COMS 3261: Fall 2024
Kiru Arjunan, Hellen Zhao

Handout 10a: Unrecognizability and Mapping Reductions

1 Unrecognizability

Definition 1.1. A language L is **recognizable** if and only if there exists a Turing Machine that accepts every string in L, and does not accept (rejects or loops) any string not in L.

Definition 1.2. A language L is **co-recognizable** if and only if its complement $\overline{L}$ is recognizable.

Theorem 1.3. *L is decidable $\iff$ L is recognizable and co-recognizable*

We've seen a few different ways to prove a language L is not recognizable:

Ways to show L is not recognizable:

1. Show that L is not decidable and $\overline{L}$ is recognizable.
2. Show that $\overline{L}$ is recognizable but undecidable.
3. Use a mapping reduction from another language that is not recognizable.
4. (Not part of required material) Using refined Rice's theorem

The first two methods follow from Theorem 1.3 (proved in lecture 14). If L is undecidable, then L is unrecognizable or $\overline{L}$ is unrecognizable. If $\overline{L}$ is recognizable, it must be that L is unrecognizable. Recall that L is decidable iff $\overline{L}$ is decidable (lecture 13), so showing L is undecidable is equivalent to showing $\overline{L}$ is undecidable. Methods 1 and 2 are essentially the same thing.

Example 1. $\overline{A_{TM}}$ is unrecognizable because its complement A_{TM} is recognizable (recognizer given in lecture 15) but undecidable (proof by diagonalization in lecture 16).

Using the complement and undecidability is not always the easiest approach though, and in doesn't apply to cases where both L and $\overline{L}$ are unrecognizable. Another method to prove L is unrecognizable is through mapping reductions. We'll give a brief summary of the key ideas below, but you can find more details in lectures 19 and 20. We also briefly cover refined Rice's theorem, which is another way to prove unrecognizability (similar to the way we used Rice's theorem for proving undecidability). Refined Rice is not part of required material for class, but you may use it if you'd like - you can find more details in lecture 18 lecture notes.

2 Mapping Reductions

Definition 2.1. A **computable** function is a function $f : \Sigma^* \to \Sigma^*$ that can be computed by a Turing Machine. That is there exists a Turing Machine that, on every input w, halts with just $f(w)$ on its tape.

Definition 2.2. A language A is **mapping reducible** to a language B, denoted $A \leq_m B$, if there exists a computable function $f : \Sigma^* \to \Sigma^*$ such that, for every $w \in \Sigma^*$,

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

Theorem 2.3. *If $A \leq_m B$, then $A \leq_T B$. (Proof in lecture 19)*

Theorem 2.4. *Suppose $A \leq_m B$ for two languages A, B.*
1. If B is recognizable, then A is recognizable.
2. If A is not recognizable, then B is not recognizable.

These two statements are actually saying the same thing (the second is a contrapositive of the first). The second statement gives us a strategy to prove that L is unrecognizable. If we can find an unrecognizable language A that is mapping-reducible to L (that is, $A \leq_m L$), then L is unrecognizable.

Theorem 2.5. $A \leq_m B \iff \overline{A} \leq_m \overline{B}$.

So given a language A that we already know to be unrecognizable, we could prove the unrecognizability of L either by showing $A \leq_m L$, or by showing $\overline{A} \leq_m \overline{L}$. This can be helpful, since sometimes it is easier to work with the complement of a language rather than the language itself. For example, working with $A_{TM} = \{\langle M, w\rangle \mid \text{M is a TM that accepts the string w}\}$ may be easier than working with $\overline{A_{TM}}$, which includes both strings encoded as $\langle M, w\rangle$ and invalid encodings.

Steps to proving L is unrecognizable by mapping reduction

1. Choose an unrecognizable language U that is mapping reducible to L. You either want to show $U \leq_m L$ or $\overline{U} \leq_m \overline{L}$.

2. Decide what mapping reduction $A \leq_m B$ and computable function f to show.

2. Prove $A \leq_m B$.

   2a. Construct a TM that computes f.

   2b. Make sure your function f is indeed computable.

   2b. Show that f is correct, meaning $w \in A \iff f(w) \in B$. This involves showing two directions:
   (i) $w \in A \implies f(w) \in B$
   (ii) $w \notin A \implies f(w) \notin B$ [equivalent to $f(w) \in B \implies w \in A$]

3. Conclude that your mapping reduction implies L is unrecognizable (think about how it is justified by theorems seen in class).

Some remarks about the steps above:

- Make sure your function f is *actually* computable! While you're only expected to give a high level description of a TM for f, you should be convinced that you could provide a more detailed implementation if needed. Saying something like "if M never halts on w, have f(w) be..." is not implementable by a TM!

- Note that there are multiple ways to show $w \in A \iff f(w) \in B$. If a statement is true, so is its contrapositive, so

  (i) Showing $w \in A \implies f(w) \in B$ is equivalent to showing $f(w) \notin B \implies w \notin A$.

  (ii) Showing $f(w) \in B \implies w \in A$ is equivalent to showing $w \notin A \implies f(w) \notin B$.

  Oftentimes it's easier to show (i) directly and (ii) by contrapositive.

3 Refined Rice's Theorem (Bonus material, not required)

Recall that one of the ways we had to prove undecidability is using Rice's theorem, which can be applied whenever the language is a non-trivial property of recognizable languages (basically, any langauge of the form $\{\langle M \rangle \mid M \text{ is a TM and } L(M) \text{ satisfies...}\}$). See lecture notes and Handout 9 for a review and examples.

We know that if a language L is undecidable, it must be either not recognizable or not co-recognizable (or both) – this follows from the Theorem 1.3 as discussed above. Refined Rice's theorem applies to exactly the same set of languages as Rice's theorem, but further lets us conclude that L is not recognizable or not co-recognizable, depending on whether the TM recognizing the empty language is in L or in $\overline{L}$

Theorem 3.1. *Refined Rice Theorem: Let P be a non-trivial property of recognizable languages. Let $M_\emptyset$ be a TM such that $L(M_\emptyset) = \emptyset$. Then:*

- *if $\langle M_\emptyset \rangle \in P$, then P is not recognizable*
- *If $\langle M_\emptyset \rangle \notin P$ then P is not co-recognizable (namely, $\overline{P}$ is not decidable).*

Steps to proving L is unrecognizable by Refined Rice's Theorem

1. Prove that L is a language property:

   1a. Check that L consists of strings of the form $\langle M \rangle$ where M is a TM.

   1b. Check that for any two TMs M_1, M_2 such that $L(M_1) = L(M_2)$ it holds that $M_1 \in L \iff M_2 \in L$

2. Prove that L is non-trivial:

   2a. Show that there exists a TM M such that $\langle M \rangle \in L$

   2b. Show that there exists a TM M' such that $\langle M' \rangle \notin L$.

3. Show that $\langle M_\emptyset \rangle \in L$

4. Conclude that L is not recognizable.

Note: if instead in step 3 you show $\langle M_\emptyset \rangle \notin L$, you can conclude that L is not co-recognizable.

For example, Refined Rice's theorem could be used in order to prove E_{TM}, REG_{TM}, and $\overline{ALL_{TM}}$ are all not recognizable (since each of these can be shown to be a non-trivial property of recognizable languages, and $M_\emptyset$ satisfies each of these properties.

4 Example Unrecognizable Languages

Proven in class

- $\overline{A_{TM}}$ = The complement of $\{\langle M, w\rangle \mid M$ is a TM and M accepts $w\}$ (lecture 19)
- $E_{TM} = \{\langle M\rangle \mid M$ is a TM and $L(M) = \emptyset\}$ (lecture 19)
- $ALL_{TM} = \{\langle M\rangle | M$ is a TM and $L(M) = \Sigma^*\}$ (lecture 20)
- $\overline{ALL_{TM}}$ (lecture 19)

Unproven

- $\overline{HALT_{TM}}$ = The complement of $\{\langle M, w\rangle | M$ is a TM and M halts on $w\}$
- $EQ_{TM} = \{\langle M_1, M_2\rangle \mid M_1$ and M_2 are TMs and $L(M_1) = L(M_2)\}$
- $REG_{TM} = \{\langle M\rangle | M$ is a TM and $L(M)$ is regular$\}$

5 Practice Problems

1. Prove that $L = \{\langle M, D\rangle | M$ is a TM, D is a DFA, and $L(M) = L(D)\}$ is not co-recognizable. That is, prove that $\overline{L}$ is not recognizable.
2. Prove that $L = \{\langle M\rangle | M$ is a TM which does not accept strings of length $\geq 50\}$ is not recognizable.
3. Prove that if L_1 and L_2 are recognizable, then $L_3 = L_1 \cdot L_2$ is also recognizable. That is, prove that recognizable languages are closed under concatenation.
4. Let A be a language. Prove that $A \leq_m A$.
5. Is it necessarily true that $A \leq_m \overline{A}$?