Alex Hubers
CSC:341 Spring 2024

Lecture Notes #8B

#1 Motivation

Later (after Spring break), we will be showing certain languages to be undecidable through a method called diagonalization. You have probably seen a diagonalization proof before—it is classically given in discrete math classes to show that the real numbers are uncountable. In other words: there are "more" real numbers than there are naturals or integers.

Why learn this?

1. To be honest, I care much more about you leaving this course with refreshed discrete math skills than I do about you remembering the theory of computation.
2. Our first few proofs of undecidability will rely on diagonalization. Our n_{th} proofs will rely on reducing problems to undecidable ones. That is, suppose the language B is undecidable. We can prove that the language A is undecidable if, under the assumption that we have some M that decides A, we can decide B. This is called a reduction, and is probably the most important idea you will learn in this course. (Of course, you already know it, really! We are just solving one problem with another.)

#2 Math Literacy Part II: All about functions

Let's review the tools we need to formally show that A_{TM} is undecidable. The following definitions are necessary.

A review of functions.

Definition 1 (Function)**.** *Let $f \in A \times B$. We say that f is a total function with domain A and codomain B, written $f : A \to B$, if:*

1. ***Totality:*** *For all $a \in A$ there exists $b \in B$ such that $(a, b) \in f$. We write $f(a) = b$ in such a case.*
2. ***Functionality:*** *For all $x, y \in A$, if $x = y$ then $f(x) = f(y)$.*

If for some $a \in A$ there does not exist $b \in B$ such that $f(a) = b$, we say that f is a partial function. For example, $f(x) = 1/x$ is partial on $x = 0$.

The **totality** and **functionality** properties above are dual to surjectivity and injectivity. Duality is an order theoretic term that more or less means "reversed". Can you see a parallel between the two above and the two below?

Definition 2 (surjectivity)**.** *A function $f : A \to B$ is surjective if, for every $b \in B$, there exists $a \in A$ such that $f(a) = b$.*

Definition 3 (injectivity)**.** *A function f is injective if, for every $x, y \in A$, if $f(x) = f(y)$ then $x = y$.*

Finally, a function f is called bijective if it is both injective and surjective.

Sets are considered to be bijective or in bijection if there exists a bijection between them. Bijections are how we measure "size" or cardinality of infinite sets.

One equivalent way (and in fact, more general) way of defining bijectivity is instead as invertibility. This equivalence is so well understood that mathematicians will often use the words "bijection" and “invertible” interchangeably. The latter is more commonly used because we can generalized bijections to *arbitrary* mathematical objects. Two mathematical objects are said to be isomorphic if they can be mapped to one another in an invertible fashion.

Definition 4 (Invertibility). *A function $f : A \to B$ is invertible if there exists $f^{-1} : B \to A$ such that*

1. *$\forall a \in A$ we have $f^{-1}(f(a)) = a$, and*
2. *$\forall b \in B$ we have $f(f^{-1}(b)) = b$.*

Theorem 1 (Bijectivity equals invertibility). *A function $f : A \to B$ is invertible iff A and B are bijective under f.*

Let’s pause to let these ideas sit with us.

Which of these are bijective? Why or why not? If not, which property do they break?

1. $f(x) = 3x + 2$
2. $f(x) = e^x$
3. $f(x) = x^2$
4. $f(x) = |x|$ for $f : \mathbb{Z} \to \mathbb{N}$.
5. $f(x) = \frac{1}{1+e^x}$ for $f : [0, 1] \to \mathbb{R}$. (This last one is bijective, but I only found it after googling for “bijection from [0, 1] to R”.)

#3 Diagonalization

The next definition is important.

Definition 5 (Countable). *A set A is countable if it is finite or it is bijective to $\mathbb{N}$.*

If A is infinite and countable then we also call it countably infinite. If it is infinite and uncountable, we call it uncountably infinite.

Here is the big claim we want to show.

Theorem 2. *The real numbers $\mathbb{R}$ are uncountable.*

In other words, there are “more” real numbers than there are natural numbers. Or, in other words, there are different “sizes” of infinity. The reason we study this now is that we will use the same technique (called diagonalization) to prove the claim.

Proof. We proceed by diagonalization. Towards a contradiction, suppose $\mathbb{R}$ is countable under bijection $f : \mathbb{N} \to [0, 1]$. (It suffices to show that even the interval $[0, 1]$ is uncountable.) In particular, represent each real number as a sequence x_i, where x_i represents the i^{th} decimal digit of the real number. For example, if $f(x) = 0.1415...$, then $x_0 = 1$, $x_1 = 4$, $x_2 = 1$, and so forth.

We will show that there exists a number $x \in [0, 1]$ that is not hit by f. In particular, construct this real number as the following sequence

$$x_i = i^{th} \text{ digit of } f(i) + 1 \mod 10$$

By example, if we have

n	f(n)
0	0.14159...
1	0.55555...
2	0.12345...
3	0.50000...

Then $x = 0.2641...$.

Now, suppose we have some $n \in \mathbb{N}$ such that $f(n) = x$. This leads to a contradiction, as each decimal place of $f(n)$, by construction, differs from each decimal place of x. Therefore no such f can "count" the interval $[0, 1]$.

Note that we showed only that $f : \mathbb{N} \to [0, 1]$ is uncountable, but this is enough: $[0, 1]$ is bijective to $\mathbb{R}$ (see examples above), and uncountability is preserved across bijection. □

Now, here is the final argument: although the set of all Turing machines is countable, the set of all languages is not. Therefore, there must be some languages unrecognizable by any Turing machine! We will cover this and related results in more depth after Spring break.

Claim 1. *There exists a language that is not Turing recognizable.*

Proof. The proof sketch goes like this:

1. Show a bijection from each (encoded) TM to $\mathbb{N}$. Therefore, TMs are countable.

2. Show by diagonalization that the set of all languages over alphabet Σ is uncountable.

Thus there are "more" languages than there are Turing machines, and so there must be some languages that are not recognized. □

For a full proof, see corrollary 4.18 in Sipser §4.2. One intuition as to why the set of languages over Σ is uncountable is that we are considering an infinite set of **infinite sets**. With TMs, we are considering an infinite set of **finite** strings.