# The Word Problem for Groups

Nate Brown

December 2022

## 1 Motivation

Apparently, computation shows up everywhere, even in group theory. One way to describe groups is to use a presentation, a set of generators and some rules for manipulating them. The word problem for groups described in this way is to determine whether two words, each some series of generators multiplied together, are equivalent. It turns out that this problem is unsolvable, due to the fact that some groups can encode a Turing machine.

This problem was worked on over the course of several years.

1911: Denn [4] proposed the word problem for groups.

1914: Thue [10] introduced effectively the word problem for semigroups.

1930s: Turing [11] established the existence of an unsolvable problem.

1947: Post [8] and Markov [5] independently showed that the word problem for semigroups is unsolvable.

1955: Boone [1] and Novikov [7] independently proved that the word problem for groups is unsolvable.

Post [8] directly encodes a Turing machine in a semigroup presentation and proves that the word problem for semigroups is unsolvable. Boone [1] proved that the word problem for groups was unsolvable as well with a new construction based off of Post's. Later, Britton [2] simplified this proof. The proof we will be covering is adapted from that and presented in Miller [6].

## 2 Background

**Definition** (Group)**.** A **group** is a set $S$ and an operation $\odot : S \times S \to S$ that satisfies:

- **Closure:** for $a, b \in S$, $a \odot b \in S$.
- **Identity:** there exists $i \in S$ such that for $a \in S$, $a \odot i = i \odot a = a$.
- **Inverses:** for $a \in S$, there exists $b \in S$ such that $a \odot b = b \odot a = i$.
- **Associativity:** for $a, b, c \in S$, $(a \odot b) \odot c = a \odot (b \odot c)$.

One example of a group is addition over the integers, another is the set of actions on a Rubik's cube where the operation is combining the two actions sequentially. Groups are often written with multiplication as the operation – $ab$ rather than $a \odot b$.

**Definition** (Semigroup)**.** A **semigroup** is like a group but it doesn't require inverses or an identity.

Some semigroups that aren't groups are: the positive integers, the reals with multiplication, the integers with multiplication, and strings of characters with concatenation.

**Definition** (Subgroup & subsemigroup)**.** A **subgroup** of a group is another group that shares the same operation and has a subset of the set. Similarly, a **subsemigroup** of a semigroup is another semigroup that shares the same operation and has a subset of the set.

Examples of subgroups include the even integers (which is a subgroup of the integers) and actions on the Rubik's cube that use only double turns (which is a subgroup of the Rubik's cube group). Similarly, examples of subsemigroups include the positive even integers (which is a subsemigroup of the positive integers) and the strings made from only `"q"`, `"A"`, and `"!"` (which is a subsemigroup of all strings).

**Definition** (Quotient Semigroup)**.** A **quotient semigroup** is a semigroup of partition $S'$ such that for all $a, b \in S$ where $a \in A \in S'$ and $b \in B \in S'$, $A \odot' B$ is the partition in $S'$ that contains $a \odot b$.

**Definition** (Group Generators)**.** A subset $S$ of a group $G$ **generates** $G$ if all elements of $G$ can be constructed with the elements of $S$ and their inverses. The definition for semigroups is similar.

For example, {`"a"`, `"A"`, `"!"`} generate the semigroup of strings mentioned above. $\{1\}$ generates the group of integers.

**Definition** (Group Presentation)**.** A **group presentation** is a pair of sets: the set of generators $S$ and a set of relations $R$. It is commonly written as $\langle S \mid R \rangle$. The resulting group is the quotient of the free group generated by $S$ and the equivalence relation generated by the transitive and symmetric closure of $R$ and the group axioms.

Informally, the corresponding group is the largest group that is generated by the generators $S$ but still satisfies all constraints imposed by $R$. For example, the Klein-4 group is $\langle a, b \mid ab = ba, aa = 1, bb = 1 \rangle$.

# 3 The Word Problem

**Definition** (The Word Problem)**.** The **word problem** for a group $G$ is the question of whether two words (of the generators) $A$ and $B$ represent the same element of the group.

This is easy for the presentation above, more complicated for some groups, and, in some cases, impossible.

# 4 Turing Machines and Semigroups

To see why, it is easiest to start with a similar problem for semigroups. Semigroups are like groups, but they do not need inverses or an identity.

Post [8] directly encodes a Turing machine in a semigroup presentation and proves that the word problem for semigroups is unsolvable. The following is a modification of Post's construction:

**Definition** (Post's $\gamma(T)$)**.** For the Turing machine $T = (Q, \Sigma, \Gamma, \delta, q_0, \sqcup, q_f, q_r)$, let $\gamma(T)$ be the semigroup with the presentation

$$\gamma(T) = \langle b, \xi, \Gamma, Q \mid R(T) \rangle$$

where the relations $R(T)$ are, for every $q \in Q \setminus q_f, q_r$ and $s, x \in \Gamma$,

$$\begin{aligned}
qsx &= s'q'x && \text{if } \delta(q,s) = (q', s', R) \\
qsb &= s'q' \sqcup b && \text{if } \delta(q,s) = (q', s', R) \\
xqs &= q'xs' && \text{if } \delta(q,s) = (q', s', L) \\
q_f x &= q_f && q_r x = q_r \\
xq_f b &= q_f b && xq_r b = q_r b \\
bq_f b &= \xi && bq_r b = \xi
\end{aligned}$$

$q$, $s$, and $x$ do what one would expect: $q$ represents $T$'s state, $s$ represents the tape symbol the head is over, and $x$ is a wildcard symbol. The generators $b$ and $\xi$ represent the boundary of the tape (creating blanks on the right if necessary) and halting respectively.

Consider the following Turing machine, that accepts a string composed of ( and ) if and only if the parentheses match.

| | $q_0$ | | | $q_1$ | | | $q_2$ | | |
|---|---|---|---|---|---|---|---|---|---|
| $($ | $q_1$ | $[$ | $R$ | $q_1$ | $($ | $R$ | $q_1$ | $/$ | $R$ |
| $)$ | $q_r$ | $-$ | $-$ | $q_2$ | $/$ | $L$ | $-$ | $-$ | $-$ |
| $/$ | $q_0$ | $/$ | $R$ | $q_1$ | $/$ | $R$ | $q_2$ | $/$ | $L$ |
| $\sqcup$ | $q_f$ | $-$ | $-$ | $q_r$ | $-$ | $-$ | $-$ | $-$ | $-$ |
| $[$ | $-$ | $-$ | $-$ | $-$ | $-$ | $-$ | $q_0$ | $/$ | $R$ |

$\gamma(T)$ mimics $T$ on the input () as follows:

$$\begin{aligned} bq_0()b &= b[q_1)b \\ &= bq_2[/b \\ &= b/q_0/b \\ &= b//q_0 \sqcup b \\ &= b// \sqcup q_f \sqcup b \\ &= b// \sqcup q_f b \\ &= b//q_f b \\ &= b/q_f b \\ &= bq_f b \\ &= \xi \end{aligned}$$

**Lemma** (Post)**.** *$bq_0wb \equiv \xi$ in $\gamma(T)$ if and only if $T$ halts on $w \in \Sigma^*$.*

*Proof.* It is pretty easy to show that any $T$ can be simulated in $\gamma(T)$. Each step has exactly one corresponding relation, including any that need to extend the tape. Thus, if $T$ transitions to state $q_f$ or $q_r$, then $bq_0wb = bxqyb$ for $q \in \{q_f, q_r\}$ and $x, y \in \Sigma^*$. By the last 6 relations, $bxqyb = bxqb = bqb = \xi$.

For the other way, we will first introduce the idea that the relations have a direction. All relations above are written in the forward direction (e.g. $q_0() \to [q_0$ is forward.) The opposite way will be considered as the backward direction.

Note that every word of the form $bxqyb$ where $q \in Q$ and $x, y \in \Sigma^*$ has a unique forward relation. To show this, we first consider that all relations require an interaction with some element of $Q$. Thus, we first filter the rules for the cases $q = q_f$ and $q = q_r$. If so, only the corresponding three rules can apply forward (Note that $q_f$ and $q_r$ have been explicitly excluded from the first three rule patterns.) We can then filter this to only 1 rule by checking the surroundings of $q$ for $b$s.

For the case where $q \in Q \setminus \{q_f, q_r\}$, the first three rule patterns could apply. We now filter this by taking $s$ and seeing what we get from $\delta(q, s)$. If $T$ goes left, we are left with exactly one rule, as the rule pattern, $q$, $s$, $q'$, $s'$, and even $x$[1] can all be pinned down. If $T$ goes right, then we first have to check for a $b$ to the right and everything can be pinned down as before.

Given that there is only one direction forward and that all relations preserve the form $bxqyb$ (or just $\xi$), if a relation is applied backward to a string, then the resulting string's unique forward relation must return to the original string. We know that each string of the from $bxqyb$ has a unique forward path, and we now know that going backward only extends that unique forward path. Using the fact that no backward relation produces $\xi$, a sting $bxqyb = \xi$ if and only if $\xi$ is reached in its unique forward path.

Thus, $bq_0wb = \xi$ if and only if $T$ halts on $w \in \Sigma^*$. □

With this lemma, the word problem for semigroups is unsolvable because the halting problem can be reduced to it.

Post's construction for semigroups does not work for groups. This is because the relations introduced restrict the group too much, and unwanted relations can be derived with the full group axioms. For example $q_f x = q_f$ can be left-multiplied by $q_f^{-1}$ to get $x = 1$. Since $x$ can be any symbol in $\Gamma$, this results in the entirety of any tape being 1.

[1] Unless $b$ is to the left. I have no clue what $T$, let alone $\gamma(T)$, does in that case.

# 5 Turing Machines and Groups

Boone [1] proved that the word problem for groups was unsolvable as well with a new construction based off of Post's. Later, Britton [2] simplified this proof. The proof we will be covering is adapted from that and presented in Miller [6].

The construction of Boone's group $\mathcal{B}(T)$ uses Post's $\gamma(T)$, with new symbols and conventions:

- $k$ is added to commute with halting pairs of Turing machines. This is used later to help confirm that the group doesn't collapse in unwanted ways.
- The $r_i$s are added to keep track of the used rules, making steps reversible.
- $x$ is added solely to keep the $r_i$s from returning to the Turing machine or commuting with each other.
- $t$ is put in between the Turing machines to allow the $r_i$s and $x$s between the TMs to let $k$ commute.

**Definition** (Boone's $\mathcal{B}(T)$)**.** Let $\mathcal{B}(T)$ be the group with the presentation

$$\mathcal{B}(T) = \langle b, \xi, \Gamma, Q, r_i, x, t, k \mid R_{\mathcal{B}}(T) \rangle$$

where $R_{\mathcal{B}}(T)$ has the following relations:

$$xs_i = s_i x^2 \tag{1}$$
$$r_i s_j = s_j x r_i x \tag{2}$$
$$r_i^{-1} F_i^{\#} q_{i_1} G_i r_i = H_i^{\#} q_{i_2} K_i \qquad \text{for each of } i \text{ rules in } \gamma(T) \tag{3}$$
$$r_i t = t r_i \tag{4}$$
$$xt = tx \tag{5}$$
$$r_i k = k r_i \tag{6}$$
$$xk = kx \tag{7}$$
$$(\xi^{-1} t \xi) k = k(\xi^{-1} t \xi) \tag{8}$$

The notation $F^{\#}$ for a word $F$ means the inversion of each generator in $F$ but the order remains the same (so not the same as $F^{-1}$). We'll define a word of $\mathcal{B}(T)$ as **special** if it's of the form $X^{\#} q_i Y$ for $X, Y \in (\Gamma \cup \{b\})^*$.

**Lemma** (Boone)**.** *For a special word $\Sigma = X^{\#} q_i Y$,*

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

*in $\mathcal{B}(T)$ iff $X q_i Y = \xi$ in $\gamma(T)$.*

*Proof.* Starting with sufficiency: assuming that, in $\gamma(T)$, $X q_i Y = \xi$, we can apply the derivation in $\gamma(T)$ to rewrite $\Sigma = L \xi R$ for $L, R \in \{x, r_i\}^*$. Then,

$$\begin{aligned} k(\Sigma^{-1} t \Sigma) &= k(R^{-1} \xi^{-1} L^{-1} t L \xi R) \\ &= k(R^{-1} \xi^{-1} t \xi R) \\ &= R^{-1} k(\xi^{-1} t \xi) R \\ &= R^{-1} (\xi^{-1} t \xi) k R \\ &= (R^{-1} \xi^{-1} t \xi R) k \\ &= (R^{-1} \xi^{-1} L^{-1} t L \xi R) k \\ &= (\Sigma^{-1} t \Sigma) k \end{aligned}$$

For necessity, first we need some machinery.

**Definition** (HNN Extension)**.** Let $G$ be a group with presentation $\langle S \mid R\rangle$, and $\alpha : H \to K$ be an isomorphism between two subgroups of $G$. Let $t$ be a new symbol not in $S$, and define

$$G*_\alpha = \left\langle S, t \mid R, tht^{-1} = \alpha(h), \forall h \in H \right\rangle.$$

The group $G*_\alpha$ is called the **HNN extension** of $G$ relative to $\alpha$. The new symbol $t$ is called the **stable letter**.

There are two important properties of these extensions that we'll need in this sketch:

**Lemma** (Higman–Neumann–Neumann)**.** *$G$ is naturally a subgroup of $G*_\alpha$ using the generators from $G$'s presentation.*

**Lemma** (Britton)**.** *If $w$ is a word of $G*_\alpha$ in which $t$ or $t^{-1}$ appears and if $w = 1$, then $w$ contains either a subword of the form $t^{-1}at$ or of the form $t\alpha(a)t^{-1}$ and so the relations of the form $t^{-1}at = \alpha(a)$ can be used to perform a $t$-pinch and reduce the number of $t$-symbols in $w$.*

Boone's construction allows a tower of groups:

$$\begin{aligned}
&\mathcal{B}_0 = \langle s \mid b\rangle \\
&\mathcal{B}_1 = \langle s, b, x \mid \text{rule (1)}\rangle \\
\mathcal{B}_1 \times Q& \\
&\mathcal{B}_2 = \langle s, b, x \mid \text{rules (1) through (3)}\rangle \\
&\mathcal{B}_3 = \langle s, b, x \mid \text{rules (1) through (5)}\rangle \\
&\mathcal{B}
\end{aligned}$$

Each is an HNN extension of the one above, with also $\mathcal{B}_2$ is also an HNN extension of $\mathcal{B}_1$.

Now we get to use it for $\mathcal{B}(T)$. Starting with $k(\Sigma^{-1}t\Sigma)k^{-1}(\Sigma^{-1}t^{-1}\Sigma) =_{\mathcal{B}} 1$, we can use $k$ as a stable letter to get

$$\begin{aligned}
k(\Sigma^{-1}t\Sigma)k^{-1}(\Sigma^{-1}t^{-1}\Sigma) &=_{\mathcal{B}} 1 && \\
W(\Sigma^{-1}t^{-1}\Sigma) &=_{\mathcal{B}_3} 1 && \text{pinch } k\text{, giving } W \in \{r_i, x, \xi^{-1}t\xi\}^* \\
W &=_{\mathcal{B}_3} \Sigma^{-1}t\Sigma && \\
W_1 &=_{\mathcal{B}_3} \Sigma^{-1}t\Sigma && \text{pinch } t \text{ out of } W \text{ leaving } W_1 \text{ with only one } t
\end{aligned}$$

So now $W_1$ looks like $R_0^{-1}\xi^{-1}t\xi R$ where $R, R_0 \in \{r_i, x\}^*$, and we can pinch again:

$$\begin{aligned}
W_1 &=_{\mathcal{B}_3} \Sigma^{-1}t\Sigma && \\
R_0^{-1}\xi^{-1}t\xi R &=_{\mathcal{B}_3} \Sigma^{-1}t\Sigma && \\
\Sigma^{-1}t^{-1}\Sigma R_0^{-1}\xi^{-1}t\xi R &=_{\mathcal{B}_3} 1 && \\
\Sigma^{-1}L\xi R &=_{\mathcal{B}_3} 1 && \text{pinch } t\text{, mapping } \Sigma R_0^{-1}\xi^{-1} \text{ to some } L \in \mathcal{B}_2 \\
\Sigma &=_{\mathcal{B}_2} L\xi R &&
\end{aligned}$$

This gives us $X^\#\xi Y R^{-1}\xi^{-1} =_{\mathcal{B}_2} L$, so $X^\#\xi Y =_{\mathcal{B}_2} L\xi R$, which looks suddenly a lot like we're back in Post's $\gamma(T)$ construction but with these extra $L$ and $R$. Fortunately, $L$ and $R$ give us a list of rules (in the $r_i$) that can be used to perform the derivation in $\gamma(T)$, giving the equivalence we need. □

Given a Turing machine $T_U$ with an unsolvable halting problem, the word problem for $\mathcal{B}(T_U)$ is unsolvable because the equivalence

$$k\left((b^\# q_0 W b)^{-1} t (b^\# q_0 W b)\right) =_{\mathcal{B}(T_U)} \left((b^\# q_0 W b)^{-1} t (b^\# q_0 W b)\right) k$$

is true if and only if

$$bq_0Wb =_{\gamma(T_U)} \xi$$

which is true if and only if $T_U$ halts on $W$.

Thus,

**Theorem** (Boone–Novikov)**.** *There exists a group with an unsolvable word problem.*

# References


[1] Boone, W. W. The word problem. *Annals of mathematics* (1959), 207–265.

[2] Britton, J. L. The word problem. *Annals of Mathematics* (1963), 16–32.

[3] Cravitz, W. An introduction to the word problem for groups. 2021.

[4] Dehn, M. Über unendliche diskontinuierliche gruppen. *Mathematische Annalen 71*, 1 (1911), 116–144.

[5] Markov, A. On the impossibility of certain algorithms in the theory of associative systems. *Doklady Akademii Nauk SSSR. New Series 55* (1947), 587–590.

[6] Miller, C. F. Turing machines to word problems. In *Turing's Legacy: Developments from Turing's ideas in logic*, R. Downey, Ed. Cambridge University Press, 2014, pp. 329–385.

[7] Novikov, P. S. On the algorithmic unsolvability of the word problem in group theory. *Trudy Matematicheskogo Instituta imeni VA Steklova 44* (1955), 3–143. translated as American Mathematical Society Translations. Second Series, vol. 9 (1958), pp. 1–122.

[8] Post, E. L. Recursive unsolvability of a problem of thue. *The Journal of Symbolic Logic 12*, 1 (1947), 1–11.

[9] Power, J. F. Thue's 1914 paper: a translation. *arXiv preprint arXiv:1308.5858* (2013).

[10] Thue, A. Probleme über veränderungen von zeichenreihen nach gegebenen regeln. *Skrifter utgit av Videnskapsselskapet i Kristiania, I. Mathematisk-naturvidenskabelig klasse 10* (1914).

[11] Turing, A. M., et al. On computable numbers, with an application to the entscheidungsproblem. *J. of Math 58* (1936), 345–363.