Notes on Group Theory:

Rubik's Cube and the Permutation Group

Coding Club

September 11, 2024

Mohammed Alshamsi
2021004826
mo.alshamsi@aurak.ac.ae

Department of Computer Science and Engineering
American University of Ras Al Khaimah
2023–24

Contents

1 Introduction

Welcome to the third set of "lecture notes" for the AURAK Coding Club. The idea is to introduce a useful topic that lends itself easily to coding applications. These topics tend to be mathematical in nature, but most (if not all) of it will be intuitive material that doesn't require much background knowledge.

1.1 How to Read this Document

This is essentially just a short monologue about group theory and related concepts. We'll go over the elementary definitions and results, and give some examples. This is by no means meant to give you a comprehensive introduction; it's just a quick tour to get you started. Check the bibliography if you want to learn more about the topic.

I'll be sure to bring up coding aplications whenever it makes sense. Any code I write at those points will be in the C language. If you've taken (or are taking) CSCI 112, you'll be able to follow along without much trouble. Furthermore, there will be coding exercises. You're free to solve them in any language.

If you don't know any coding, check this link for a quick intro to C.

1.2 Sets

Briefly, a group is a set that has some additional structure, so the following few subsections will review sets and introduce a few related concepts that will be helpful later. Feel free to skim this section to confirm what you (don't?) know.

Sets. The concept of a "set" is no doubt very familiar to you, at least intuitively. It's simply a collection of things—a collection that is *well-defined* in the sense that it's always clear whether a given object is (or is not) a part of that collection.

Sets are denoted by curly braces $\{\}$, with their *elements* (the "things" in them) being contained between the braces and separated by commas. For instance,

$$S = \{a, b, c\}$$

is a set consisting of the elements a, b and c.

The set S was defined *explicitly*. This means its constituent elements were all listed. If you were presented with c, for example, you would very easily be able to say that "c is an element of S", or "$c \in S$", for short. Had you been presented with d, you would also be able to say that "d is not an element of S", or "$d \notin S$".

You could also define sets implicitly, by defining a "property" that is held by the elements of that set, and by nothing else:

$$\{x \mid x \text{ is a member of the AURAK coding club's management as of September 11, 2024.}\}$$

This reads as "The set of all x, where x is a [...]". Of course, descriptions like this need additional context that should be provided by the reader's knowledge.

Another way to implicitly define sets is by using the "..." notation, for instance

$$S = \{1, \ldots, 16\}.$$

But this, too, may rely on the reader's knowledge (or even be vague, leaving it to the reader to guess the right "rule" to identify the set by). For example, is S the set of positive integers between 1 and 16 inclusive, or perhaps the set $\{1, 2, 4, 8, 16\}$ of the first five nonnegative powers of 2? Or the set $\{1, 4, 16\}$ of powers of 4?

Subsets. Given two sets S and T, we say that T is a *subset* of T if every element of T is in S. This is denoted by $T \subseteq S$, and if T is not a subset of S (meaning it contains elements that are outside of S), we denote that by $T \not\subseteq S$. Here we list all the subsets of S:

$$\{\} \qquad \{a\} \qquad \{b\} \qquad \{c\}$$
$$\{a, b\} \qquad \{a, c\} \qquad \{b, c\} \qquad \{a, b, c\}$$

Note that S itself is also there; every set is a subset of itself. Also note the presence of the empty set $\{\}$, which is usually denoted by $\emptyset$. This set contains nothing, but is a subset of every other set.

Exercise 1. For a set of n elements, how many subsets does it have?

Exercise 2. Is the empty set a subset of itself?

Operations on sets. There's three primary operations that involve sets:

1. $A - B$ is the set of all elements of A that are not in B. This may also be denoted $A \setminus B$, and is named the *difference* of A and B[1]or the *relative complement* of B in A.

   You may also subtract individual elements from a set. Knowing that $\mathbb{Z}$ is the set of all integers, $\mathbb{Z} - \{2\}$ is the set of all integers I don't like.

2. $A \cap B$ is the *intersection* of A and B, and is the set of all elements of A that are also in B (or vice versa; it's the same thing).

3. $A \cup B$ is the *union* of A and B, and is the set of all elements of A and all elements of B.

1.3 Functions

Next, we build up to the fundamental notion of a function.

Cartesian products. An important property of sets is that "order doesn't matter". For a quick example, the sets $\{a, b\}$ and $\{b, a\}$ are equal. So, how do we express the notion of order?

There's a different notation for this, which is (a, b), and is called an *ordered couple*, or an *ordered pair*. It establishes a as the "first" element, and b as the "second" element.[2] We may also

[1]Yes, this may be vague in that it may mean $A - B$ or $B - A$.

[2]This can actually be defined in terms of sets, too:

$$\{\{a\}, \{a, b\}\}.$$

There are other methods as well.

have ordered triples, 4-tuples, or in general n-tuples for any natural number n, following the same concept of "first, second, third, ...". If we don't want to specify that n be a fixed number, we may also simply say "tuple".

The cartesian products of two sets X and Y is denoted $X \times Y$, and refers to the set of all ordered pairs (x, y) where $x \in X$ and $y \in Y$. A good example is the xy-plane you're familiar with; it's the cartesian product of the set of real numbers $\mathbb{R}$ with itself. In cases like that, there's a notational shortcut that tends to be used: $\mathbb{R}^2$ is the same as $\mathbb{R} \times \mathbb{R}$.

Functions. A *function* f from a set X to a set Y is a "rule" that assigns to each element x of X exactly one element $f(x)$ of Y. X is said to be the *domain* of the function, and Y is said to be its *codomain*. This is summarized by the notation $f : X \to Y$. We call $f(x)$ the *value* of f at x.

Functions act on single elements of sets, but these elements may be anything; even tuples. So we can do something like $f : X \times Y \to Z$, assigning for each pair (x, y) in the domain $X \times Y$ an element of the codomain Z. Note that multiple elements of the domain can map to the same element of the codomain, but for a given element of the codomain, there is at most one element of the domain that maps to it. This means some elements of the codomain cannot be "reached" by the function; for instance, $f : \mathbb{Z} \to \mathbb{Z}$ where the rule is doubling (so 2 goes to 4). The codomain is $\mathbb{Z}$, but all the odd numbers have no elements mapping to them. We call the set of all "reachable" elements in the codomain the *range* of the function, or alternatively, its *image*.

We'll use the word "operation" often when discussing groups, so now is a good time to define them. An operation $*$ on a set X is simply a function mapping $X \times X$ to X; that is, $* : X \times X \to X$. Keep in mind that they're just a type of function that occurs often enough to warrant having more convenient notation for them. In the case of operations, we don't write $*(x_1, x_2)$ to denote the value of $*$ at (x_1, x_2)—instead, we write $x_1 * x_2$.

Injection, Surjection, Bijection. Big words.

2 What is a Group?

Here we define what a group is, and provide some examples.

2.1 Cayley Tables

These are a nice way to represent smaller groups. We list a few properties of them that indicate things about the group (e.g. symmetric means it's abelian).

2.2 Subgroups

A subgroup is just a subset that is also a group under the same operation. Okay, next section.

2.3 Generators and Cyclic Groups

We define generators and cyclic groups, as well as generating sets of not-necessarily-cyclic groups. We note that their ability to "generate" the group relies heavily on the operation in question.

2.4 Groups of Permutations

Yeah these are pretty cool. We'll define groups of permutations, show the two slick notations, and maybe the algorithm(s?) from TAOCP S1.3.3. (The scope creep has already begun, but it won't be as bad as the whole string search algorithm from last time... hopefully.)

3 Cosets and Quotient Groups

As usual, definitions and examples, but we *might* want to break this down a little with additional examples to make sure it really sticks.

3.1 Lagrange's Theorem

This is so cool.

4 The Rubik's Cube Group

We'll introduce the notations for moves and how the moves are actually generators for the whole set. I think we might want to talk about group actions before this point but I'm not yet sure.

4.1 Coset Spaces

I don't know what these are yet.

4.2 Thistlethwaite's Algorithm

But this algorithm uses coset spaces, so we need to talk about those before we introduce it.