Recitation - Undecidability

1 Countability Cheat Sheet

There are 2 equivalent (and useful) definitions for "countable set":

1. Set A is countable if $|A| \leq |\mathbb{N}|$ (there is an injection from A to $\mathbb{N}$).
2. Set A is countable if $|A| \leq |\Sigma^*|$ for some finite alphabet Σ (A is encodable).

You are given a set A. Is it countable or uncountable?
Options for showing A is countable:

- Show A is "listable": The elements of A can be listed so that every element in A eventually appears in the list. This is equivalent to showing there is a surjection f from $\mathbb{N}$ to A: we define $f(n)$ as the n'th element in the list.
- Show $|A| \leq |B|$, where B is a known countable set like $\mathbb{Z}$, $\mathbb{Z} \times \mathbb{Z}$, $\mathbb{Q}$, $\mathbb{Q}[x]$, etc.
- Show that A is encodable, i.e. the elements of A have finite-length string representations/encodings.

Options for showing A is uncountable:

- Assume for the sake of contradiction that A is countable, then diagonalize against A to reach a contradiction.
- Show $|B| \leq |A|$, where B is a known uncountable set like $\{0,1\}^\infty$ (the set of all infinite-length binary strings). Note that $\{0,1\}^\infty \leftrightarrow \wp(\mathbb{N})$.

2 These Decidable Definitions Have Undecidable Ends

- A **decider** is a TM that halts on all inputs.
- A language L is **undecidable** if there is no decider TM such that $M(x)$ accepts if and only if $x \in L$.
- A language A **reduces** to B (i.e. solving A reduces to solving B) if it is possible to decide A using an algorithm that decides B as a subroutine. Denote this as $A \leq B$ (read: B can be used to solve A so A is *at most* as hard as B.)

3 Counting to Infinity

Is the set $\mathbb{N}^*$ countable? Does it make a difference if we replace $\mathbb{N}$ with another countable set?

Solution. https://www.youtube.com/embed/vXiG624kD3Q

First, let's make sure we have the correct understanding of what $\mathbb{N}^*$ is:

$$\mathbb{N}^* = \{(x_1, x_2, \dots, x_k) : k \in \mathbb{N},\ \forall i\ x_i \in \mathbb{N}\}.$$

So $\mathbb{N}^*$ denotes the set of all finite-length tuples where each coordinate is an element of $\mathbb{N}$.

We'll show $\mathbb{N}^*$ is encodable/countable. Consider the alphabet $\Sigma = \{\texttt{0}, \texttt{1}, \texttt{2}, \texttt{3}, \texttt{4}, \texttt{5}, \texttt{6}, \texttt{7}, \texttt{8}, \texttt{9}, \texttt{\$}\}$. Any element of $\mathbb{N}^*$ can be uniquely represented with a finite-length string where commas are replaced with $. (Note that there is really no need to put the opening and closing parentheses for the tuple.) To make the encoding definition a bit more explicit, if $\text{Enc} : \mathbb{N} \to \Sigma_9^*$ is the usual encoding of the naturals with digits (where $\Sigma_9 = \{\texttt{0}, \texttt{1}, \dots, \texttt{9}\}$), then $\text{Enc}' : \mathbb{N}^* \to \Sigma^*$ is defined as follows: for $(x_1, x_2, \dots, x_k) \in \mathbb{N}^*$,

$$\text{Enc}'((x_1, x_2, \dots, x_k)) = \text{Enc}(x_1)\texttt{\$}\text{Enc}(x_2)\texttt{\$} \dots \texttt{\$}\text{Enc}(x_k).$$

In this case, it is rather obvious that Enc' is an injective function (i.e. that every element of $\mathbb{N}^*$ maps to a unique finite-length string.) Or in other words, if two elements of $\mathbb{N}^*$ map to the same string, then it is clear that they represent the same element of $\mathbb{N}^*$. (Explicitly writing a proof of this would be too pedantic in our opinion.) This shows $\mathbb{N}^*$ is encodable, i.e. countable.

Similarly, we can conclude that A^* is countable whenever A is a countable set. ■

4 I'm Undecided About These

Prove that the following languages are undecidable (below, M, M_1, M_2 refer to TMs).

1. $\text{REGULAR} = \{\langle M \rangle : L(M) \text{ is regular}\}$.
2. $\text{DOLORES} = \{\langle M_1, M_2 \rangle : \exists w \in \Sigma^* \text{ such that both } M_1(w) \text{ and } M_2(w) \text{ accept}\}$.

Solution. https://www.youtube.com/embed/S0ci0tKs1DI

Part 1: We show that REGULAR is undecidable via a reduction from HALTS (i.e. we show that HALTS reduces to REGULAR).

Suppose M_{REG} decides REGULAR. We define a decider for HALTS as follows

```
def M_HALTS(<TM M, string x>):
    <HELP> =
    """def HELP(w):
        if w is of the form 0^n1^n for some natural number n:
            ACCEPT
        else:
            run M(x)
            ACCEPT"""
    return M_REG(<HELP>)
```

We now have to prove that M_{HALTS} is a correct decider for HALTS. In order to do this, we consider all possible inputs to M_{HALTS} and argue that it gives the correct output.

First, consider the case when the input is $\langle M, x\rangle$ such that $M(x)$ halts. Then notice that the helper TM we define accepts all strings, i.e. $L(HELP) = \Sigma^*$. Therefore $M_{\text{REG}}(\langle HELP\rangle)$ accepts (because Σ^* is a regular language), which means M_{HALTS} accepts, as desired.

Next, consider the case when the input is $\langle M, x\rangle$ such that $M(x)$ loops (in the context of TMs, the word "loops" is synonymous with "does not halt"). Then notice that $L(HELP) = \{0^n1^n | n \in \mathbb{N}\}$ is irregular. Therefore $M_{\text{REG}}(\langle HELP\rangle)$ rejects, which means M_{HALTS} rejects, as desired.

Thus, we've shown that HALTS $\leq$ REGULAR, so REGULAR is undecidable.

(*Note that there is also the case when the input string to M_{HALTS} does not correspond to a valid encoding of a TM M together with a string x. We would like to remind you once again that even though this is not explicitly written, we implicitly assume that the first thing our machine M_{HALTS} does is check whether the input is a valid encoding of a TM M together with a string x. If it is not, the machine rejects. If it is, then it will carry on with the specified instructions. Recall that this step of checking whether the input string has the correct form is never explicitly written. And in the proof of correctness, we do not have to explicitly argue what happens if the input string does not have the expected format since we can assume this step is handled properly.*)

Part 2: We show that DOLORES is undecidable via a reduction from HALTS (i.e. we show that HALTS reduces to DOLORES).

Suppose M_{DOLORES} decides DOLORES. We define a decider for HALTS as follows

```
def M_HALTS(<TM M, string x>):
    <HELP> =
    """def HELP(w):
        run M(x)
        ACCEPT"""
    return M_DOLORES(<HELP, HELP>)
```

We now argue we have a correct decider for HALTS as in the previous part.

Suppose the input is $\langle M, x\rangle$ such that $M(x)$ halts. Then $HELP$ accepts all inputs. Therefore $M_{\text{DOLORES}}(\langle HELP, HELP\rangle)$ accepts, which means M_{HALTS} accepts, as desired.

Suppose the input is $\langle M, x\rangle$ such that $M(x)$ loops. Then $HELP$ does not accept any inputs. Therefore $M_{\text{DOLORES}}(\langle HELP, HELP\rangle)$ rejects, which means M_{HALTS} rejects, as desired.

Thus, we've shown that HALTS $\leq$ DOLORES, so DOLORES is undecidable. ■

5 (Extra) Lose All Scripted Responses. Improvisation Only

Let FINITE $= \{\langle M\rangle : M$ is a TM and $L(M)$ is finite$\}$.
Let TOTAL $= \{\langle M\rangle : M$ is a TM which halts on all inputs$\}$.

Show that TOTAL $\leq$ FINITE.

Solution. https://www.youtube.com/embed/C7-r3sudvTU

Suppose that M_{FINITE} decides FINITE.

We define a decider for TOTAL. In the description below, we use the symbol <= (less than or equal to) to compare two strings. If string x is less than or equal to string y, it means x is lexicographically less than or equal to y (this is a total order on the set of all finite-length strings).

```
def M_TOTAL(<TM M>):
    <HELP> =
    """def HELP(w):
        for y <= w:
            run M(y)
        ACCEPT"""
    return not M_FINITE(<HELP>)
```

We now have to prove that M_{TOTAL} is a correct decider for TOTAL. In order to do this, we consider all possible inputs to M_{TOTAL} and argue that it gives the correct output.

Suppose the input string is $\langle M \rangle$ such that M halts on all inputs. Then notice that $L(HELP) = \Sigma^*$. Therefore $M_{\text{FINITE}}(\langle HELP \rangle)$ rejects, which means M_{TOTAL} accepts, as desired.

Suppose the input string is $\langle M \rangle$ such that M does not halt on all inputs. Let x be the lexicographically smallest string that M does not halt on. Then $HELP$ will not accept any string lexicographically greater (or equal to) than x, as $M(x)$ will not halt. Since there are only finitely many strings lexicographically smaller than x, $L(HELP)$ is finite. Thus, $M_{\text{FINITE}}(\langle HELP \rangle)$ accepts, which means M_{TOTAL} rejects, as desired. ■

6 (Bonus) Spicy Reals

Let S be a subset of the positive reals with the property that for every real number x, the set $\{y \in S : y > x\}$ has a minimum element. For example $\mathbb{N}^+$ is such a set, but $\mathbb{Q}^+$ is not. Determine whether S is necessarily countable.

Solution. `https://www.youtube.com/embed/bhjjkS_yASQ`

The set is countable. We'll use the fact that rationals are dense in the reals. We sketch the main idea without giving a full proof: Inject S to $\mathbb{Q}$ by mapping each element s of S to some rational between s and the next largest element of S.

Note: If you try to inject to $\mathbb{N}$ by mapping the smallest element of S to 0, the second smallest to 1, etc. it won't work. If $S = \{1, 1.5, 1.55, 1.555, \ldots, 3, \ldots\}$, the 3 will not be assigned to anything. ■