Lecture 3

Ben Rosenberg

June 9, 2021

Mathematical background (cont.)

From last time:

Definition of proper subset

We say that $A \subset B$ iff $A \subseteq B \wedge A \neq B$, which is the same as $A \subseteq B \wedge B \nsubseteq A$

Theorem: $A \subset B \implies (\exists x)(x \in B \wedge x \notin A)$

Now we return to the main curriculum.

We are going to discover the number of elements in the power set.

Theorem: If $|A| = m$, then $|\mathcal{P}(A)| = 2^m$.

Proof:

Think of a set as a collection of icons, $A = a_1, a_2, a_3, \ldots, a_{m-1}, a_m$. Underneath that line we have one pixel or area for every icon, which can be lit or not lit – on, or off. Then, we can represent a subset of A as the set of icons a_i with their light on. Every subset corresponds, in essence, to a sequence of `1`s and `0`s.

So, we want to know how many possible strings of `1`s and `0`s there are given the fixed length of the string. There are as many subsets as there are *bit strings* of length m. Note that a bit is either `0` or `1`. There are two different ways to choose the first digit, and two different ways to choose the second digit and so on; therefore, there are 2^m ways to choose all m digits.

Definition: A quantifier with domain restriction such as $(\forall x \in D)(P(x))$ is the same as $(\forall x)(x \in D \implies P(x))$.

$(\forall x \in D)(P(x)) \Leftrightarrow (\forall x)(x \in D \implies P(x))$

$(\exists x \in D)(P(x)) \Leftrightarrow (\exists x)(x \in D \wedge P(x))$

Note that the above is somewhat counterintuitive. Rather than continuing the implication, we switch to the logical `and` operator.

We can use DeMorgan's laws to make this more clear:

$(\exists x \in D)(P(x)) \cong \neg(\forall x \in D)(\neg P(x)) = \neg(\forall x)(x \in D \implies \neg P(x))$ (by DeMorgan's laws, and definition of constrained quantifiers)

$\neg(\forall x)(x \in D \implies \neg P(x)) = \neg(\forall x)(\neg(x \in D \wedge P(x)))$ (by equivalences from last time)

$\neg(\forall x)(\neg(x \in D \wedge P(x))) = (\exists x)(x \in D \wedge P(x))$ (by DeMorgan's laws again)

□

Functions

Definition: A function $f : A \to B$, read as "a function f from A to B", where A and B are sets, is a subset $f \subset A \times B$ of ordered pairs from the Cartesian product of A and B, such that:

1. $(\forall x \in A)(\exists y \in B)((x, y) \in f)$
2. A is the **domain** of the function, and B is the **codomain** (or **target**) of the function
3. Given a pair (x, y) from the function, x is called the **argument**, and y is called the **image**
4. Every element in the domain has at least one image
5. $(\forall x \in A)(\forall y_1, y_2 \in B)((x, y_1) \in f \land (x, y_2) \in f) \implies y_1 = y_2$
    1. In other words, there cannot be two different images for the same element in the domain – each element has at least one image, and at most one image, and therefore *exactly one image* in B.

Another way to describe the relationship between f, x, and y, is that f "sends" x to y. More ways are $f(x) = y$ or $f : x \mapsto y$.

It is, however, possible, for a single element in the codomain to be mapped to by multiple elements in the domain, and for an element in the codomain to not be mapped to by any elements in the domain.

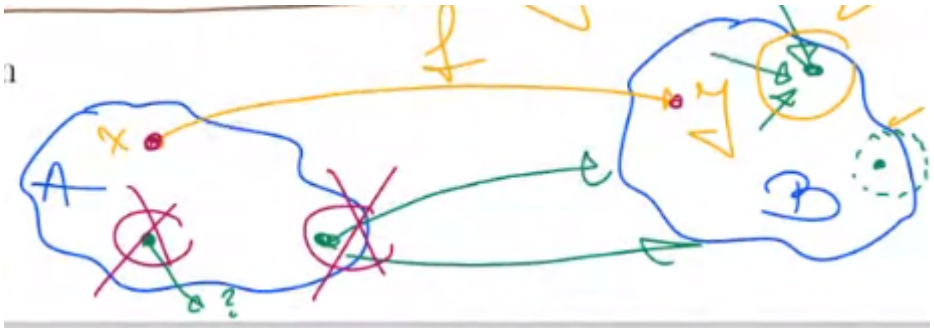

Figure 1: Illustration of a function

An analogy for a function might be a gradesheet, which maps each student to a single grade. Not every grade needs to be given out, and students can be given the same grade.

Definition: The range of a function $f : A \to B$ is the following set:

$$f(A) = \{b \in B | (\exists a \in A)(b = f(a))\}$$

For the gradesheet analogy, the range is the set of all grades that have been given to students. For a function, the range is the set of all values in the target or codomain that have actually been assigned – for which there exists an argument that maps to the value.

Injective functions

Example: Make a function to assign parking spots to cars, so that every car knows where to park. The function goes from car to parking spot.

This is an issue because two cars might get the same spot, which would not go well. In order to deal with this, we define injective functions.

Definition: An **injective function** $f : A \to B$ (also known as one-to-one or $1 : 1$) has the following properties:

1. All of the propoerties of functions described above, and
2. $(\forall x, y \in A)((f(x) = f(y)) \implies x = y)$

In other words, no output can have more than one input assigned to it. In our parking example, this would be "if any two cars are assigned to the same spot, they have to be the same car."

Function inverses

Example: Consider a function $f : S \to G$ from students to grades, with ordered pairs $(s_1, g(s_1)), (s_2, g(s_2)), \ldots$.

Then, we can construct a set $f^{-1} \subseteq G \times S$ by transposing the function "table" as follows:

$$(g, s) \in f^{-1} \Leftrightarrow (s, g) \in f$$

Is this still a function? No. This is because multiple students can have the same grade, which means that multiple outputs come from the same input which violates the definition of a function. It violates both the rules of "at least one output for each input" and "at most one output for each input", as there might be a grade which nobody receives, and there might be a grade that multiple students receive.

Theorem: If $f : A \to B$ is an injection, then f^{-1} (as defined above) is a function: $f^{-1} : f(A) \to A$, where $f(A) \subseteq B$ is the range of f in B, *and* f^{-1} is injective. We can't have issues with multiple mappings because f is injective, and we can't have issues with elements not being mapped, since the domain of f^{-1} is the range of f, $f(A)$. Lastly, we can't have issues in which multiple elements are mapped to because f is a function.

f^{-1} is called the **partial inverse function**.

Definition: A **partial function** is defined as follows:

Given a function $f : D \to N$ where $D \subseteq N$, we say that f is a partial function from N to N. When $x \in N \backslash D$, we say that $f(x)$ is undefined. If $D = N$, then f is *total*.

This definition allows us to work with functions that are occasionally undefined for certain values in the domain, without knowledge of the relevant set $D \subseteq N$.

New definitions

Definition: An **alphabet** is any *finite* set. Typically, an alphabet is denoted with a capital Greek sigma Σ. An example alphabet might be $\Sigma = \{a, b, c\}$, although it should be noted that an alphabet may have any finite number of letters within it.

Definition: A **letter** is an element of the alphabet.

Definition: A **string** is any *finite* sequence of letters. An example string in the above-given alphabet might be `abbabbab`.

Definition: The **length** of a string is the number of letters in the string, denoted $|s|$ for some string s.

Definition: The empty string, denoted ε or λ, is the string with no letters. (By convention in this class, we use λ exclusively; in the textbook and elsewhere, ε is predominantly used.) Thus, $|\lambda| = 0$.

Definition: The concatenation of strings x and y, denoted $x \circ y$, is xy (x written next to y). For $x =$ `abca` and $y =$ `bbc`, $xy =$ `abcabbc`. $|xy| = |x + |y|$.

Definition: A **language** is any (possibly *infinite*) set of strings (over the given alphabet).

Induction and recursion

Note: We will never have to write an induction proof for something, but we will need to be able to think through the reasoning for that kind of proof.

First, we define the infinite language over alphabet $\Sigma = \{a, b, c\}$. Let L be the set of strings over Σ defined as follows:

1. $\texttt{cc} \in L$
2. If some string x belongs to L, then $\texttt{a}x\texttt{bb} \in L$.
3. L has no elements other than those obtained by the application of (1) and (2).

So:

- $\texttt{cc} \in L$
- $\texttt{c} \notin L$
  - c is not in L by rule (1)
  - c is not in L by rule (2). This is because we only concatenate letters to a good string. So, we need to start with a good string and add three more letters. So, length must be at least 3. Since $|\texttt{c}| = 1$, c could never come out of rule (2).
- $\texttt{cccc} \notin L$ by rule (1)
- cccc is not in L by rule (2) because rule (2) adds as and bs and so it cannot be in L
- acbb is not in L by rule (1) or (2) because c is not a good string.
- aaccbbbb and aaaccbbbbbb are both good strings, since they come from (2)'s recursive application.

Claim: Every good string has the form:

$$\texttt{a}^n\texttt{ccb}^{2n}$$

Proof: We will prove the above by induction, using the number of applications of rule (2).

We need to prove that every string that is obtained after n applications of (2) is $\texttt{a}^n\texttt{ccb}^{2n}$.

Base case: zero applications of (2). The only good string here is $\texttt{cc} = \texttt{a}^0\texttt{ccb}^{2\cdot 0}$. (Note that the convention here is that exponentiation is short for the number of concatenations of a letter in a string – $\texttt{x}^0 = \lambda$.)

Inductive hypothesis: Assume that the string obtained by n applications of (2) is $\texttt{a}^n\texttt{ccb}^{2n}$. We need to prove that the string obtained by $n+1$ applications of (2) is $\texttt{a}^{n+1}\texttt{ccb}^{2n+2}$.

Let $w \in L$ be the string obtained by $n+1$ applications of (2). Look at the last application: $\underbrace{\texttt{a}\boxed{z}\texttt{bb}}_{w}$.

$z \in L$, and z was obtained by n applications of (2).By the inductive hypothesis, $z = \texttt{a}^n\texttt{ccb}^{2n}$.

And so, $w = \texttt{azbb} = \texttt{aa}^n\texttt{ccb}^{2n}\texttt{bb} = \texttt{a}^{n+1}\texttt{ccb}^{2n+2}$.

□

Cardinality

Definition: Given two sets, A and B, we say that if the cardinality of A is less than or equal to the cardinality of B, we have

$$|A| \leq |B|,$$

if there exists an injective function (an injection) from A to B. We can think about our parking lot analogy – there is an injective function if there are fewer cars than parking spots.

There are variants of this definition:

- $|A| = |B| \Leftrightarrow |A| \leq |B| \wedge |B| \leq |A|$. This means that there are injections from both A to B and from B to A.
  - If A and B have equal cardinalities, there are injections both ways.
- $|A| < |B| \implies |A| \leq |B| \wedge |B| \not\leq |A|$. This means that an injection exists from A to B, and an injection does not exist from B to A.

"How to fail": Cardinality of subsets of finite sets

In finite sets: If $S \subset A$, then $|S| < |A|$. We fail when we attempt to apply this logic to infinite sets.

Definition: A set is **infinite** if it has a proper subset of equal cardinality – that is, a set A is infinite if A has a proper subset $S \subset A$ such that $|S| = |A|$. This means that there are injections from S to A and from A to S.

Theorem: The set $\mathbb{N}$ of natural numbers is infinite.

Proof:

We need, by the definition of infinite sets, a proper subset $S \subset \mathbb{N}$n an injection from $S \to \mathbb{N}$, and an injection from $\mathbb{N} \to S$. These last two say that $|\mathbb{N}| = |S|$, which will allow us to prove (in conjunction with the fact that $S \subset \mathbb{N}$) that $\mathbb{N}$ is infinite.

First, we take $S = \{x \in \mathbb{N} | (\exists y \in N)(x = 2y)\}$. S is, in plain English, the set of even natural numbers.

We will continue constructing these injections tomorrow.