 1, \ldots, n$.

We also set $x^0 y = y x^0 = y$ and $y x^0 z = z$, where x^0 is the zeroth power of x, for arbitrary values of x, y, z.

Again, for any fixed partial recursive function $f : \mathbb{N} \to \mathbb{N}$ and corresponding Minsky machine M_f as per (1.50), we let X be the domain of definition of f and let Y be the set of k for which $f(k)$ is undefined *and* for which M_f does not cycle. By (1.53), X and Y are recursively inseparable whenever f has nonrecursive domain of definition. We fix this notation for the remainder of this section.

Definition 5.17 (Slobodskoĭ's semigroup presentation)**.**
Define the finite presentation P_{Slob} of a semigroup with generators the variables of $\Phi(k)$ and relators the formulae $1-4$, as per (5.16).

Theorem 5.18.
If $k \in Y$, then the formula $\Phi(k)$ is true in all finite semigroups.
That is, for $k \in Y$, in any finite quotient H of $\overline{P}_{\text{Slob}}$ we have that either $h_1A_1A_2a_1^{2^k}cq_1 = \overline{\theta}$ or $h_2A_1A_2a_1^{2^k}cq_1 = \overline{\theta}$ or both, in H.

Proof. Suppose $k \in Y$. Let H be our finite semigroup quotient of $\overline{P}_{\text{Slob}}$, and let

$$\theta, q_0, \ldots, q_n, a_1, a_2, A_1, A_2, A_1', A_2', h_1, h_2, e_1, e_2, c$$

be elements of H satisfying the relations of Φ^*; that is, 1–4 of (5.16).

We show that, if the machine M starts in configuration $(2^k, 0; q_1)$ and hits the leftmost square of the first tape an infinite number of times, then in H the following holds:

$$h_1A_1A_2a_1^{2^k}cq_1 = \theta$$

Claim: $\exists N > 0$ such that $A_1^N = e_1$ in H.
Proof of claim: First notice that, as H is finite, we have that $A_1^m = A_1^n$ for some m, n with $m > n$. Thus, using the relations (2), we see that

$$\begin{aligned}
A_1^{m-n} &= A_1^{m-n}e_1^n \ \ (\text{as } A_1 = A_1e_1 \text{ and } m-n>0)\\
&= A_1^{m-n}(A_1A_1')^n \ \ (\text{as } A_1A_1' = e_1)\\
&= A_1^{m-n}A_1^n(A_1')^n \ \ (\text{as } A_1A_1' = A_1'A_1)\\
&= A_1^m(A_1')^n\\
&= A_1^n(A_1')^n \ \ (\text{as } A_1^m = A_1^n)\\
&= (A_1'A_1)^n \ \ (\text{as } A_1A_1' = A_1'A_1)\\
&= (A_1'A_1)^{n-1}(A_1A_1')\\
&= (A_1'A_1)^{n-1}e_1 \ \ (\text{as } A_1A_1' = e_1)\\
&= (A_1'A_1)^{n-1} \ \ (\text{as } A_1 = A_1e_1)\\
&= \vdots\\
&= (A_1'A_1)\\
&= e_1
\end{aligned}$$

Thus $A_1^N = e_1$ (where $N = m - n > 0$), thus proving our claim.

We now apply relations 1–3 to the word $h_1A_1A_2a_1^{2^k}cq_1$, keeping in mind that the relations 1 mimic the action of the Minsky machine M (observe that the power of A_i counts the number of times we have 'hit' the left end of tape i). By the fact that M reaches the leftmost square of the first tape infinitely often,

we apply relations 1–3 until the precise moment that M reaches the leftmost square of the first tape the $(N-1)^{\text{th}}$ time. Thus there exist τ, s, l, i such that

$$h_1 A_1 A_2 a_1^{2^k} c q_1 = c^\tau h_1 A_1^N A_2^s a_2^l q_i \text{ in } H$$

We have used here that fact that c commutes with a_j, A_j, h_j for $j = 1, 2$ (relators in 2), as well as the fact that when M reaches the leftmost square of the first tape, the power of a_1 must necessarily be 0.

It is now a straightforward application of relations 2–4, and the fact that $A_1^N = e_1$, to see that in H we have

$$\begin{aligned} c^\tau h_1 A_1^N A_2^s a_2^l q_i &= c^\tau A_2^s a_2^l h_1 A_1^N q_i \quad (\text{ relators 3 }) \\ &= c^\tau A_2^s a_2^l h_1 e_1 q_i \quad (A_1^N = e_1) \\ &= c^\tau A_2^s a_2^l \theta \quad (\text{ relators 4 }) \\ &= \theta \quad (\text{ relators 2 }) \end{aligned}$$

Performing an analogous analysis of the case when M hits the leftmost square of the second tape an infinite number of times, which we do not repeat here, we see that

$$h_2 A_1 A_2 a_1^{2^k} c q_1 = \theta \text{ in } H$$

Thus, when $k \in Y$, we have that $\Phi(k)$ is true in all finite semigroups. □

We now define a particular semigroup which is a finite quotient of $\overline{P}_{\text{Slob}}$, and in which $\Phi(k)$ is *not* true when $k \in X$.

Definition 5.19.
We define the semigroup presentation P_1 (for any fixed $T, t \geq 0$) with generators

$$\theta, q_0, \ldots, q_n, a_1, a_2, A_1, A_2, A_1', A_2', h_1, h_2, e_1, e_2, c$$

and we take the relating set of P_1 to be 1–4, along with the following:

5.
$q_j\theta = q_j a_i = q_j A_i = q_j A_i' = q_j h_i = q_j c = q_j$
where $i = 1, 2$, $j = 1, \ldots, n$.

6.
$\theta h_i = h_i \theta, A_i h_i = a_i h_i = A_i' h_i = h_i,\ h_i^2 = h_i,\ \theta^2 = \theta$, where $i = 1, 2$.

7.
$q_i q_j = q_i$ where $i, j = 1, \ldots, n$.

Observe that $\overline{P}_1$ is a finite quotient of $\overline{P}_{\text{Slob}}$.

First we show that we have some sense of 'normal form' for elements in $\overline{P}_1$:

Lemma 5.20.
Every element of $\overline{P}_1$ is equivalent to a word of the form

$$\theta^\gamma h_1^{\epsilon_1} h_2^{\epsilon_2} A_1^{s_1} A_2^{s_2} (A_1')^{t_1} (A_2')^{t_2} a_1^{m_1} a_2^{m_2} c^\tau q_i^\delta \quad (*)$$

where $\gamma, \delta, \epsilon_1, \epsilon_2 \in \{0, 1\}, s_i, t_i, m_i \in \mathbb{N}$.

Proof. First, notice that every generator of P_1 is already in the form $(*)$, except for e_1, e_2. However, we can write $e_i = A_i A_i'$ $(i = 1, 2)$, which is the form $(*)$. So now it suffices to show that the product of any two words in the form $(*)$ can be written in the form $(*)$.

So, take two words w_1, w_2 in the form $(*)$:

$$w_1 = \theta^{\gamma'} h_1^{\epsilon_1'} h_2^{\epsilon_2'} A_1^{s_1'} A_2^{s_2'} (A_1')^{t_1'} (A_2')^{t_2'} a_1^{m_1'} a_2^{m_2'} c^{\tau'} q_i^{\delta'}$$
$$w_2 = \theta^{\gamma} h_1^{\epsilon_1} h_2^{\epsilon_2} A_1^{s_1} A_2^{s_2} (A_1')^{t_1} (A_2')^{t_2} a_1^{m_1} a_2^{m_2} c^{\tau} q_i^{\delta}$$

Then, in P_1, their product $w_1 w_2$ is equivalent to

$$\theta^{\gamma'} h_1^{\epsilon_1'} h_2^{\epsilon_2'} A_1^{s_1'} A_2^{s_2'} (A_1')^{t_1'} (A_2')^{t_2'} a_1^{m_1'} a_2^{m_2'} c^{\tau'} q_{i'}^{\delta'} \theta^{\gamma} h_1^{\epsilon_1} h_2^{\epsilon_2} A_1^{s_1} A_2^{s_2} (A_1')^{t_1} (A_2')^{t_2} a_1^{m_1} a_2^{m_2} c^{\tau} q_i^{\delta}$$

First, consider the case where $\delta' = 1$. Then the relations in 5 give that $q_{i'}$ 'eats' all the letters to its right. Thus $w_1 w_2 = \theta^{\gamma'} h_1^{\epsilon_1'} h_2^{\epsilon_2'} A_1^{s_1'} A_2^{s_2'} (A_1')^{t_1'} (A_2')^{t_2'} a_1^{m_1'} a_2^{m_2'} c^{\tau'} q_i^{\delta'}$ as desired. So from hereon we need only consider the case where $\delta' = 0$ (that is, there is no appearance of $q_{i'}$ in w_1).

We will now deal with each successive letter in w_2, starting with θ^{γ} and moving all the way to q_i^{δ}.

$\boldsymbol{\theta^{\gamma}}$: If $\delta = 1$, then by the relation in 2, θ 'eats' all the A_j, A_j', a_j, c to its left. Also, θ commutes with all the h_j, by relations in 6. Finally, we have that $\theta^2 = \theta$. So, in the greatest generality, we can reduce $w_1 w_2$ to a word of the form

$$\theta^{\gamma''} h_1^{\epsilon_1'} h_2^{\epsilon_2'} A_1^{s_1''} A_2^{s_2''} (A_1')^{t_1''} (A_2')^{t_2''} a_1^{m_1''} a_2^{m_2''} c^{\tau''} h_1^{\epsilon_1} h_2^{\epsilon_2} A_1^{s_1} A_2^{s_2} (A_1')^{t_1} (A_2')^{t_2} a_1^{m_1} a_2^{m_2} c^{\tau} q_i^{\delta}$$

$\boldsymbol{h_1^{\epsilon_1}}$: By the relations in 2 and 3, h_1 commutes with a_2, A_2, A_2', c, h_2. Moreover, by the relations in 6, h_1 'eats' all the a_1, A_1, A_1' to its left. Finally, $h_1^2 = h_1$. So, in the greatest generality, we can further reduce $w_1 w_2$ to a word of the form

$$\theta^{\gamma''} h_1^{\epsilon_1''} h_2^{\epsilon_2'} A_1^{s_1'''} A_2^{s_2''} (A_1')^{t_1'''} (A_2')^{t_2''} a_1^{m_1'''} a_2^{m_2''} c^{\tau''} h_2^{\epsilon_2} A_1^{s_1} A_2^{s_2} (A_1')^{t_1} (A_2')^{t_2} a_1^{m_1} a_2^{m_2} c^{\tau} q_i^{\delta}$$

$\boldsymbol{h_2^{\epsilon_2}}$: A symmetric argument to that applied to $h_1^{\epsilon_1}$ gives that, in the greatest generality, we can further reduce $w_1 w_2$ to a word of the form

$$\theta^{\gamma''} h_1^{\epsilon_1''} h_2^{\epsilon_2''} A_1^{s_1'''} A_2^{s_2'''} (A_1')^{t_1'''} (A_2')^{t_2'''} a_1^{m_1'''} a_2^{m_2'''} c^{\tau''} A_1^{s_1} A_2^{s_2} (A_1')^{t_1} (A_2')^{t_2} a_1^{m_1} a_2^{m_2} c^{\tau} q_i^{\delta}$$

$\boldsymbol{A_1^{s_1}}$: By the relations in 2 and 3, A_1 commutes with a_2, A_2, A_2', c, and 'eats' any a_1 to its left. So, in the greatest generality, we can further reduce $w_1 w_2$ to a word of the form

$$\theta^{\gamma''} h_1^{\epsilon_1''} h_2^{\epsilon_2''} A_1^{s_1''''} A_2^{s_2'''} (A_1')^{t_1'''} (A_2')^{t_2'''} a_1^{m_1''''} a_2^{m_2'''} c^{\tau''} A_2^{s_2} (A_1')^{t_1} (A_2')^{t_2} a_1^{m_1} a_2^{m_2} c^{\tau} q_i^{\delta}$$

$\boldsymbol{A_2^{s_2}}$: A symmetric argument to that applied to $A_1^{s_1}$ gives that, in the greatest generality, we can further reduce $w_1 w_2$ to a word of the form

$$\theta^{\gamma''} h_1^{\epsilon_1''} h_2^{\epsilon_2''} A_1^{s_1''''} A_2^{s_2''''} (A_1')^{t_1'''} (A_2')^{t_2'''} a_1^{m_1''''} a_2^{m_2''''} c^{\tau''} (A_1')^{t_1} (A_2')^{t_2} a_1^{m_1} a_2^{m_2} c^{\tau} q_i^{\delta}$$

$\boldsymbol{(A_1')^{t_1}}$: By the relations in 2 and 3, A_1' commutes with a_2, A_2', c, and 'eats' any a_1 to its left. So, in the greatest generality, we can further reduce $w_1 w_2$ to a word of the form

$$\theta^{\gamma''} h_1^{\epsilon_1''} h_2^{\epsilon_2''} A_1^{s_1''''} A_2^{s_2''''} (A_1')^{t_1''''} (A_2')^{t_2'''} a_1^{m_1'''''} a_2^{m_2''''} c^{\tau''} (A_2')^{t_2} a_1^{m_1} a_2^{m_2} c^{\tau} q_i^{\delta}$$

$\boldsymbol{(A_2')^{t_2}}$: A symmetric argument to that applied to $(A_1')^{t_1}$ gives that, in the greatest generality, we can further reduce $w_1 w_2$ to a word of the form

$$\theta^{\gamma''} h_1^{\epsilon_1''} h_2^{\epsilon_2''} A_1^{s_1''''} A_2^{s_2''''} (A_1')^{t_1''''} (A_2')^{t_2''''} a_1^{m_1'''''} a_2^{m_2'''''} c^{\tau''} a_1^{m_1} a_2^{m_2} c^{\tau} q_i^{\delta}$$

$\boldsymbol{a_1^{m_1}, a_2^{m_2}, c^{\tau}}$: From the relators in 2 and 3, we see that a_1, a_2, c all commute. Thus, we can further reduce $w_1 w_2$ to a word of the form

$$\theta^{\gamma''} h_1^{\epsilon_1''} h_2^{\epsilon_2''} A_1^{s_1''''} A_2^{s_2''''} (A_1')^{t_1''''} (A_2')^{t_2''''} a_1^{m_1''''''} a_2^{m_2''''''} c^{\tau''} q_i^{\delta}$$

which is in the desired form $(*)$. □

Using this, we show that a very straightforward quotient of $\overline{P}_{\text{Slob}}$ is indeed our desired semigroup:

Lemma 5.21.
If $k \in X$, then there exists a finite semigroup on which the formula $\Phi(k)$ is false.
In particular, for $k \in X$, there is a finite presentation $P_2(k)$ (depending on k) of a quotient semigroup of $\overline{P}_1$ (and hence of $\overline{P}_{\text{Slob}}$) for which $\overline{P_2(k)}$ is finite and *we have that both $h_1 A_1 A_2 a_1^{2^k} cq_1 \neq \theta$ and $h_2 A_1 A_2 a_1^{2^k} cq_1 \neq \theta$ in $\overline{P_2(k)}$.*

Proof. Suppose $k \in X$. Now, choose t to be the number of working steps of the Minsky machine M before it reaches its halt state q_0. Suppose M reaches the leftmost square of the first tape $k_0 - 1$ times, and the leftmost square of the second tape $m_0 - 1$ times, during the steps $\tau = 0, 1, \ldots, t-1$. Then, by applying the relations from 1, we see that

$$h_i A_1 A_2 a_1^{2^k} cq_1 = c^t h_i A_1^{k_0} A_2^{m_0} a_1^{2^{g(k)}} cq_1$$

in the semigroup $\overline{P}_1$, for $i = 1, 2$.

If we restrict outselves to only relators from 1 (i.e., Minsky machine relators), then the word $h_i A_1 A_2 a_1^{2^k} cq_1$ can only be equal to words of the form

$$W = \{h_i A_1^l A_2^m a_1^s a_2^p c^\tau q_j \mid 0 \leq j \leq n,\ 1 \leq l, m < T,\ 0 \leq s, p \leq T,\ 0 < \tau \leq t\}$$

where we choose $T := 2t + 2^k + 1$.

We now define the finite presentation $P_2(k)$, depending on k, by taking the presentation P_1 from (5.20) and adding the following relations:

8.
$A_i^T = e_i,\ a_i^{T+1} = a_i^T,\ c^{t+1} = c^t,\ i = 1, 2.$

Clearly, $\overline{P_2(k)}$ is a quotient of $\overline{P}_{\text{Slob}}$, for all k. Moreover, it follows from the definition of $P_2(k)$ that $\overline{P_2(k)}$ is a finite semigroup, as there are only finitely many different normal forms as given in (5.20) (using the relations 8.). We now show that, in $\overline{P_2(k)}$,

$$\overline{h_i A_1 A_2 a_1^{2^k} cq_1} = \overline{c^t h_i A_1^{k_0} A_2^{m_0} a_1^{2^{g(k)}} cq_1} \neq \overline{\theta}$$

Assume $\overline{h_i A_1 A_2 a_1^{2^k} cq_1} = \overline{\theta}$. Then there is a sequence of words

$$h_i A_1 A_2 a_1^{2^k} cq_1 = w_0, w_1, \ldots, w_n = \theta$$

such that each successive pair w_j, w_{j+1} differ by precisely one application of a relator from 1-8; see (2.28). Take N to be a sequence of shortest possible length. We proceed by contradiction.

First, notice that the only way to introduce θ into a word with no occurence of θ is via the following relation in 4:

$$h_i e_i q_j = \theta,\ i = 1, 2,\ j = 1, \ldots, n$$

Let w_{J+1} be the first place where θ appears, so thus $w_J \to w_{J+1}$ is via the relation $h_i e_i q_j \to \theta$.

Claim 1: Take the word $w = h_i A_1 A_2 a_1^{2^k} cq_1$. Then applying a sequence of relators not of types 1 or 4 to w will always result in a word of the form $\alpha q_1 \beta$, where α contains no occurence of any q_i.

Proof of claim 1: The only relators which can change the q_1 are in 1 or 4. The other relators involving q_1 introduce things to the right of q_1. So the q_1 remains 'static'. Moreover, there is no way to introduce another q_i to the left of q_1. QED.

Observe that there is no relator of the type vq_iv', where both v, v' are non-empty. So we need only consider the 'first half' of w (up to the first occurence of q_i).

Claim 2: Take a word of the form $w = h_i A_1^{m_1} A_2^{m_2} a_1^{n_1} a_2^{n_2} c q_i$. Then we can apply precisely two type-1 relators to w; a 'forwards in time' relator and a 'backwards in time' relator (unless $i = 1$, in which case it might be only a forwards relator). Moreover, if we apply a sequence of relators of types 2-8 to w, then to the resulting word $w' = \alpha q_1 \beta$ we can either apply the SAME type 1 relators as to w to the subword αq_1, or NO type 1 relators to this subword.

Proof of claim 2: By the determinism of the Minsky machine, we can always apply a type 1 forwards relation to w. If $i \neq 1$, then we can also apply a type 1 backwards relation; if $i = 1$, then this might not be possible (indeed, it is possibe iff q_1 is a state that can be 'reached' after > 0 moves). Now, suppose we have applied a sequence of relators of type 2-8. So now we have a word $w' = \alpha q_1 \beta$. By claim 1, α contains no q_j. Moreover, by analysing the relators of types 2-8, we see that there is no way to bring a non-0 power of any of the A_1, A_2, a_1, a_2 to 0, and nor is there any way to permute A_1, a_1 or permute A_2, a_2. So either we can still apply the same type-1 relators to αq_1 as to w, or we have changed things sufficiently that we can apply no type-1 relator. QED.

Claim 3: Take a word of the form $w = h_i A_1^{s_1} A_2^{s_2} (A_1')^{t_1} (A_2')^{t_2} a_1^{m_1} a_2^{m_2} c q_j$. Then any sequence of relators not of types 1 or 4 will result in a word of the form $w = \alpha h_i A_1^{s_1'} A_2^{s_2'} (A_1')^{t_1'} (A_2')^{t_2'} a_1^{m_1'} a_2^{m_2'} c q_j \beta$ (maybe slightly 'jumbled' in the middle), where for $l = 1, 2$ we have that

$$s_l - t_l \equiv s_l' - t_l' \mod T$$

That is, the difference of the powers of A_l and A_l' cannot change (mod T) if we do not use a relator of type 1.

Proof of claim 3: Simply analyse all relators of type 2-8, to see what they do to the powers of A_l and A_l'. The only relevant relators are $e_l = A_l A_l'$, $A_l^T = e_l$, and $(A_l')^T = e_l$, and it can be seen that they preserve the powers mod T. Notice that there is no way to move A_l or A_l' past a h_i when the h_i is on the left. QED.

So the only way to get the power of A_l and A_l' to match, mod T, is to use relators of type 1. Note that each relator of type 1 increases the power of A_l by either 0 or 1.

Claim 4: Take the word $w = h_i A_1 A_2 a_1^{2^k} c q_1$. Then application of type 1 relators will have the power of each A_l ($l = 1, 2$) appearing between h_i and q_j in the range $1, \ldots, t$, where t is the number of steps the Minsky machine takes to halt. In particular, type 1 moves will not bring this power up to T, as $T = 2t + 2^k + 1 > t$.

Proof of claim 4: Suppose the Minsky machine progresses through configurations

$$C_1 \to C_2 \to \ldots \to C_t \downarrow$$

Then the type 1 relators we can apply to w will place it in 'configuration' sequence

$$C_1 \to C_{i_2} \to C_{i_3} \to \ldots$$

where $|i_{m+1} - i_m| = 1$ (so each step is either fowards or backwards). So the power of A_l at place C_j will always be $\leq j$ (and at least 1). In particular, we always have $1 \leq |A_l| \leq t$. QED.

In order to apply the relation 4 ($h_i e_i q_j = \theta$, which is the only way to get rid of the q_j, we must get the powers of A_i and A'_i to match, mod T, between h_i and q_j. But given that we start with $|A_i| = 1$ and $|A'_i| = 0$, we must increase the power of A_i to T, and this can only be done with type 1 moves. But claim 4 shows that we can only get this power up to t at most. So we can never apply relation 4, and thus we have

$$h_i A_1 A_2 a_1^{2^k} c q_1 \neq \theta \text{ in } \overline{P_2(k)}, \quad i = 1, 2$$

□

So using these, we get the following theorem:

Theorem 5.22 (Slobodskoĭ's theorem for semigroups)**.**
The universal (and hence first-order) theory of finite semigroups is not recursive.

Proof. Assume the universal theory of finite semigroups is recursive, and let $D = \{k \mid \Phi(k) \text{ is true in all finite semigroups }\}$. Then D is recursive. But then $Y \subseteq D$ by (5.18), and $X \cap D = \emptyset$ by (5.21). But X and Y are recursively inseparable, which contradicts the fact that D is recursive. □

We summarise the properties of Slobodskoĭ's semigroup here.

Theorem 5.23.
Let P_{Slob} be the finite presentation of Slobodskoĭ's semigroup from (5.17).
If $k \in Y$, then in every finite quotient H of $\overline{P}_{\text{Slob}}$ we have that either $\overline{h_1 A_1 A_2 a_1^{2^k} c q_1} = \overline{\theta}$ or $\overline{h_2 A_1 A_2 a_1^{2^k} c q_1} = \overline{\theta}$ or both, in H.
If $k \in X$, then in the finite quotient $\overline{P_2(k)}$ of $\overline{P}_{\text{Slob}}$ we have that both $h_1 A_1 A_2 a_1^{2^k} c q_1 \neq \theta$ and $h_2 A_1 A_2 a_1^{2^k} c q_1 \neq \theta$ in $\overline{P_2(k)}$.

So now we turn to the group theory case. The proof is much more complicated, and we will not provide full details here. See Slobodskoĭ's work in [6] for a full summary.

First, we require a small, but useful, group theoretic construction.

Definition 5.24. Let G be a group, and $g_1, g_2, g_3 \in G$. Then we define the following operation:

$$g_1 * g_2 := g_1 g_2 g_1^{-2} g_2 g_1$$

We use this to inductively define the following operations:

$$g_1^{(0)} * g_2 := g_2$$

$$g_1^{(1)} * g_2 := g_1 * g_2$$

$$g_1^{(m)} * g_2 := g_1 * (g_1^{(m-1)} * g_2) \quad m \geq 1$$

$$g_1^{(k)} * g_2^{(s)} * g_3 := g_1^{(k)} * (g_2^{(s)} * g_3) \quad k, s \geq 0$$

Note. With the usual group operation $\cdot$, it always true that $g \cdot h = h \Rightarrow h = e$, in any group G. However, it is not the case that $g * h = h \Rightarrow h = e$, or that $g \cdot h = h \Rightarrow h = e$. We will make use of this fact later, as we wish to avoid 'left cancellation'.

Here is the 'special sentence' in the group case, the analogue of the sentence $\Phi(k)$ in the semigroup case from (5.16).

Definition 5.25.
With the Minsky machine M computing function f as before, we define the following sentence $\Psi(k)$ in the language of groups, depending only on $k \in \mathbb{N}$:

$$\Psi(k) := \forall x_{ij}, \overline{x}_{ij} (i = 0, 1, 2; j = 0 \dots, n), x_\theta, c, a_i, A_i, A_i', h_i (i = 1, 2),$$

$$K_i, \overline{K}_i, M_i, \overline{M}_i (i = 1, 2), L_y, \overline{L}_y, S_y, \overline{S}_y \ (y \in F := \{c, a_i, A_i, A_i', h_i (i = 1, 2)\})$$

$$\left(\Phi^* \to h_1 * A_1 * A_2 * a_1^{(2^k)} * c * x_{0,1} = x_\theta \ \vee \ h_2 * A_1 * A_2 * a_2^{(2^k)} * c * x_{0,1} = x_\theta\right)$$

where Φ^* is the conjunction of the formulas 1'–8' below.

1′. (Corresponding to the instructions of the Minsky machine M)
$a_1 * a_2 * c * x_{0i} = a_1^{(1+\alpha)} * a_2^{(1+\beta)} * c^{(2)} * x_{0j}$ (where $q_i 00 \to q_j T_\alpha T_\beta$),
$A_1 * a_2 * c * x_{0i} = A_1^{(2)} * a_2^{(\alpha)} * a_1^{(1+\beta)} * c^{(2)} * x_{0j}$ (where $q_i 10 \to q_j T_\alpha T_\beta$),
$A_2 * a_1 * c * x_{0i} = A_2^{(2)} * a_1^{(1+\alpha)} * a_2^{(\beta)} * c^{(2)} * x_{0j}$ (where $q_i 01 \to q_j T_\alpha T_\beta$),
$A_1 * A_2 * c * x_{0i} = A_1^{(2)} * A_2^{(2)} * a_1^{(\alpha)} * a_2^{(\beta)} * c^{(2)} * x_{0j}$ (where $q_i 11 \to q_j T_\alpha T_\beta$).

2′.
$a_i c = c a_i, \ A_i c = c A_i, \ h_i c = c h_i, \ A_i A_i' = A_i' A_i, \ $ where $i = 1, 2$.

3′.
$a_i a_j = a_j a_i, \ A_i A_j = A_j A_i, h_i h_j = h_j h_i$, where $i, j = 1, 2$.
$a_i A_j = A_j a_i, \ h_i a_j = a_j h_i, \ A_i h_j = h_j A_i$, where $i, j = 1, 2$, $i \neq j$.

4′.
$A_i' * A_i * x_{0j} = A_i * A_i' * x_{0j} = x_{ij}, \ A_i * x_{ij} = A_i * x_{0j}$,
$h_i * x_{ij} = x_\theta, \ a_i * x_\theta = A_i * x_\theta = c * x_\theta = x_\theta$
$i = 1, 2, \ j = 1, \dots, n$.

5′.
The set $\{x_{i,j}, \overline{x}_{i,j}, x_\theta \mid i = 0, 1, 2, \ j = 0 \dots n\}$ generates an elementary abelian 2-group. That is, $pq = qp$ and $p^2 = e$, for every p, q in that set.

6′.
$K_i A_i = A_i K_i, \ a_i^3 K_i = K_i a_i^3, \ \overline{K}_i A_i = A_i \overline{K}_i, \ a_i^3 \overline{K}_i = \overline{K}_i a_i^3$,
$M_i A_i = A_i M_i, \ a_i^3 M_i = M_i a_i^3, \ \overline{M}_i A_i = A_i \overline{M}_i, \ a_i^3 \overline{M}_i = \overline{M}_i a_i^3$,
$M_i a_i^m x_{\tau j} a_i^{-m} M_i^{-1} = a_i^m M_i x_{\tau j} M_i^{-1} a_i^{-m} = a_i^m x_{\tau j} \overline{x}_{\tau j} a_i x_{\tau j} a_i^{-m-1}$,
$M_i a_i^m \overline{x}_{\tau j} a_i^{-m} M_i^{-1} = a_i^m M_i \overline{x}_{\tau j} M_i^{-1} a_i^{-m} = a_i^m \overline{x}_{\tau j} a_i x_{\tau j} a_i^{-m-1}$,
$\overline{M}_i a_i^m x_{\tau j} a_i^{-m} \overline{M}_i^{-1} = a_i^m \overline{M}_i x_{\tau j} \overline{M}_i^{-1} a_i^{-m} = a_i^m x_{\tau j} a_i \overline{x}_{\tau j} a_i^{-m-1}$,
$\overline{M}_i a_i^m \overline{x}_{\tau j} a_i^{-m} \overline{M}_i^{-1} = a_i^m \overline{M}_i \overline{x}_{\tau j} \overline{M}_i^{-1} a_i^{-m} = a_i^m x_{\tau j} \overline{x}_{\tau j} a_i \overline{x}_{\tau j} a_i^{-m-1}$,
$K_i a_i^m x_{\tau j} a_i^{-m} K_i^{-1} = a_i^m K_i x_{\tau j} K_i^{-1} a_i^{-m} = a_i^{m+1} x_{\tau j} a_i^{-m-1}$,
$K_i a_i^m \overline{x}_{\tau j} a_i^{-m} K_i^{-1} = a_i^m K_i \overline{x}_{\tau j} K_i^{-1} a_i^{-m} = a_i^{m+1} \overline{x}_{\tau j} a_i^{-m-1}$,
$\overline{K}_i a_i^m x_{\tau j} a_i^{-m} \overline{K}_i^{-1} = a_i^m \overline{K}_i x_{\tau j} \overline{K}_i^{-1} a_i^{-m} = a_i^{m+1} x_{\tau j} a_i^{-m-1}$,
$\overline{K}_i a_i^m \overline{x}_{\tau j} a_i^{-m} \overline{K}_i^{-1} = a_i^m \overline{K}_i \overline{x}_{\tau j} \overline{K}_i^{-1} a_i^{-m} = a_i^{m+1} \overline{x}_{\tau j} a_i^{-m-1}$,
where $i = 1, 2$, $j = 1, \dots, n$, $m = 0, 1, 2$, $\tau = 0, 1, 2$, $i \neq \tau$.

$7'$.
For every $y_1, y_2 \in F$ for which the relation $y_1y_2 = y_2y_1$ appears in $6'$, add the following relations:
$L_{y_1}y_2 = y_2L_{y_1},\ \overline{L}_{y_1}y_2 = y_2\overline{L}_{y_1},$
$S_{y_1}y_2 = y_2S_{y_1},\ \overline{S}_{y_1}y_2 = y_2\overline{S}_{y_1}.$

$8'$.
$L_y x_{ij} L_y^{-1} = yx_{ij}y^{-1},\ L_y\overline{x}_{ij}L_y^{-1} = \overline{x}_{ij},$
$\overline{L}_y x_{ij}\overline{L}_y^{-1} = x_{ij},\ \overline{L}_y\overline{x}_{ij}\overline{L}_y^{-1} = y\overline{x}_{ij}y^{-1},$
$S_y x_{ij} S_y^{-1} = x_{ij}\overline{x}_{ij}yx_{ij}y^{-1},\ S_y\overline{x}_{ij}S_y^{-1} = \overline{x}_{ij}yx_{ij}y^{-1},$
$\overline{S}_y x_{ij}\overline{S}_y^{-1} = x_{ij}y\overline{x}_{ij}y^{-1},\ \overline{S}_y\overline{x}_{ij}\overline{S}_y^{-1} = x_{ij}\overline{x}_{ij}yx_{ij}y^{-1},$
$i = 0, 1, 2,\ j = 1, \ldots, n,\ y \in F.$

Keep in mind that this sentence $\Psi(k)$ will play the same role in groups that the sentence $\Phi(k)$ played in semigroups.

Definition 5.26 (Slobodskoĭ's group presentation)**.**
Define the finite presentation Q_{Slob} of a group with generators the variables of $\Psi(k)$ and relators the formulae $1' - 8'$, as per (5.25).

Now the technical results begin. We state several of these without proof.

Lemma 5.27 ([6, Lemma 3])**.**
The following sugroups of $\overline{Q}_{\text{Slob}}$ are abelian, for fixed $j = 1, \ldots, n$ and fixed i in the range specified:

$$H_1^i = \langle (c^\tau A_1^k A_2^m a_1^s) x_{ij} (a_1^{-s} A_2^{-m} A_1^{-k} c^{-\tau}) | \tau, k, m, s \in \mathbb{Z} \rangle, i = 0, 2$$
$$H_2^i = \langle (c^\tau A_1^k A_2^m a_2^s) x_{ij} (a_2^{-s} A_2^{-m} A_1^{-k} c^{-\tau}) | \tau, k, m, s \in \mathbb{Z} \rangle, i = 0, 1$$
$$H_3^0 = \langle (c^\tau A_1^k a_1^m a_2^s) x_{0j} (a_2^{-s} a_1^{-m} A_1^{-k} c^{-\tau}) | \tau, k, m, s \in \mathbb{Z} \rangle$$
$$H_4^0 = \langle (c^\tau A_2^k a_1^m a_2^s) x_{0j} (a_2^{-s} a_1^{-m} A_2^{-k} c^{-\tau}) | \tau, k, m, s \in \mathbb{Z} \rangle$$
$$H_5^i = \langle (h_1^\tau c^m A_2^k a_2^s) x_{ij} (a_2^{-s} A_2^{-k} c^{-m} h_1^{-\tau}) | \tau, k, m, s \in \mathbb{Z} \rangle, i = 0, 1$$
$$H_6^i = \langle (h_2^\tau c^m A_1^k a_1^s) x_{ij} (a_1^{-s} A_1^{-k} c^{-m} h_2^{-\tau}) | \tau, k, m, s \in \mathbb{Z} \rangle, i = 0, 2$$

Proof. The proof is a huge formula-bashing exercise, with some inductive arguments, spanning 3 pages. The case for H_1^i can be found as the proof of [6, Lemma 3]. Note that it only makes use of relators 5'–8'. □

Lemma 5.28 ([6, Lemma 4])**.**
Let $y_1, \ldots, y_m \in F$ be such that $y_iy_j = y_jy_i$ in $\overline{Q}_{\text{Slob}}$ for $i, j = 1, \ldots, m$. Let $g \in \overline{Q}_k$ be such that the subgroup
$H = \langle (y_1^{k_1} \ldots y_m^{k_m}) g (y_m^{-k_m} \ldots y_1^{-k_1}) | k_1, \ldots, k_m \in \mathbb{Z} \rangle$ *is abelian. Then*

$$y_1 * \ldots * y_m * g = y_{i_1} * \ldots * y_{i_m} * g \text{ in } \overline{Q}_{\text{Slob}}$$

for any permutation $i_1, \ldots, i_m$ of $1, \ldots, m$.

Proof. We have

$$\begin{aligned} y_1 * \ldots * y_m * g =& y_1 \cdot \ldots \cdot y_m \cdot (g \cdot \Pi_{i=1}^m y_i^{-2} \cdot g \cdot y_i \cdot (\Pi_{i<j} y_i^{-2} y_j^{-2} g y_j^2 y_i^2 \cdot (\cdot \\ & (\Pi_{i<j<k} \ldots \cdot (y_1^{-2} \cdot \ldots \cdot y_m^{-2} \cdot g \cdot y_m^2 \cdot \ldots \cdot y_1^2) y_m^{-1} \cdot \ldots \cdot y_1^{-1} \\ =& y_{i_1} * \ldots * y_{i_m} * g \end{aligned}$$

□

Lemma 5.29 ([6, Lemma 5])**.**
Let H be a finite quotient of $\overline{Q}_{\text{Slob}}$. Then there exists $s > 0$ such that

$$A_i^{(s)} * x_{0j} = x_{ij} \quad \text{in } H, \text{ where } i = 1, 2,\ j = 1, \ldots, n$$

Proof. Fix i, j, and consider the sequence of elements $A_i^{(1)} * x_{0j}, A_i^{(2)} * x_{0j}, \ldots$. Since H is finite, then at least two elements in this infinite sequence must be equal. In particular, there must be some $k_{ij} > m_{ij}$ with

$$A_i^{(k_{ij})} * x_{0j} = A_i^{(m_{ij})} * x_{0j} \quad \text{in } H.$$

Repeating the proof of (5.27), we see that the subgroups

$$H_7^i = \langle (A_i')^k A_i^m x_{0j} A_i^{-m} (A_i')^{-k} | k, m \in \mathbb{Z} \rangle, i = 1, 2$$

are abelian. Hence, by (5.28) and the relations 4′, the following holds in H:

$$\begin{aligned}
(A_i')^{(m_{ij})} * (A_i)^{(k_{ij})} * x_{0j} &= \underbrace{A_i' * \ldots * A_i'}_{m_{ij}} * \underbrace{A_i * \ldots * A_i}_{k_{ij}} * x_{0j} \\
&= \underbrace{A_i' * \ldots * A_i'}_{m_{ij}-1} * \underbrace{A_i * \ldots * A_i}_{k_{ij}-1} * A_i' * A_i * x_{0j} \quad \text{(by (5.28), as } H_7^i \text{ is abelian)} \\
&= \underbrace{A_i' * \ldots * A_i'}_{m_{ij}-1} * \underbrace{A_i * \ldots * A_i}_{k_{ij}-1} * x_{ij} \quad \text{(using } A_i' * A_i * x_{0j} = x_{ij} \text{ in relator 4')} \\
&= \underbrace{A_i' * \ldots * A_i'}_{m_{ij}-1} * \underbrace{A_i * \ldots * A_i}_{k_{ij}-1} * x_{0j} \quad \text{(using } A_i * x_{ij} = x_{0j} \text{ in relator 4')} \\
&= \vdots \\
&= \underbrace{A_i * \ldots * A_i}_{k_{ij}-m_{ij}} * x_{0j} \\
&= A_i^{(k_{ij}-m_{ij})} * x_{0j}
\end{aligned}$$

But we also have that the following holds in H:

$$\begin{aligned}
(A_i')^{(m_{ij})} * (A_i)^{(k_{ij})} * x_{0j} &= (A_i')^{(m_{ij})} * (A_i)^{(m_{ij})} * x_{0j} \quad \text{(as } A_i^{(k_{ij})} * x_{0j} = A_i^{(m_{ij})} * x_{0j}) \\
&= \underbrace{A_i' * \ldots * A_i'}_{m_{ij}} * \underbrace{A_i * \ldots * A_i}_{m_{ij}} * x_{0j} \\
&= \underbrace{A_i' * \ldots * A_i'}_{m_{ij}-1} * \underbrace{A_i * \ldots * A_i}_{m_{ij}-1} * A_i' * A_i * x_{0j} \quad \text{(by (5.28), as } H_7^i \text{ is abelian)} \\
&= \underbrace{A_i' * \ldots * A_i'}_{m_{ij}-1} * \underbrace{A_i * \ldots * A_i}_{m_{ij}-1} * x_{ij} \quad \text{(using } A_i' * A_i * x_{0j} = x_{ij} \text{ in relator 4')} \\
&= \underbrace{A_i' * \ldots * A_i'}_{m_{ij}-1} * \underbrace{A_i * \ldots * A_i}_{m_{ij}-1} * x_{0j} \quad \text{(using } A_i * x_{ij} = x_{0j} \text{ in relator 4')} \\
&= \vdots \\
&= A_i' * A_i * x_{0j} \\
&= x_{ij}
\end{aligned}$$

Thus we see that

$$A_i^{(k_{ij}-m_{ij})} * x_{0j} = x_{ij}$$

Finally, observe that, for any $t > 0$, we have that the following holds in H:

$$\begin{aligned} A_i^{\left(t(k_{ij}-m_{ij})\right)} * x_{0j} &= A_i^{\left((t-1)(k_{ij}-m_{ij})\right)} * A_i^{(k_{ij}-m_{ij})} * x_{0j} \\ &= A_i^{\left((t-1)(k_{ij}-m_{ij})\right)} * x_{0j} \\ &= \vdots \\ &= A_i^{(k_{ij}-m_{ij})} * x_{0j} \\ &= x_{0j} \end{aligned}$$

Now set $s_{ij} := k_{ij} - m_{ij}$, and $s := \mathrm{lcm}(s_{ij})$. Then

$$A_i^{(s)} * x_{0j} = A_i^{\left((s/(k_{ij}-m_{ij}))(k_{ij}-m_{ij})\right)} * x_{0j} = x_{ij} \quad \text{in } H$$

and the lemma is proved. □

We now present the group-theoretic analogue of (5.22).

Lemma 5.30.
If $k \in Y$ then the formula $\Psi(k)$ is true on the class of all finite[10] *groups. That is, for $k \in Y$, if we take the finite presentation Q_{Slob}, then in any finite quotient H of $\overline{Q}_{\mathrm{Slob}}$ we have that either $h_1 * A_1 * A_2 * a_1^{(2^k)} * c * x_{0,1} = \overline{x_\theta}$ or $h_2 * A_1 * A_2 * a_2^{(2^k)} * c * x_{0,1} = \overline{x_\theta}$ or both, in H.*

Proof. Let $k \in Y$. Recall that this means that the Minsky machine M, when starting in the configuration $(2^k, 0; q_1)$, hits the leftmost cell of at least one of the tapes infinitely many times. Let H be a finite quotient of $\overline{Q}_{\mathrm{Slob}}$.
Suppose that starting in the configuration $(2^k, 0; q_1)$, M hits the leftmost cell of the first tape infinitely many times. By (5.29), we can take $s \in \mathbb{N}$ with $s > 0$ such that

$$A_1^{(s)} * x_{0j} = x_{ij} \quad \text{in } H \text{ for } j = 1, \ldots, n$$

Consider the following word in H:

$$w = h_1 * A_1 * A_2 * a_1^{(2^k)} * c * x_{01}.$$

Using the fact that H_1^0 is abelian (5.27), combined with (5.28), we get

$$w = h_1 * c * A_1 * A_2 * a_1^{(2^k)} * x_{01}.$$

Note that this represents the initial state of M in the configuration $(2^k, 0; q_1)$: M has hit the left side of both tapes 0 times.
We prove by induction that w is equal to word that represents each step of the machine M when started in the configuration $(2^k, 0; q_1)$. Suppose that the machine has been working for t_1 steps and has hit leftmost square of the first tape $s_1 - 1$ times and the leftmost square of the second tape $m_1 - 1$ times. The form of w depends on the configuration of M:

(C1) if M has configuration $(k_1, 0; q_i)$ where $k_1 \neq 0$ then

$$w = h_1 * c^{(t_1+1)} * A_2^{(m_1)} * A_1^{(s_1)} * a_1^{(k_1)} * x_{0i}.$$

[10] Slobodskoĭ proves this for all *periodic* groups.

(C2) if M has configuration $(k_1, e_1; q_i)$ where $k_1, e_1 \neq 0$ and there have been exactly t_2 steps since M has hit left leftmost cell of either tape then

$$w = h_1 * c^{(t_1 - t_2)} * A_2^{(m_1 - 2)} * A_1^{(s_1)} * c^{(t_2)} * A_2^{(2)} * a_1^{(k_1)} * a_2^{(e_1)} * x_{0i}.$$

(C3) if M has configuration $(0, e_1; q_i)$ where $e_1 \neq 0$ then

$$w = h_1 * c^{(t_1+1)} * A_2^{(m_1)} * a_2^{(e_1)} * A_1^{(s_1)} * x_{0i}.$$

(C4) if M has configuration $(0, 0; q_i)$ then

$$w = h_1 * c^{(t_1+1)} * A_2^{(m_1)} * A_1^{(s_1)} * x_{0i}.$$

We separate the proof into cases. The first case is when M is in configuration $(k_1, 0; q_i)$. By the induction hypothesis we have:

$$w = h_1 * c^{(t_1+1)} * A_2^{(m_1)} * A_1^{(s_1)} * a_1^{(k_1)} * x_{0i}.$$

The unique $(t_1+1)^{\text{th}}$ working step of M is $q_i 10 \to q_j T_\alpha T_\beta$ where $\beta \neq -1$. Using the fact that H_1^0 is abelian (5.27) and relations $1'$ we obtain:

$$\begin{aligned} w &= h_1 * c^{(t_1)} * A_2^{(m_1-1)} * A_1^{(s_1)} * a_1^{(k_1-1)} * A_2 * a_1 * c * x_{0i} \\ &= h_1 * c^{(t_1)} * A_2^{(m_1-1)} * A_1^{(s_1)} * a_1^{(k_1-1)} * A_2^{(2)} * a_1^{(1+\alpha)} * a_2^{(\beta)} * c^{(2)} * x_{0j} \end{aligned}$$

Using the fact that H_4^0 is abelian we can rearrange to get:

$$w = h_1 * c^{(t_1)} * A_2^{(m_1-1)} * A_1^{(s_1)} * c * A_2^{(2)} * a_1^{(k_1+\alpha)} * a_2^{(\beta)} * c * x_{0j}.$$

If $\beta = 1$ then we have achieved (C2). If $\beta = 0$ then the fact that H_1^0 is abelian allows us to get back to (C1) or (C4).
The other cases of the induction are similar, so we do not write out the details. Now, there exists a j, τ, m and e such that

$$w = h_1 * c^{(\tau)} * A_2^{(m)} * a_2^{(e)} * A_1^{(s)} * x_{0j}.$$

Therefore, since $A_1^{(s)} * x_{0j} = x_{1j}$, we have that

$$w = h_1 * c^{(\tau)} * A_2^{(m)} * a_2^{(e)} * x_{1j}.$$

Since H_5^1 is abelian (5.27), we can move h_1 to get

$$w = c^{(\tau)} * A_2^{(m)} * a_2^{(e)} * h_1 * x_{1j} = c^{(\tau)} * A_2^{(m)} * a_2^{(e)} * x_\theta = x_\theta.$$

The case where M hits the leftmost cell of the second tape is similar. □

Having dealt with the (easier) case of $k \in Y$, we now turn our attention to the (harder) case of $k \in X$. We will construct, for each $k \in X$, a finite quotient H of $\overline{Q}_{\text{Slob}}$ in which $h_1 * A_1 * A_2 * a_1^{(2^k)} * c * x_{0,1} \neq x_\theta$ and $h_2 * A_1 * A_2 * a_2^{(2^k)} * c * x_{0,1} \neq x_\theta$ in H. However, this is much more complicated than the semigroup case.

The main tool here will be to 'simulate' the semigroup $\overline{P}_1$ within a group, and then use the previous results for semigroups. We begin by giving a (general, and useful) way of representing *any* finitely presented semigroup within a group (that is, mimicking multiplication in the semigroup by multiplication in the group).

Definition 5.31.
Let S be a finite semigroup, whose elements are numbered $\{w_1, \ldots, w_m\}$. For each w_i we associate the 2^m symbols $x^j_{w_i}$ for $1 \leq j \leq 2^m$. Define $X := \{x^j_{w_i} \mid 1 \leq i \leq m,\ 1 \leq j \leq 2^m\}$, and let G_S be the elementary abelian 2-group with X as a generating set. That is,

$$G_S := \langle X \mid x^2 = e\ \forall x \in X,\ [y,z] = e\ \forall y, z \in X \rangle.$$

Now, for each w_i we associate an automorphism $\alpha_i \in \mathrm{Aut}(G_S)$ defined as follows:

$$\alpha_i^{-1}(x^j_{w_k}) := \begin{cases} x^j_{w_i w_k} x^{j+2^{i-1}}_{w_k} & \text{if } j = 2^i d + \tau \text{ and } 1 \leq \tau \leq 2^{i-1} \\ x^{j-2^{i-1}}_{w_k} & \text{if } j = 2^i d + \tau \text{ and } 2^{i-1} < \tau \leq 2^i \end{cases}$$

where $1 \leq k \leq m$, $1 \leq j \leq 2^m$, and $w_i w_k$ is the product of w_i and w_k in S.

Lemma 5.32.
If $w_i w_k = w_k w_i$ in S, then $\alpha_i \alpha_k = \alpha_k \alpha_i$ in $\mathrm{Aut}(G_S)$.

Proof. This can be found in [6, p. 150]. □

Definition 5.33.
Let K be a group. If $a \in K$, we define the *left translation map* $L_a : K \to K$, $L_a(x) := ax$ for any $x \in K$. Write $K^l := \{L_a \mid a \in K\}$.

We write Ω_K for the group of permutations of K.

Lemma 5.34.
Let K be a group. Then $K \cong K^l$, which is a subgroup of Ω_K.

Clearly, $\mathrm{Aut}(K)$ is a subgroup of Ω_K.

Definition 5.35.
The *holomorph* of a group K, $\mathrm{Hol}(K)$, is the subgroup of Ω_K generated by K^l and $\mathrm{Aut}(K)$. That is

$$\mathrm{Hol}(K) := \langle K^l, \mathrm{Aut}(K) \rangle \leq \Omega_K$$

Observe that $\mathrm{Hol}(K)$ is finite whenever K is.

$\mathrm{Hol}(K)$ is the semi-direct product $K \rtimes \mathrm{Aut}(K)$, and so viewing $\mathrm{Hol}(K)$ as a subgroup of Ω_K gives that $\alpha^{-1} x \alpha = \alpha(x)$ for any $\alpha \in \mathrm{Aut}(K)$ and any $x \in K$. For more details about holomorphs, see Rotman's book [12, p.164].

Definition 5.36.
Let $\overline{G}_S$ be an isomorphic copy of G_S, say $\overline{G}_S = \gamma(G_S)$ for some $\gamma \in \mathrm{Aut}(G_S)$. Define the group $G^*_S := G_S \times \overline{G}_S$.

Lemma 5.37.
Let S, G_S be as above. Then for any $1 \leq i, \tau \leq m$ and any $1 \leq j \leq 2^m$ we have, in the finite group $\mathrm{Hol}(G^*_S)$, that

$$\alpha_i * x^j_{w_\tau} = x^j_{w_i w_\tau} \quad \text{where } w_i w_\tau \text{ is some element of } S$$

Proof. This can be found in [6, p. 153]. □

Slobodskoĭ implicitly proves the following result:

Theorem 5.38 (Simulating a finite semigroup within a finite group)**.**
Let S be a finite semigroup, whose elements are numbered $\{w_1, \ldots, w_m\}$. Then we can algorithmically construct a finite group $\mathrm{Hol}(G_S^*)$ as given above, containing two families of elements: $\{\alpha_1, \ldots, \alpha_m\}$ (in bijection with the elements of S), and $\{x_{w_i}^j \mid 1 \leq i \leq m,\ 1 \leq j \leq 2^m\}$ (the j is an index, not a power). Let $w_{i_1}, \ldots, w_{i_k}$ be a sequence of elements of S, with $k > 1$. Then

$$\alpha_{i_1} * \alpha_{i_2} * \cdots * \alpha_{i_{k-1}} * x_{w_{i_k}}^1 = x_{w_{i_1} \cdots w_{i_k}}^1$$

That is, we can simulate multiplication in S through multiplication in $\mathrm{Hol}(G_S^*)$.

Proof. This is simply a corollary of (5.37), realling that we interpret $\alpha_i * \alpha_l * x_{w_\tau}^j := \alpha_i * (\alpha_l * x_{w_\tau}^j)$. □

For the rest of this section, take $k \in X$, take $S = \overline{P_2(k)}$ to be Slobodskoĭ's finite semigroup from (5.21), and take all notation from earlier.

Definition 5.39 (Extending α_i)**.**
We extend each automorphism $\alpha_i \in \mathrm{Aut}(G_{\overline{P_2(k)}})$ to an automorphism $\alpha_i \in \mathrm{Aut}(G^*_{\overline{P_2(k)}})$ (same symbol) by defining

$$\alpha_i(\overline{x}) := \gamma \circ \alpha_i \circ \gamma^{-1}(x)\ \forall x \in \overline{G}_{\overline{P_2(k)}}$$

We will now define automorphisms of $G^*_{\overline{P_2(k)}}$ corresponding to the variables $K_i, \overline{K}_i, M_i, \overline{M}_i (i = 1, 2), L_y, \overline{L}_y, S_y, \overline{S}_y$ (for all $y \in F := \{c, a_i, A_i, A_i', h_i (i = 1, 2)\}$). For convenience, number the set F as $F = \{\alpha_1, \ldots, \alpha_9\}$ (where $|F| = 9$).

Definition 5.40 (S_{α_i} and $\overline{S}_{\alpha_i}$)**.**
We define the automorphisms $S_{\alpha_i}, \overline{S}_{\alpha_i} \in \mathrm{Aut}(G^*_{\overline{P_2(k)}})$ for $1 \leq i \leq 9$ by setting

$$S_{\alpha_i}^{-1}(x) := x\gamma(x)\alpha_i^{-1}(x),\ S_{\alpha_i}^{-1}(\gamma(x)) := \gamma(x)\alpha_i^{-1}(x)$$
$$\overline{S}_{\alpha_i}^{-1}(x) := x\alpha_i^{-1} \circ \gamma(x),\ \overline{S}_{\alpha_i}^{-1}(\gamma(x)) := x\gamma(x)\alpha_i^{-1} \circ \gamma(x)$$

for all $x \in G$.

Definition 5.41 (L_{α_i} and $\overline{L}_{\alpha_i}$)**.**
We define the automorphisms $L_{\alpha_i}, \overline{L}_{\alpha_i} \in \mathrm{Aut}(G^*_{\overline{P_2(k)}})$ for $1 \leq i \leq 9$ to be the restrictions of α_i (which is now an automorphism of $G^*_{\overline{P_2(k)}}$) to $G_{\overline{P_2(k)}}$ and $\overline{G}_{\overline{P_2(k)}}$ respectively.

Definition 5.42.
Now consider the subset of $\overline{P_2(k)}$ of the following elements:

$$U := \{A_1^{m_1} a_1^{s_1} A_2^{m_2} a_2^{s_2} q_l \mid 1 \leq l \leq n,\ 1 \leq m_1, m_2, s_1, s_2 \leq T\}$$

and consider the sub semigroup $W_1 := \langle U \rangle \leq P_1$. We then define

$$G_w := \langle \{x_w^j \mid w \in W_1,\ 1 \leq j \leq 2^m\} \rangle \leq G_{\overline{P_2(k)}}$$
$$G_w' := \langle \{x_w^j \mid w \notin W_1,\ 1 \leq j \leq 2^m\} \rangle \leq G_{\overline{P_2(k)}}$$

Lemma 5.43.
Let $w \in W_1$ and $v \in P_1$ be words. Then $w = v$ as semigroup elements in $\overline{P_2(k)}$ iff v can be obtained from w using only relations $2 - 8$ of $P_2(k)$. In this case, $w = w_1 q_l$ in $\overline{P_2(k)}$ for some $w_1 \in \langle A_1, a_1, A_2, a_2 \rangle$.

Definition 5.44 (K_{α_i} and $\overline{K}_{\alpha_i}$)**.**
For x_w^j, where $w \in W_1$ and where $w = vq_l$ (c/f (5.43)) and where

$$\alpha_i^{-1}(x_{vq_l}^j) = x_{a_i vq_l}^j x_{vq_l}^\tau$$
$$\alpha_i^{-1}(x_{vq_l}^\tau) = x_{vq_l}^j$$

(for j, τ defined appropriately from the action of α_i), we then define the automorphisms $K_{\alpha_i} \in \mathrm{Aut}(G^*_{\overline{P_2(k)}})$ by setting:

$$K_i^{-1}(x_{vq_l}^j) = x_{va_i q_l}^j x_{vq_l}^\tau$$
$$\alpha_i^{-1}(x_{vq_l}^\tau) = x_{vq_l}^j$$

and by saying that K_i acts *trivially*[11] on G'_w and $\overline{G}_{\overline{P_2(k)}}$. Thus we have defined K_{α_i}.
Recall that $\gamma : G_{\overline{P_2(k)}} \to \overline{G}_{\overline{P_2(k)}}$ is a fixed automorphism of $G_{\overline{P_2(k)}}$. We define the automorphism $\overline{K}_i \in \mathrm{Aut}(G^*_{\overline{P_2(k)}})$ by setting:

$$\overline{K}_i(x) := x, \quad \overline{K}_i(\gamma(x)) := \gamma(K_i(x))$$

for every $x \in G$ (this defines $\overline{K}_i$ on both G and $\overline{G}_{\overline{P_2(k)}}$). Thus we have defined $\overline{K}_i$.

Definition 5.45 (M_{α_i} and $\overline{M}_{\alpha_i}$)**.**
For each $x \in G_{\overline{P_2(k)}}$ we define M_i and $\overline{M}_i$ by setting:

$$M_i^{-1}(x) = x\gamma(x)K_i^{-1}(x), \quad M_i^{-1}(\gamma(x)) = \gamma(x)K_i^{-1}(x),$$
$$\overline{M}_i^{-1}(x) = x\overline{K}_i^{-1}(\gamma(x)), \quad \overline{M}_i^{-1}(\gamma(x)) = x\gamma(x)K_i^{-1}(\gamma(x))$$

We now take a (finite) subgroup of the finite group $\mathrm{Hol}(G^*_{\overline{P_2(k)}})$ as our desired finite quotient group.

Theorem 5.46.
If $k \in X$, then there exists a finite group on which the formula $\Psi(k)$ is false. In particular, for $k \notin X$, there is a finite quotient H_k of $\overline{Q}_{\mathrm{Slob}}$ (depending on k), with $H_k \leq \mathrm{Hol}(G^*_{\overline{P_2(k)}})$, in which $h_1 * A_1 * A_2 * a_1^{(2^k)} * c * x_{0,1} \neq \overline{x_\theta}$ and $h_2 * A_1 * A_2 * a_2^{(2^k)} * c * x_{0,1} \neq \overline{x_\theta}$.
In particular, the subgroup $H_k \leq \mathrm{Hol}(G^*_{\overline{P_2(k)}})$ is generated by the following elements in $\mathrm{Hol}(G^*_{\overline{P_2(k)}})$ (and we give the corresponding elements that are mapped to them from $\overline{Q}_{\mathrm{Slob}}$):
For $i = 1, 2; j = 0 \ldots, n$,

$$x_{0j} := x_{q_j}^1$$
$$x_{ij} := x_{e_i q_j}^1$$
$$\overline{x}_{0j} := \overline{x}_{q_j}^1$$
$$\overline{x}_{ij} := \overline{x}_{e_i q_j}^1$$
$$x_\theta := x_\theta^1$$

[11] This is stated in the Russian text, but lost in translation in the English.

Then

$$\begin{aligned} a_1 &:= \alpha_1 \\ a_2 &:= \alpha_2 \\ A_1 &:= \alpha_3 \\ A_2 &:= \alpha_4 \\ c &:= \alpha_5 \\ A_1' &:= \alpha_6 \\ A_2' &:= \alpha_7 \\ h_1 &:= \alpha_8 \\ h_2 &:= \alpha_9 \end{aligned}$$

And finally take $K_i, \overline{K}_i, M_i, \overline{M}_i$ *(*$i = 1, 2$*) and* $L_{\alpha_j}, \overline{L}_{\alpha_j}, S_{\alpha_j}, \overline{S}_{\alpha_j}$ *(*$j = 1, \ldots, 9$*) as defined in* (5.40), (5.41), (5.44), (5.45).

Written out in the order they appear in $\Psi(k)$*, these are:*

$$x^1_{q_j}, \overline{x}^1_{q_j}, x^1_{e_1 q_j}, \overline{x}^1_{e_1 q_j}, x^1_{e_2 q_j}, \overline{x}^1_{e_2 q_j} \ (j = 0, \ldots, n), x^1_\theta$$
$$\alpha_1, \ldots, \alpha_9, K_i, \overline{K}_i, M_i, \overline{M}_i \ (i = 1, 2), L_{\alpha_j}, \overline{L}_{\alpha_j}, S_{\alpha_j}, \overline{S}_{\alpha_j} \ (j = 1, \ldots, 9)$$

Proof. We must first show that relations $1'$–$8'$ are satisfied. It is immediate from the definitions that $1'$–$5'$, $7'$, $8'$ are all satisfied. Thus it only remains to show that $6'$ is satisfied. This is done in [6, p. 153–155]
To conclude, we must show that, with the appropriate variable substitutions, in $\mathrm{Hol}(G^*_{\overline{P_2(k)}})$ we have the following:

1. $h_1 * A_1 * A_2 * a_1^{(2^k)} * c * x_{0,1} \neq x_\theta$, and
2. $h_2 * A_1 * A_2 * a_2^{(2^k)} * c * x_{0,1} \neq x_\theta$

Using (5.38), we see that

$$h_1 * A_1 * A_2 * a_1^{(2^k)} * c * x_{0,1} = x^1_{h_1 A_1 A_2 a_1^{(2^k)} c q_1} \text{ in } \mathrm{Hol}(G^*_{\overline{P_2(k)}})$$

a But as $k \in X$, we can now apply our earlier semigroup result (5.21) to see that

$$h_1 A_1 A_2 a_1^{(2^k)} c q_1 \neq \theta \text{ in } \overline{P_2(k)}$$

and thus that

$$x^1_{h_1 A_1 A_2 a_1^{(2^k)} c q_1} \neq x^1_\theta \text{ in } \mathrm{Hol}(G^*_{\overline{P_2(k)}})$$

Thus we have shown 1. An identical argument works for 2. □

We summarise the properties of Slobodskoĭ's semigroup here.

Theorem 5.47.
Let Q_{Slob} *be the finite presentation of Slobodskoĭ's group from* (5.26)*.*
If $k \in Y$*, then in every finite quotient* H *of* $\overline{P}_{\mathrm{Slob}}$ *we have that either* $\overline{h_1 * A_1 * A_2 * a_1^{(2^k)} * c * x_{0,1}} = \overline{x_\theta}$ *or* $\overline{h_2 * A_1 * A_2 * a_2^{(2^k)} * c * x_{0,1}} = \overline{x_\theta}$ *or both, in* H*.*
If $k \in X$*, then in the finite quotient* H_k *of* $\overline{q}_{\mathrm{Slob}}$ *we have that both* $\overline{h_1 * A_1 * A_2 * a_1^{(2^k)} * c * x_{0,1}} \neq \overline{x_\theta}$ *and* $\overline{h_2 * A_1 * A_2 * a_2^{(2^k)} * c * x_{0,1}} = \overline{x_\theta}$ *in* H_k*.*

5.3. **Bridson and Wilton theorem: undecidability of finite quotients.**
Observe that we can derive the following useful consequence of (5.47), phrasing things in terms of the finite residual:

Theorem 5.48.
Let Q_{Slob} be the finite presentation of Slobodskoĭ's group from (5.26)*. Define the words $w_i(k) := (h_i * A_1 * A_2 * a_1^{(2^k)} * c * x_{0,1})(x_\theta)^{-1}$ for $i = 1, 2$. Then we have*

$$k \in Y \;\Leftrightarrow\; \text{at least one of } \overline{w_1(k)},\ \overline{w_2(k)} \in R_{\overline{Q}_{\text{Slob}}}$$
$$k \in X \;\Leftrightarrow\; \text{both of } \overline{w_1(k)},\ \overline{w_2(k)} \notin R_{\overline{Q}_{\text{Slob}}}$$

We begin by proving the following (fairly straightforward) consequence of Slobodskoĭ's work [6]; a corollary to (5.48).

Theorem 5.49.
Take the finite presentation $Q_{\text{Slob}} = \langle X|R\rangle$ from (5.26)*. Then the set*

$$\{w \in F(X) \mid \overline{w} \notin R_{\overline{P}}\}$$

of words which lie outside *the finite residual $R_{\overline{Q}_{\text{Slob}}}$ is r.e. but not recursive. Thus there is no algorithm to determine whether or not a given word w has trivial image in every finite quotient $\overline{Q}_{\text{Slob}}$.*

Proof. Clearly the set of words lying outside $R_{\overline{P}}$ is r.e.; for any finite presentation P, given any word w from P we can enumerate all finite quotients of P by (2.67) and (2.68), and in each of these start checking if the image of w is non-trivial using (2.51). This will eventually halt iff $\overline{w} \notin R_{\overline{P}}$.

Conversely, take the finite presentation Q_{Slob} from (5.26). Suppose $R_{\overline{Q}_{\text{Slob}}}$ was recursive. Given $k \in \mathbb{N}$, form the words $w_i(k)$ for $i = 1, 2$, from (5.48). Form the set

$$D := \{k \in \mathbb{N} \mid \forall i = 1, 2,\ \overline{w_i(k)} \notin R_{\overline{Q}_{\text{Slob}}}\}$$

Then D is recursive, as $R_{\overline{Q}_{\text{Slob}}}$ is. But observe that, with X, Y defined as in (1.53), $X \subseteq D$ and $Y \cap D = \emptyset$, which contradicts the fact that X, Y are recursively inseparable. So $R_{\overline{Q}_{\text{Slob}}}$ is not recursive. □

We will make use of this result shortly, when we try and develop an Adian-Rabin construction for finite quotients. First, let us look more closely at the existing construction that we have.

Theorem 5.50.
Let $P = \langle X|R\rangle$ be a finite presentation of a group, and $w \in F(X)$. Then the finite presentation $P(w)$ as constructed in (4.1) *satisfies the following condition: w has trivial image in every finite quotient of $G \Rightarrow \overline{P(w)}$ has no non-trivial finite quotients.*

Proof. Recall from the proof of (4.1) that in any homomorphic image $\phi(\overline{P(w)})$ of $\overline{P(w)}$, if w has trivial image (that is, $\phi(\overline{w}) = e$), then ϕ has trivial image (that is, $\phi(\overline{P(w)}) = \{e\}$).
Now, let ψ be a surjective homomorphism $\overline{P(w)} \twoheadrightarrow H$ to some finite group H. Then by the hypothesis in our theorem, $\psi(\overline{w}) = e$ as H is finite. But then by what we have said above, $\psi(\overline{P(w)}) = \{e\}$, and so $H = \{e\}$. Thus $\overline{P(w)}$ has no non-trivial finite quotients. □

Suppose the implication in (5.50) was bi-directional: $\overline{w}$ has trivial image in every finite quotient of $G \Leftrightarrow \overline{P(w)}$ has no non-trivial finite quotients. Then, combining this with (5.49), we could immediately show that there is no algorithm to determine if a finitely presented group has a non-trivial finite quotient (in the same way that we used an f.p. group with IWP, combined with (4.1), to show that there is no algorithm to determine if a finitely presented group is trivial).

However, there is no reason to immediately assume that this reverse implication is true of the existing Adian-Rabin construction (4.1). The work of Bridson and Wilton [31] was dedicated to precisely this task: they designed a new Adian-Rabin construction in which the condition in (5.50) is bi-directional, thus proving the following (recalling that a group has a proper finite index subgroup iff it has a proper finite index normal subgroup iff it has a non-trivial finite quotient; (5.6)):

Theorem 5.51 (Bridson, Wilton [31, Theorem A])**.**
There is no algorithm to determine whether or not a finitely presented group has a proper subgroup of finite index. In particular, the set

$$\{P \mid \overline{P} \text{ has a proper subgroup of finite index } \}$$

is r.e. but not recursive.

In order to 'force' our new $P(w)$ to have a non-trivial finite quotient whenever w survives in some finite quotient of P, we must use some (very powerful) tools in group theory. Unfortunately, we only have time in this course to state these theorems; to prove them would require understanding much more theory, and take a lot more time.

We begin with a standard definition.

Definition 5.52.
A subgroup $H \leq G$ is said to be *malnormal* in G if $gHg^{-1} \cap H = \{e\}$ for every $g \in G \setminus H$.

That is, malnormal means 'as non-normal as possible'.

Lemma 5.53.
Take the free group $F_2 = \overline{\langle a, b | - \rangle}$, and fix any $n > 1$. Then the subgroup $\langle \overline{[a,b]}, \overline{[a^2, b^2]}, \ldots, \overline{[a^n, b^n]} \rangle$ is malnormal in F_2.

Here are two definitions which, though based on ideas we have not seen, can be 'reduced' to identifying malnormality.

Definition 5.54.
For T a tree, the action $\Gamma \curvearrowright T$ is *k-acylindrical* if, $\forall \gamma \in \Gamma \setminus \{e\}$, we have $\mathrm{Diam}(\mathrm{Fix}(\gamma)) \leq k$.

Definition 5.55.
A splitting $A *_C B$ is *acylindrical* if its graph of groups decomposition is k-acylindrical for some $k \geq 1$.

Here is the reduction that we need.

Lemma 5.56.
Consider the splitting $A *_C B$. Then:
1. *If C is malnormal in both A, B then the splitting is 1-acylindrical.*
2. *If C is malnormal in one of A, B then the splitting is 2-acylindrical.*

Definition 5.57.
Let $P = \langle X|R\rangle$ be a group presentation, with $|X|$ finite. A word $w \in F(X)$ is said to be a *geodesic* in P if it is of minimal length for the element it represents. That is, if $v \in F(X)$ with $\overline{w} = \overline{v}$ in $\overline{P}$, then $|v| \geq |w|$.

Definition 5.58.
Let $P = \langle X|R\rangle$ be a group presentation, with $|X|$ finite. A subgroup $H \leq \overline{P}$ is said to be *quasiconvex* in $\overline{P}$ if there exists some $k > 0$ such that if $w = x_1 \cdots x_m \in F(X)$ is a geodesic (with $x_i \in X \cup X^{-1}\ \forall i$), then for each $1 \leq j \leq m$ the element $\overline{x_1 \cdots x_j}$ is k-close to an element in H. That is, there is some word $v \in F(x)$ with $|v| \leq k$ and $\overline{x_1 \cdots x_j v} \in H$.

Definition 5.59.
Let $P = \langle X|R\rangle$ be a group presentation, with $|X|$ finite. We say that $\overline{P}$ is *word-hyperbolic* if there is some $\delta > 0$ such that for every geodesic triangle in P we have that each edge is contained in the δ-neighbourhood of the other two edges.

The following is often referred to as the *combination theorem*.

Theorem 5.60 (Bestvina, Feign [34])**.**
If $\Gamma = A *_C B$ with A, B hyperbolic and $C \hookrightarrow A$, $C \hookrightarrow B$ both quasiconvex, and the splitting is acylindrical, then Γ is hyperbolic.

Definition 5.61.
Let ρ be an algebraic property of groups. A group G is said to be *virtually-ρ* if G has a finite index subgroup $H \in \rho$.

The following is (a consequence of) the very deep results of Wise [35].

Theorem 5.62 (Wise [35])**.**
If $\Gamma = A *_C B$ is hyperbolic, C is quasiconvex in Γ, and A, B both virtually free, then Γ is residually finite.

To keep the value of the above results in context, note the following (somewhat counter-intuitive) result:

Theorem 5.63 (Burger, Mozes [36])**.**
There exists $m \in \mathbb{N}$ and a subgroup $A \leq F_m$ of finite index i (so by (2.15) *$A \cong F_k$ where $k = i(m-1)+1$) such that the amalgamated product $F_m *_{F_k} F_m$ is simple, and thus has* no *finite index subgroups.*

So now that we have the above observations, how might we make use of them? First, observe the following:

Lemma 5.64.
Let $P = \langle X|R\rangle$ be a finite presentation. Let S be a subset of $F(X)$, and take any $w \in F(X)$. Then the presentation $\langle X|R \cup S\rangle(w)$ constructed as per (4.1) *is the presentation $\langle X|R\rangle(w)$ with S adjoined to its relating set.*

Proof. The construction of $\langle X|R\rangle(w)$ is completely uniform in the relating set R. Thus we can add the extra relations S either before or after doing the construction from (4.1) and it does not change the final presentation. □

Theorem 5.65.
Let $P = \langle X|R\rangle$ be a finite presentation of a group. Take $w \in F(X)$. Suppose there is a surjection $\phi : \overline{P} \to H$ where $\phi(\overline{w}) \neq e$ in H. Take $Q = \langle X|R \cup S\rangle$ to

be a presentation for H, where the map ϕ is the extension of the identity on X. Then this induces a surjection $\psi : \overline{P(w)} \twoheadrightarrow \overline{Q(w)}$. That is, a surjection

$$\psi : (\overline{P} * F_2) *_\varphi F_2 \twoheadrightarrow (\overline{Q} * F_2) *_\varphi F_2$$

where φ is the map given in (4.1) *for the amalgamated product.*

There is a slight abuse of notation; we've used the map φ twice in the above statement, even though it denotes two 'different' maps. The meaning should be clear from the construction in (4.1).

Proof. Consider the construction in (4.1), starting with finite presentation $P = \langle X|R\rangle$ and word w. As $\overline{w} \neq e$ in $\overline{P}$, the group $\overline{P(w)}$ that we construct is the amalgamated product $(\overline{P} * F_2) *_\varphi F_2$, with $\varphi : A \to B$ as in (4.1).

As $\phi(\overline{w}) \neq e$ in $\overline{Q}$, the group $\overline{Q(w)}$ that we construct is the amalgamated product $(\overline{Q} * F_2) *_\varphi F_2$, with $\varphi' : A' \to B'$ as in (4.1) (we introduce $'$ on the subgroups and map to avoid confusion).

As $\phi(\overline{w}) \neq e$ in $\overline{Q}$, then from the quotient map $\phi : \overline{P} \twoheadrightarrow \overline{Q}$ we have that the induced quotient map $f : \overline{P} * F_2 \twoheadrightarrow \overline{Q} * F_2$ is injective on the subgroup A (which is isomorphically mapped to A'). This is immediate from the construction in (4.1).

Thus f extends to a quotient map $\psi : (\overline{P} * F_2) *_\varphi F_2 \twoheadrightarrow (\overline{Q} * F_2) *_{\varphi'} F_2$, given by extending the identity map on the generators of $(\overline{P} * F_2) *_\varphi F_2$ to the generators of $(\overline{Q} * F_2) *_{\varphi'} F_2$. □

Corollary 5.66.
Let $P = \langle X|R\rangle$ be a finite presentation of a group. Take $w \in F(X)$. Suppose there is a surjection $\phi : \overline{P} \to H$ where H is finite and $\phi(\overline{w}) \neq e$ in H. Then there is a surjection

$$\psi : \overline{P(w)} \twoheadrightarrow (H * F_2) *_\varphi F_2$$

where φ is the map given in (4.1) *for the amalgamated product.*

So now we know that if w survives in *some* finite quotient H of $\overline{P}$, then $\overline{P(w)}$ surjects onto the amalgamated product $(H * F_2) *_\varphi F_2$, where H is finite. Consider this group $(H * F_2) *_\varphi F_2$, and recall the notation of (4.1): $A \leq H * F_2$ and $B \leq F_2$. *Assume* that we can show that B is malnormal in the copy of F_2 that it sits in, and moreover that A, B are quasiconvex in $H * F_2, F_2$ respectively. Then we could use (5.56) to show that $(H * F_2) *_\varphi F_2$ is an acylindrical splitting. Moreover, as F_2 and $H * F_2$ are both hyperbolic[12], we could use (5.60) to give that $(H * F_2) *_\varphi F_2$ is hyperbolic.

As H is finite, it follows that $H * F_2$ is virtually free, by the following argument: Consider the kernel $\langle\langle F_2\rangle\rangle^{H*F_2}$ of the natural quotient map $f : H * F_2 \twoheadrightarrow H$. We have that $\langle\langle F_2\rangle\rangle^{H*F_2}$ is generated by the set $\{hgh^{-1}\}_{h\in H, g\in F_2}$. By the normal form theorem for free products (2.12), this set is a free basis of the subgroup it generates, and so $\langle\langle F_2\rangle\rangle^{H*F_2}$ is free in $H * F_2$ and has index $|H|$.

Now if we now *also assume* that A $(= B)$ is quasiconvex in $(H * F_2) *_\varphi F_2$, then we could use (5.62) to conclude that $(H * F_2) *_\varphi F_2$ is residually finite (and non-trivial). But then $\overline{P(w)}$ surjects onto the non-trivial residually finite group $(H * F_2) *_\varphi F_2$, which in turn surjects onto some non-trivial finite group. So $\overline{P(w)}$ has a non-trivial finite quotient whenever $\overline{w} \notin R_{\overline{P}}$.

[12]Finitely generated free groups, and finite groups, are hyperbolic. Moreover, a free product of a finite number of hyperbolic groups is hyperbolic.

In general, showing quasiconvexity is fairly straightforward. The problem is showing malnormality. We don't yet know how to show all the above assumptions for the construction (4.1). However, Bridson and Wilton [31] give a slightly different construction (with similar results) to that given in (4.1), and for their construction they verify all the assumptions listed above. We state here the construction they use, without going in to the details of why each step is necessary.

Construction 5.67.
Here is the Adian-Rabin type construction used by Bridson and Wilton in [31]. We have tried to be as consistent as possible with their notation and numberings.
INPUT:

- Finite presentation $G = \langle A \mid R\rangle = \langle a_1, \ldots, a_M \mid r_1, \ldots, r_n\rangle$.
- Word $w \in F(A)$.

CONSTRUCTION:
Step 1a.

- Take $2M+1$ copies of G; call these $G^{(1)}, \ldots, G^{(2M+1)}$.
- Let a_{ij} be the copy of a_i in $G^{(j)}$.
- Let w_j be the copy of w in $G^{(j)}$.
- Set $G^\dagger := G^{(1)} * \ldots * G^{(2M+1)}$ (free product presentation).
- Set $A^\dagger := \{a_{ij}w_{j+M+1}w_{i+j} \mid 1 \leq i \leq M, 1 \leq j \leq 2M+1\} \cup \{w_j \mid 1 \leq j \leq 2M+1\}$ (taking all j indicies modulo $2M+1$).
- Observe that $\overline{A^\dagger}$ generates $\overline{G^\dagger}$, as we can recover all the $\overline{a}_{ij}$.

Step 1b.

- Re-write $A^\dagger$ as $A^\dagger = \{a_1^\dagger, \ldots, a_m^\dagger\}$.
- Re-write the relators of $G^\dagger$ accordingly as $R^\dagger$; $G^\dagger = \langle A^\dagger \mid R^\dagger\rangle$.
- Define the finite presentation $G' := G^\dagger * \langle a_0' \mid -\rangle$.
- Set $a_i' := a_i^\dagger a_0'$ $\forall 1 \leq i \leq m$, $w^\dagger := w_1$.
- Set $w' := [w^\dagger, a_0']$.

Step 1c.

- Re-name G' as G.
- Re-name $A' := \{a_0', \ldots, a_m'\}$ as $A := \{a_0, \ldots, a_m\}$.
- Re-name w' as w.

Step 2.

- Define $G_1 := (G * \langle b_0, \ldots, b_m \mid -\rangle)/\langle\langle b_i w b_i^{-1} = a_i \ \forall 0 \leq i \leq m\rangle\rangle$ (this is an $m+1$ multi-HNN-extenion, gluing $\langle w\rangle$ to $\langle a_i\rangle$ for $0 \leq i \leq m$).
- Set $F_0 := \langle \overline{b}_0 \ldots, \overline{b}_m\rangle \leq \overline{G}_1$; this is free (of rank $m+1$) on the given generating set.

Step 3.

- Let $G_2 := G_1 * \langle t \mid -\rangle$.
- Let $F_1 := \langle F_0, \overline{t}\rangle \leq G_2$; this is free (of rank $m+2$) on the given generating set.

- Let $c_j := [w^{j+1}, t^{j+1}]$ for $0 \leq j \leq m+1$.
- Observe that $F_2 := \langle \bar{c}_0, \ldots, \bar{c}_{m+1} \rangle$ is a malnormal subgroup of $\langle \overline{w}, \bar{t} \rangle$ by (5.53).
- Set $F := \langle F_1, F_2 \rangle \leq \overline{G}_2$.

Step 4.

- Set $b_{m+1} := t$.
- Take a copy G_2' of G_2.
- Amalgamate $\overline{G}_2, \overline{G}_2'$ over $\varphi : F \cong F'$ via the extension of the map $c_i \mapsto b_i' \ \forall 0 \leq i \leq m+1$, $b_i \mapsto c_i' \ \forall 0 \leq i \leq m+1$.
- Thus define the finite presentation $G(w)$ by

$$G(w) := G_2 *_\varphi G_2' = (G_2 * G_2')/\langle\langle c_i = b_i', b_i = c_i' \ \forall 0 \leq i \leq m+1 \rangle\rangle$$

- We have that $\overline{w} \in R_{\overline{G}} \Leftrightarrow R_{\overline{G(w)}} = \overline{G(w)}$; the latter occuring iff $\overline{G(w)}$ posesses no non-trivial finite quotients.