

Theory of Computing Notes

Chapter 1: Introduction

Fraser Watt

2022-09-02

Chapter Notes

Sets and Notation

With set notation being represented with curly braces, an empty set would be $\{\}$, whilst a collection which only holds that empty set would be $\{\{\}\}$. Other important set notation:

- In / not in: \in , \notin .
- x as a subset of y : $x \subset y$.
- Natural numbers (positive integers): N .

Set logic might read something like this:

$$\{2n | n \in N\}$$

The above would be read as “ $2n$, given n is an element within the set of all natural numbers.”

Fraser’s Q: Why does $N - \{0\}$ denote the set of all positive numbers, and not something like $N \neq 0$?

- They’re equivalent. Whilst $N - \{0\}$ means the set of all natural numbers *except* a set only containing 0, $N \neq 0$ is just all natural numbers which do not equal 0.
- You might prefer the first format as it’s more flexible - you can just add more numbers to exclude. E.g. $N - \{0, 1, 2\}$

Joining Sets

- Intersection: Take elements common to both sets, written as $A \cap B$. This is like a SQL JOIN
- Union: New set includes all unique / distinct values of component sets, written as $A \cup B$. This is like a SQL FULL JOIN
- Cartesian Product: Takes elements from both sets and returns every possible combination from them, written as $A \times B$. This is like a SQL CROSS JOIN

Relations

Throughout the book, there's a focus on using strings. Strings (words) are a collection of symbols where the symbols are all taken from a fixed alphabet. The alphabet can be thought of as a set!

- Length of word w would be represented as $|w|$.
- Reverse of word w would be represented as w^R (palindrome).

The complement of a set is *everything that isn't in a given set*. So if A was a set of all natural numbers, $A' = \{n | n \notin N\}$. This is sort of like writing `!(yourLogicalStatement)` in Javascript, but $!$ in mathematical notation is factorial multiplication.

Transitive, Reflexive, and Symmetric relations.

- **Transitive:** Modular arithmetic (% in Python) useful to deconstruct this: if $4 \bmod 3 = 10 \bmod 3$ and $10 \bmod 3 = 13 \bmod 3$, then there is a transitive relation which says that $4 \bmod 3 = 13 \bmod 3$.
- **Symmetric:** If you have equivalence from $a \rightarrow b$ and another arrow from $b \rightarrow a$, then this is symmetric. Again, all or nothing — it is symmetric if for every arrow drawing equivalence, there is an equal and opposing relation backwards.
- **Reflexive:** If we are also going to use modular arithmetic here too, then we would be looking more like $10 \bmod 3 = 10 \bmod 3$.
 - If you add more arrows going to other items, it doesn't matter — still reflexive! On the other hand, if you have two sets $\{a, b, c, d\}$, and you have arrows going from both a 's, b 's, & c 's — but no equivalence between d_1 and d_2 — this is not reflexive. It's all or nothing!
- **Equivalence:** If it has a Transitive, Symmetric and Reflexive relation, then we can say it has an equivalence relation. I.e. $A = B$

Exercises

Exercise 1.1: Give the notation for a set that contains only the page number of this exercise (page 9).

$$R = \{9\}$$

Exercise 1.2: True or False

- a) $\{2, 4, 5\} \subset \{2n \mid n \in N\}$ False
- b) $\{e, i, t, c\} \subset \{a, \dots, z\}$ True
- c) $\{2n \mid \bar{n} \in N\} = \{2n + 1 \mid n \in N\}$ True
- d) $\{1, 2, 3, 4, 5\} = \{1, \dots, 5\}$ True

Exercise 1.3: Give a representation for the odd natural numbers.

$$\{n \mid \frac{n}{2} \notin N\}$$

Exercise 1.4:

Exercise 1.5: Let $A = \{1, 2, \dots, 100\}$ and $B = \{a, b, c, d\}$. How large is $A \times B$?

$N = 400$

Exercise 1.6:

Exercise 1.7: