Aqil Azmi

Midterm

1. Proof

Theorem *Let H be any subgroup of the group G. The set of the left cosets of H form a partition of G.*

Proof. First, we define a group $(G, *)$ with its basic axioms. For convenience, we also prove the result of the right-cancellation law.

Lemma 1 (Right-cancellation law) Let $x, y, z \in G$. Suppose that $y \cdot x = z \cdot x$, then $y = z$.

Proof.

$$
\begin{gathered}
y \cdot x = z \cdot x \\
(y \cdot x) \cdot x^{-1} = (z \cdot x) \cdot x^{-1} \\
y \cdot (x \cdot x^{-1}) = z \cdot (x \cdot x^{-1}) \\
y \cdot e = z \cdot e \\
y = z \, \blacksquare
\end{gathered}
$$

Next, we define a subgroup $(H, \cdot)$ with the same properties. For convenience, we prove the following result about inverses involving multiplication:

Lemma 2 (Inverses of multiplications) Let $x, y \in G$. Then, $(x \cdot y)^{-1} = y^{-1} \cdot x^{-1}$.
Proof.

$$
\begin{gathered}
(x \cdot y) \cdot (x \cdot y)^{-1} = e \\
x^{-1} \cdot (x \cdot y) \cdot (x \cdot y)^{-1} = x^{-1} \cdot e \\
(x^{-1} \cdot x) \cdot y \cdot (x \cdot y)^{-1} = x^{-1} \\
y^{-1} \cdot (x^{-1} \cdot x) \cdot y \cdot (x \cdot y)^{-1} = y^{-1} \cdot x^{-1} \\
y^{-1} \cdot e \cdot y \cdot (x \cdot y)^{-1} = y^{-1} \cdot x^{-1} \\
y^{-1} \cdot y \cdot (x \cdot y)^{-1} = y^{-1} \cdot x^{-1} \\
e \cdot (x \cdot y)^{-1} = y^{-1} \cdot x^{-1} \\
(x \cdot y)^{-1} = y^{-1} \cdot x^{-1} \, \blacksquare
\end{gathered}
$$

Then, we introduce and define the notion of an equivalence relation in general.

Definition 3 (Equivalence relation) An equivalence relation on S is a binary relation $\sim$ such that for any $x, y, z \in S$, we have:

1. (Reflexivity) $x \sim x$
2. (Symmetry) $x \sim y \iff y \sim x$
3. (Transitivity) $x \sim y$ and $y \sim z$ implies $x \sim z$

Then, we specify it in the context of a subgroup.

Definition 4
Then, we need to prove that this equivalence relation holds for all three properties.

Proposition 5 The relation of a subgroup to the group holds for all three properties, thus it is an equivalence relation.

Proof. 1. (Reflexivity) We know that $e \in H$. Thus, we pick that as our element of H and we have $x = x \cdot e$. Therefore, $x \sim x$.

2. (Symmetry) Given that $x \sim y$, we know that $x = y \cdot h$ for some $h \in H$. If we multiply both sides with h^{-1} we will get $x \cdot h^{-1} = y \cdot h \cdot h^{-1}$ and thus, $y = x \cdot h^{-1}$. Since $h^{-1} \in H$, then we know that $y \sim x$.

3. (Transitivity) Given $x \sim y$ and $y \sim z$, we know that $x = y \cdot h$ for some $h \in H$ and $y = z \cdot k$ for some $k \in H$. We substitute y into the first equation and we get $x = z \cdot k \cdot h$. Since k and $h \in H$ and a subgroup is closed under multiplication, then we have $(k \cdot h) \in H$. Therefore, $x \sim z$. ∎

Next, we define the set that is known as the left coset:

Definition 6 (Left coset) Given some $g \in G$. We define the left coset of g in H (denoted gH) as:

$$gH = \{x \in G | x \sim g\} = \{x \in G | x = gh, h \in H\}$$

Next, we prove a necessary lemma that would help us later in the proof.

Lemma 7 (Identity of a subgroup) If e_G is the identity element of a group G, then the identity element of a subgroup H, e_H is the same element, i.e. $e_G = e_H$.

Proof. We know that H must be non-empty and that it must at least contain the identity element, e_H. Since the elements of a subgroup are a subset of the elements of a group, then $e_H \in G$. Next, we take an arbitrary element $x \in H$, and by the same reasoning, $x \in G$ is necessary. Using group axioms, we know that $\exists x^{-1} \in G$ and that $x \cdot x^{-1} = e_G$. The same would apply for the subgroup as well, thus we have $\exists x^{-1} \in H$ and $x \cdot x^{-1} = e_H$. Equating these two results together, we have $e_G = e_H$. ∎

Then, we show an important step which is that every element in G is in some coset of H

Lemma 8 (Every element of G is in some coset of H) For every $x \in G$, where G is a group and $H \leq G$ is a subgroup, $\exists g \in G$ such that x is in the left coset of H, i.e. $x \in gH$.

Proof. Given that we have $x \in G$. We know that H is non-empty and that it must contain the identity element, i.e. $e_H \in H$. Consider the equation $x = x \cdot e_G$. By Lemma 7, we know that this identity is the same as the identity in H. Thus, we can rewrite this as $x = x \cdot e_H$. Since we know that $e_H \in H$, thus $x \in gH$ for all $x \in G$. ∎

Then, we show another important step which is that any two distinct left cosets of H are disjoint.

Lemma 9 (Distinct cosets are disjoint) Given some $g_1, g_2 \in G$ where $g_1 \neq g_2$. The two distinct cosets of H formed by these two elements, g_1H and g_2H, where $g_1H \neq g_2H$, are disjoint i.e. $g_1H \cap g_2H = \emptyset$.

Proof. Assume, for contradiction, that $g_1H \cap g_2H \neq \varnothing$. Then, $\exists x, x \in g_1H \cap g_2H$. Using the definition of set intersection, this means that $x \in g_1H$ and $x \in g_2H$. By the definition of left cosets, this means that there is some $h_1 \in H$ such that $x = g_1 \cdot h_1$ and some $h_2 \in H$ such that $x = g_2 \cdot h_2$. Thus, we also have the equality that $g_1 \cdot h_1 = g_2 \cdot h_2$. If we multiply both sides by h_1^{-1}, we will obtain $g_1 = g_2 \cdot h_2 \cdot h_1^{-1}$. Since $h_2 \cdot h_1^{-1} \in H$ by the closure and inverse properties, we now have $g_1 \in g_2H$. Therefore, $g_1 = g_2 \cdot h$ for some $h \in H$. Take some arbitrary $x \in g_1H$. Thus, $x = g_1 \cdot h'$ for some $h' \in H$. Putting the two equations together, we get $x = g_2 \cdot h \cdot h'$. Since H is closed under multiplication, $h \cdot h' \in H$. Therefore, $x \in g_2H$. This means that $g_1H \subseteq g_2H$. Using the same reasoning again, we can then obtain $g_2H \subseteq g_1H$. These two subset equations together imply $g_1H = g_2H$, which is a contradiction. ∎

Finally, we can show that the union of the left cosets of H is equal to G.

Lemma 10 (Union of left cosets of subgroup is the entire group) Let H be a subgroup of a group G. The group G is the disjoint union of the left cosets of H in G. That is,

$$G = \bigcup_{g \in G} gH$$

Proof. Using the extensionality axiom of sets, we can say they are equal if and only if they both have the same elements. Thus, we take an arbitrary element x in one of the sets, and prove that it is also in the other set, in both directions.

Case 1: $x \in \bigcup_{g \in G} gH \to x \in G$

First, we take an arbitrary $y \in G$, and thus have the left coset yH. (In the code in Lean, I call this arbitrary left coset s). Take an arbitrary element x from this left coset. Thus, $x \in yH$. This means that $x = y \cdot h$ for some $h \in H$. But since the elements of a subgroup is a subset of the elements of a group, we thus have $h \in G$. Since a group is closed under multiplication, this means that $y \cdot h \in G$. Therefore, $x \in G$.

Case 2: $x \in G \to x \in \bigcup_{g \in G} gH$

Pick an arbitrary element x in G. This element will have a left coset of H, which is xH. Consider the equation $x = x \cdot e$. Since $e \in H$, we know that $x \in xH$. Since the union of left cosets of H contains all left cosets, this union will contain the coset xH. Thus, x will be a member of that union of left cosets of H, $x \in \bigcup_{g \in G} gH$. ∎

Therefore, the left cosets of H in G partition G. ∎

2. General Mathematical Remarks

This proof is an important proof in group theory, because it is a necessary result that we use in proving Lagrange's Theorem, arguably the most central theorem in group theory. The theorem states that for any finite group G, the order of every subgroup of G (that is, the number of elements) divides the order of G. Some important concepts used in this paper/my code include things like equivalence relations and cosets, which are used widely in group theory beyond just proving Lagrange's Theorem.

3. Comments on Lean Implementation

One of the trickiest things about working with group theory in Lean if you are defining your groups yourself and not relying on *mathlib* is having to implement many basic rules that we take for granted on paper. For example, on paper, we might say we “pick an arbitrary element from a group” but implicitly, this is only allowed because we assume that a group is non-empty. In Lean, in order to be able to specify an arbitrary element you want to pick, you will have to implement an axiom saying that every group is non-empty. There are some other small details like this which I keep having to go back to my group definition and add new axioms or lemmas to make it possible for me to use them in my proofs, such as the subgroup operation being the same as the group operation, and that the inverse operation in subgroups is the same as the inverse operation in groups.

Another difficult thing was in defining new structures or lemmas/theorems, I was not familiar with the syntax of setting them up, since homework in class usually has these already set up. Therefore, learning what arguments are necessary, declaring the types of objects we are working with and specifying our variables (and in the right order to do these things) took me quite a while to get used to. Luckily, one can learn them from looking at our past homework and lecture notes and figuring out by trial and error what each piece does.

In general, working with group theory in Lean is quite tedious compared to working on paper because we tend to gloss over things like bracket order, operations, etc. This was one of the things that took up most of my time, which was making Lean happy with the syntax so I could apply operations on them in exactly the way they are strictly defined. This was what made the lemma *disjoint_cosets* so lengthy in the code, because to pick something arbitrarily on paper, or to assume a certain equation requires using the *have* comment, which then requires us to prove/justify why it is legitimate for us to have that certain statement.

4. Future Improvements

There are a few (many!) things one can improve on in my code. Firstly, and perhaps most fundamentally, is to not define a subgroup with a separate definition—this caused me to have to redefine every operation and axiom that we already have in the group. Instead, one should define the subgroup as *inheriting* all these properties, ready to use. This also caused it to be very tedious for me to have to put the specifier “*H.[command]”* or “*subgroup.[command]*” because it is defined separately from the group commands. This makes things like *subgroup.mul_inverse* redundant because you already have *mygroup.mul_inverse*.

Another improvement that can be made is to use a different tactic rather than declaring *have* for any statement we want to use. It would be far more efficient to minimally declare *have* for statements that we only need, and to modify the existing ones as we go. This was the problem in my *disjoint_cosets* lemma and *subgroup.inverse_products* lemma.