CS40 Winter 2021 — Lecture Notes
Computability and Density

Shlomi Steinberg

February 16, 2021

1 The Relation Between Countability and Computability

We say that something is *computable*, if the logic that is required to generate it can be summarised as a finite artefact. That is, if we can write a finite sequence of instructions—an algorithm—that produce that object, then it is computable.

The study of computability aims to answer the most fundamental question in Computer Science: "Can we write an algorithm to produce a desired output?". This question comes before the more practical question, "If so, how well (in terms of resources) can we do so?". The question of computability is ultimately about classifying problems into a pair of simple classes of *feasible*, and *infeasible*. A feasible, i.e. a computable, problem could theoretically be done with unlimited resources. For an infeasible problem no finite algorithm will exist, and no new computational model, new class of computers or even new physics will change that.

Theorem 1.1. *The set of all finite computer programs is countable.*

Proof. Let S be the finite set of all valid symbols used in some given programming language. Let a computer program of length $l > 0$ be an ordered sequence of l symbols from S, i.e. a string of length l over the alphabet S (it might not be syntactically valid, but we don't care). Then, we denote $Prog_l = S^l = S \times S \times \ldots \times S$ the set of all computer programs of length l. The set of all computer programs of any finite length is then $Prog = \bigcup_{l=1}^{\infty} Prog_l$, which is a countably infinite union of finite sets, and therefore is countable. □

This is a direct consequence of Cantor's diagonalization argument. We have used Cantor's diagonalization to show that the real numbers are uncountable, and a more general version was used to show that the powerset of any set will have greater cardinality than that set. We will now show additional immediate applications of that argument to computability.

Incomputable numbers

Lemma 1.2. *You have shown that the set of all functions $f : \mathbb{Z}^+ \to \{0, 1\}$ is not countable, however we have shown that all finite computer programs are countable. Therefore, for almost any such function f, we can not write a finite computer program that outputs f.*

Corollary 1.1. *Almost no real number is computable!*

Proof. Take an arbitrary real number in the interval $[0, 1)$. Its (possibly infinite) decimal expansion is a function from $\mathbb{Z}^+$ (the position in the decimal expansion) to the set of digits $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. The set of all those functions is clearly uncountable by Lemma 1.2. □

A computable real number is then a number such that a (finite) computer program can be written to compute it to some arbitrary precision.

The halting problem Given some computer program P (which can be understood as a function from its input domain to its output codomain), we ask the question: Does this program ever stop? Or does it run forever (due to, e.g., an infinite loop)? This is the famous *halting problem* and it is well-known that the problem is undecidable, that is we can not write a program that tells us if an input program P halts. We will use Cantor's diagonalization argument to prove so.

Lemma 1.3. *Halting problem is incomputable.*

Proof. Assume, by contradiction, that we can write a program `HALT`(P, x) that will always give us the correct boolean answer: Does P halt on input x, or not. Note that programs are just data, as mentioned any computer program can be represented as a finite string. Therefore, let's examine the behaviour of P when run on itself, i.e. when the input to the (arbitrary) program P is just P. To do so we define the program `TEST`(P) as follows:

```
procedure TEST(P)
    if HALT(P, P) = "yes" then
        loop forever;
    else
        Halt;
```

That is, if P halts on input P, then `TEST` loops forever, otherwise `TEST` halts. Under the assumption that the program `HALT` exists, `TEST` is a well-defined program and is very easy to construct. Consider now the behaviour `TEST(TEST)`:

1. Either it halts, which means that `HALT(TEST, TEST)` returned "yes", claiming that `TEST` does NOT halt when run on itself. But this contradicts this case where `TEST(TEST)` halts.

2. Otherwise, it does not halt. Again, this contradicts the definition of `TEST`.

Therefore, the assumption that the program `HALT` exists can not hold. □

Gödels Incompleteness Theorem Recall that mathematics is a formal system and the *meaning*, e.g., the truth value, of a statement in that system is based on some set of axioms. We ask the question: Is such a system consistent? That is, is it possible, in that system, to prove a proposition p as well as its negation $\neg p$. If so, we say the system is inconsistent. Remember, we can infer anything from the false statement $p \wedge \neg p$, therefore if our formal system is inconsistent, we can deduce whatever we wish. We also ask a second question: Is such a system complete? Given an arbitrary proposition p, can we deduce, in our formal system, that p or $\neg p$? That is, can any formal statement be proven or disproven. If so, then the system is said to be complete.

The remarkable conclusion that any formal system that is "sufficiently powerful" (a system that is rich enough to formalize the set of integers) can not be both consistent and complete. This is Gödels Incompleteness Theorem, and the interested reader is welcome to search for a proof online, as well as for the deep connection between the Halting problem and Gödels Incompleteness Theorem.

2 Dense Sets

Definition 2.1. A set $S \subseteq \mathbb{R}$ is said to be *dense*, if

$$\forall s_1, s_2 \in S(s_1 < s_2 \to \exists s \in S,\ s_1 < s < s_2)$$

That is, if between every pair of numbers in S, we can always find another number, then S is dense.

You may think about a dense set of numbers as a set that can not be drawn as discrete points on the real line. For example, the integers (or any subset of the integers) are discrete and not dense.

Corollary 2.1. *The set of real numbers, $\mathbb{R}$, is dense.*

Proof. Trivially by the closure of real numbers under addition and multiplication: Given $r_1 < r_2$ real numbers, $r = \frac{1}{2}(r_1 + r_2)$ fulfils $r_1 < r < r_2$. □

Corollary 2.2. *The set of rational numbers, $\mathbb{Q}$, is dense.*

Proof. Identical proof: Given $q_1 < q_2$ rational numbers, $q = \frac{1}{2}(q_1 + q_2)$ fulfils $q_1 < q < q_2$. To see that $q \in \mathbb{Q}$, write $q_1 = \frac{a}{b}, q_2 = \frac{c}{d}$, then

$$q = \frac{q_1 + q_2}{2} = \frac{\frac{a}{b} + \frac{c}{d}}{2} = \frac{\frac{ad+cb}{bd}}{2} = \frac{ad + cb}{2bd}$$

which must be a rational as $b \neq 0$ and $d \neq 0$ and thus $2bd \neq 0$. □

Lemma 2.1. *Given $x, y \in \mathbb{R}$ such that $y - x > 1$, then $\exists m \in \mathbb{Z}$ such that $x < m < y$.*

Proof. Let $m = \lceil x \rceil$ if $x \notin \mathbb{Z}$ and $m = x + 1$ otherwise. m is the minimal integer strictly greater than x. Then $x < m \leq x + 1 < y$. □

We continue with a simple but interesting result that should really challenge your intuition!

Corollary 2.3. *Between every pair of unequal real numbers, there exists a rational number. Or formally:*

$$\forall x, y \in \mathbb{R}(x < y \to \exists q \in \mathbb{Q},\ x < q < y)$$

Proof. Let $x < y$ be real numbers. Denote $\epsilon = r_2 - r_1 > 0$. Let $n \in \mathbb{Z}$ be an integer such that $0 < 1/n < \epsilon$. Thus $ny - nx > 1$ and by Lemma 2.1 $\exists m \in \mathbb{Z}$ such that $nx < m < ny$, and we immediately conclude that $x < \frac{m}{n} < y$ as requested. □

And yet, $|\mathbb{Q}| < |\mathbb{R}|$! Despite the fact that between every pair of real numbers there is a rational, we can still "cover" all the rational numbers with an interval of arbitrary short length:

Theorem 2.2. *Given any real $\epsilon > 0$, we can cover all the rational numbers using open intervals of the form $(a, b) \subset \mathbb{R}$, such that the length of all covering intervals is ϵ.*

Proof. Take an interval of length ϵ and denote it $I = (a, a + \epsilon)$, for any a. $\mathbb{Q}$ is countably infinite therefore there exists a bijection $f : \mathbb{N} \to \mathbb{Q}$.
Take half of the interval I, $I_0 = (a + \frac{\epsilon}{2}, a + \epsilon)$, and cover $f(0)$ with I_0.
We are left with $I \setminus I_0 = (a, a + \frac{\epsilon}{2}]$. Use half of that (discarding the point $a + \frac{\epsilon}{2}$), $I_1 = (a + \frac{\epsilon}{4}, a + \frac{\epsilon}{2})$, and cover $f(1)$ with I_1.
We can keep going, and for every $n \in \mathbb{N}$ we will have a non-empty interval $I_n \subset I$ to cover $f(n)$ with. Indeed, the total length of those intervals is

$$\sum_{n=0}^{\infty} |I_n| = \sum_{n=0}^{\infty} \epsilon \frac{1}{2^{n+1}} = \epsilon$$

□

We have shown that non-density is (somewhat surprisingly) not a necessary condition for countability. But, is it a sufficient condition? Can a set that is not dense anywhere be uncountable? No, as a counterexample we'll build the *Cantor set* later on class.