Handout 9a: Countability, Reductions, Proving Undecidability and Unrecognizability

Ananya Gandhi and Nicolas Hortiguera

COMS 3261 Fall 2022

Credit to Eli Goldin and Alan Du (Fall 2020 TAs)

1 Countability

Definition 1.1. A set S is countable if S is finite or $|S| = |\mathbb{N}|$. We say S is uncountable otherwise. Examples of countable sets include the set of all integers, $\mathbb{Z}$, and the set of all rational numbers, $\mathbb{Q}$.

Definition 1.2. Recall that two sets S and T have the same cardinality, denoted as $|S| = |T|$, if there exists a bijection $f : S \to T$. If no such bijection exists, then we say that S and T have different cardinalities and write $|S| \neq |T|$.

Definition 1.3. Recall the definitions of injective, surjective, and bijective functions from the Discrete Math Handout: Let S and T be sets and $f : S \to T$ be a function. Then we say that f is injective (or one-to-one), or an injection, if for any $x, y \in S, f(x) = f(y)$ implies that $x = y$.

We say that f is surjective (or onto), or a surjection, if for all $y \in T$, there exists some $x \in S$ such that $f(x) = y$

We say that f is bijective, or a bijection, if f is both injective and surjective.

Given these definitions, the above discussion translates to saying that a set S has n elements if and only if there exists a bijection $f : \{1, 2, ...n\} \to S$.

Example 1.4. Let $\mathbb{Z}$ be the set of all integers, and let N = 1, 2, 3, ... be the set of all positive integers. Then $|\mathbb{N}| = |\mathbb{Z}|$: consider the function$f : \mathbb{N} \to \mathbb{Z}$ given by:

$$f(n) = \begin{cases} \frac{n}{2} & \text{if n is even} \\ -\frac{n-1}{2} & \text{if n is odd} \end{cases}$$

Theorem 1.5. *Any countable union of countable sets is countable, i.e. if* $\{E_i\}_{n=1}^{\infty}$ *are countable sets, then*

$$S = \bigcup_{n=1}^{\infty} E_n$$

is countable.

Theorem 1.6. *Any finite product of countable sets is countable, i.e. if $E_1, ..., E_n$ are countable sets, then*

$$E_1 \times E_2 \times ... \times E_n = \{(a_1, ..., a_n) | a_i \in E_i\}$$

is countable.

Example 1.7. The rational numbers $\mathbb{Q}$ are countable.

Proof. We can consider every element of $\mathbb{Q}$ as a pair (p, q) written as $\frac{p}{q}$, where $p, q \in \mathbb{Z}$ and $q \neq 0$. Of course, there is some overlap: we say that $\frac{p}{q} = \frac{p'}{q'}$ if $pq' = p'q$. The upshot is that we can map every element of $\mathbb{Q}$ to a pair $(p, q) \subseteq \mathbb{Z} \times \mathbb{Z}$ injectively, e.g. by taking p, q to have greatest common divisor 1 and taking $q > 0$.

From Theorem 1.6, since $\mathbb{Z}$ is countable, $\mathbb{Z} \times \mathbb{Z}$ is countable. Also, since we just exhibited an injection $\mathbb{Q} \to \mathbb{Z} \times \mathbb{Z}$ we know that $\mathbb{Q}$ is in bijection with a subset of a countable set, hence $\mathbb{Q}$ is countable.

Example 1.8. The set of all infinite binary sequences, i.e. $\{(x_1, x_2, ...) | x_i \in \{0, 1\}\}$, which we shall denote $\{0, 1\}^{\mathbb{N}}$, is uncountable.

Proof. Suppose that this set were countable, then we can list its elements $\{x_1, x_2, ...\}$ where each $x_i \in \{0, 1\}^{\mathbb{N}}$. Then construct the infinite sequence $y = (y_1, y_2, ...) \in \{0, 1\}^{\mathbb{N}}$ where each y_i is the opposite of the i-th entry of x_i, i.e. if the i-th entry of $x_i = 0$, then $y_i = 1$, otherwise $y_i = 0$. Then y is not equal to any x_i because their i-th entries are different. Since we assumed that the list $\{x_1, x_2, ...\}$ contained all elements of $\{0, 1\}^{\mathbb{N}}$, that means $y \notin \{0, 1\}^{\mathbb{N}}$, but this is a contradiction. Hence, $\{0, 1\}^{\mathbb{N}}$ is uncountable. This kind of argument where we list the elements of a set, then change something along the diagonal, is known as **Cantor diagonalization**.

Example 1.9. Let Σ be a finite alphabet, then Σ^* is countable.

Proof. Notice that for every $n = 0, 1, ...$ that Σ^n is finite; for any string $s \in \Sigma^n$ there are $|\Sigma|$ choices for each character of s, so $|\Sigma^n| = |\Sigma|^n$ is finite. Now recall that

$$\Sigma^* = \bigcup_{n=0}^{\infty} \Sigma^n$$

is a countable union of finite sets, so Σ^* is countable as a result.

1.1 Examples of Countable Sets

- $\mathbb{N}$
- $\mathbb{Z}$, $2\mathbb{Z}$ (even numbers), primes, etc.
- $\mathbb{Q}$
- Σ^*, any language $L \subseteq \Sigma^*$ (for any finite alphabet Σ)

- $\{\langle M\rangle | M$ is a Turing Machine$\}$
- $\{L \subseteq \Sigma^* | L$ is decidable$\}$
- $\{L \subseteq \Sigma^* | L$ is recognizable$\}$
- Any set where each element in the set has a finite representation (note that all the above examples are a special case of this one).

1.2 Examples of Uncountable Sets

- $\mathbb{R}$
- $\mathcal{P}(\mathbb{N})$ (namely the power set of $\mathbb{N}$).
- The set of all languages over a finite alphabet, $\mathcal{P}(\Sigma^*)$
- The set of all undecidable languages
- The set of all unrecognizable languages

2 Turing Reductions and Undecidability

Definition 2.1. We say a language A is Turing-reducible to a language B, written $A \leq_T B$, if given an oracle that decides B, there exists a decider for A.

We use the $\leq$ sign here because, intuitively, A is "easier" than (or equal to) B. Essentially, $A \leq_T B$ means that if B is decidable, so is A. This gives us the following theorem:

Theorem 2.2. *If $A \leq_T B$ and B is decidable, then A is decidable.*

and taking the contrapositive gives us the following (very useful) corollary:

Corollary 2.3. *If $A \leq_T B$ and A is undecidable, then B is undecidable.*

Thus, if we know that a language is undecidable, we can use this to show many other languages are undecidable!

2.1 Examples of undecidable languages

-$A_{TM} = \{\langle M, w\rangle | M$ is a TM and M accepts $w\}$
-$E_{TM} = \{\langle M\rangle | M$ is a TM and $L(M) = \emptyset\}$
-$HALT_{TM} = \{\langle M, w\rangle | M$ is a TM and M halts on $w\}$
-$EQ_{TM} = \{\langle M_1, M_2\rangle | M_1, M_2$ are TMs and $L(M_1) = L(M_2)\}$

2.2 Exercises

1. Prove that $HALT_{TM} \leq_T A_{TM}$. (Note: in class we proved the other direction, namely that $A_{TM} \leq_T HALT_{TM}$).

2. Prove that $L = \{\langle M, D\rangle | M$ is a TM, D is a DFA, and $L(M) = L(D)\}$ is undecidable.

3. Prove that the following are equivalent
   1) $A \leq_T B$
   2) $\overline{A} \leq_T B$
   3) $\overline{A} \leq_T \overline{B}$
   4) $A \leq_T \overline{B}$

3 Using Rice's Theorem to prove undecidability

Note that many of the undecidable languages we have learned about fit a common pattern. That is, they are languages of the form $\{\langle M\rangle | M$ is a TM and $L(M)$ satisfies...$\}$. We call such languages "properties of recognizable languages." Formally,

Definition 3.1. P is a property of recognizable languages if
-$P \subseteq \{\langle M\rangle | M$ is a TM$\}$
-If M_1, M_2 are TMs and $L(M_1) = L(M_2)$, then $\langle M_1\rangle \in P \iff \langle M_2\rangle \in P$.

In fact, there is a very convenient way to tell if a language of this form is decidable, using the following theorem:

Theorem 3.2. *(Rice's Theorem) Let P be a non-trivial property of recognizable languages. That is, P is a property of recognizable languages such that $P \neq \emptyset$ and $P \neq \{\langle M\rangle | M$ is a TM$\}$. Then P is not decidable.*

We can prove the theorem by showing that A_{TM} Turing-reduces to any non-trivial P. Since A_{TM} is undecidable, P cannot be decidable. (The full proof was given in Lecture 21, Nov 28).

Example:

Since Rice's theorem seems to apply to a very broad class of languages, it is worth pointing out some types of languages to which it does **not** apply:

- Languages (decision problems) where the elements (inputs) are not encodings of TMs $\langle M\rangle$. This is because such languages do not represent TM properties.

- Languages for which the TM property depends on the implementation (for example, $\{\langle M\rangle|$M is a TM that always moves right$\}$

3.1 Exercises

Identifying cases where Rice's theorem applies:

1. Does Rice's theorem apply to $L = \{\langle M\rangle | M$ is a TM and M accepts 0$\}$?
2. Does Rice's theorem apply to $L = \{\langle M\rangle | M$ is a TM and M has exactly two states$\}$?
3. Does Rice's theorem apply to $L = \{\langle M\rangle | M$ is a TM and M rejects 0$\}$?
4. Does Rice's theorem apply to $E_{TM} = \{\langle M\rangle | M$ is a TM and $L(M) = \emptyset\}$?
5. Does Rice's theorem apply to $L = \{\langle M\rangle | M$ is a TM and $L(M) = \overline{A_{TM}}\}$?
6. Does Rice's theorem apply to $L = \{\langle M\rangle | M$ is a TM and $L(M)$ is recognizable$\}$?
7. Does Rice's theorem apply to $L = \{\langle M\rangle | M$ is a TM and $L(M)$ is decidable$\}$?

Note that in the above exercises, if the answer is yes, then the language is not decidable. If the answer is no, then the language may or may not be decidable (try to figure out which is the case in each of the above).

Using Rice's theorem to prove undecidability: (Problem 5.18 in Sipser, p. 240)
Use Rice's theorem to prove the undecidability of the following language:
$INFINITE_{TM} = \{\langle M\rangle | M$ is a TM and $L(M)$ is an infinite language$\}$

4 Proving L is unrecognizable - Overview:

Recall that a language L is recognizable if and only if there exists a TM that accepts strings in that language and does not accept strings that are not in that language (namely if $x \in L$ the TM accepts x, and if $x \notin L$, the TM may either reject x or run forever on x).

There are different ways to prove that L is not recognizable:

1. Use a mapping reduction from another language that is not recognizable
2. Show that the language L is not decidable (see above for how to do this) and then show that $\overline{L}$ is recognizable.
3. (Not part of the material in the class) Using refined Rice's theorem
4. (Not covered in this review) Using diagonalization

5 Using complements and undecidability to prove unrecognizability

One way to prove that L is unrecognizable is to prove that L is undecidable (using one of the techniques above) and that $\overline{L}$ is recognizable.

Recall that L and $\overline{L}$ are recognizable if and only if L is decidable. Hence, if we show that L is not decidable, then it must be the case that at least one of L or $\overline{L}$ is unrecognizable. If we also prove that $\overline{L}$ is recognizable, then it must be the case that L is unrecognizable; otherwise, we would have "L and $\overline{L}$ are recognizable $\implies$ L is decidable", which would be a contradiction given the proof of undecidability of L.

Example: $\overline{A_{TM}}$ = The complement of $\{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ accepts } w\}$ is unrecognizable.

We know that A_{TM} is Turing-recognizable (proved in Lecture 18, Nov 14). We also know that A_{TM} is undecidable (proved in Lecture 19, Nov 21). If $\overline{A_{TM}}$ was Turing-recognizable, we would have that A_{TM} is decidable, which is a contradiction. Hence, $\overline{A_{TM}}$ must be unrecognizable.

6 Mapping reductions for unrecognizability

Definition 6.1. We say a language A is mapping-reducible to a language B, written $A \leq_m B$, if there exists some computable function $f : \Sigma^* \to \Sigma^*$ such that, for every $w \in \Sigma^*$,

$$w \in A \iff f(w) \in B.$$

Theorem 6.2. *If $A \leq_m B$, then $A \leq_T B$.*

The theorem above says that we can view mapping reductions as a special case of Turing reductions. The idea for the proof of this statement is that if $A \leq_m B$ via f and $\mathcal{O}$ is a decider for B, we can construct a decider R for A. Decider R takes input x to problem A, computes $y = f(X)$, and runs $\mathcal{O}(y)$, finally returning the same result. Thanks to the computability of f, R can be implemented as a Turing machine. Moreover, R will accept x if and only if $x \in A$ (try to show this is the case).

Theorem 6.3. *If $A \leq_m B$ and B is recognizable, then A is recognizable.*

Proof. The idea is the following: if $A \leq_m B$, then we have a Turing-reduction in the format as above: $A \leq_T B$. In reductions of that form, if we plug in a recognizer for B as $\mathcal{O}$, we get a recognizer for A. Thus, if B is recognizable, so is A. A more detailed proof follows.

Assume that $A \leq_m B$ via mapping f, and that B is recognized by some TM T. Then we can construct a recognizer R for A. On input x, R computes $y = f(x)$ and runs $T(y)$, returning the same output. We know that we can implement T because f is computable, and T is assumed to be a TM. We show that R recognizes A:

- If $x \in A$, $y = f(x) \in B$ by $A \leq_m B$. Therefore, y is accepted by T for B. Hence, the recognizer R also accepts x.
- If $x \notin A$, $y = f(x) \notin B$ by $A \leq_m B$. Hence, $T(y)$ either rejects or runs forever, which means that R either rejects or runs forever. In any case, R does not accept x.

$\square$

Corollary 6.4. *If $A \leq_m B$ and A is unrecognizable, then B is unrecognizable.*

The above corollary is equivalent to the previous theorem, since it is the contrapositive statement. This corollary is very relevant because it gives us a strategy to prove that L is unrecognizable. If we can find a language A that is unrecognizable and that is mapping-reducible to L (that is, $A \leq_m L$), then L is unrecognizable.

Theorem 6.5. $A \leq_m B \iff \overline{A} \leq_m \overline{B}$

The above theorem gives another related strategy to prove the unrecognizability of L by working through $\overline{L}$ and a language A whose complement is mapping-reducible to $\overline{L}$. If we prove that $\overline{A} \leq_m \overline{L}$ and that A is unrecognizable, we have that L is unrecognizable.

6.1 Exercises

1. Prove that $L = \{\langle M, D\rangle | M \text{ is a TM}, D \text{ is a DFA, and } L(M) = L(D)\}$ is not co-recognizable. That is, prove that $\overline{L}$ is not recognizable.
2. Prove that $L = \{\langle M\rangle | M \text{ does not accept strings of length } \geq 50\}$ is not recognizable.
3. Let A be a language. Prove that $A \leq_m A$.
4. Is it necessarily true that $A \leq_m \overline{A}$?

7 Examples of unrecognizable languages

- $E_{TM} = \{\langle M\rangle \mid M \text{ is a TM and } L(M) = \emptyset\}$
- $EQ_{TM} = \{\langle M_1, M_2\rangle \mid M_1 \text{ and } M_2 \text{ are TMs and } L(M_1) = L(M_2)\}$
- $\overline{A_{TM}}$ = The complement of $\{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ accepts } w\}$
- $REG_{TM} = \{\langle M\rangle | M \text{ is a TM and } L(M) \text{ is regular}\}$

- $ALL_{TM} = \{\langle M \rangle | M \text{ is a TM and } L(M) = \Sigma^*\}$
- $\overline{ALL_{TM}}$