COMS 3261: Fall 2024
Kiru Arjunan, Hellen Zhao

Handout 10b: Unrecognizability and Mapping Reductions - Solutions

Problem 1

Prove that $L = \{\langle M, D\rangle \mid M \text{ is a TM}, D \text{ is a DFA, and } L(M) = L(D)\}$ is not co-recognizable. That is, prove that $\overline{L}$ is not recognizable.

Answer:
Since $\overline{A_{TM}}$ is unrecognizable, it would suffice to prove $\overline{A_{TM}} \leq_m \overline{L}$. Note this is equivalent to showing $A_{TM} \leq_m L$, which is what we will show. Consider the computable function f as follows:

Algorithm 1 Function f

Input: $\langle M, w\rangle$ where M is a TM and w is a string

---

1: Build a TM M' as follows:

- M': "On input x
  - If $x \neq w$, reject x.
  - If $x = w$
  - Run M on w.
  - If M accepted w, accept x.
  - Otherwise reject x."

2: Build a DFA D such that $L(D) = L(w) = \{w\}$.
3: Return $\langle M', D\rangle$.

This f is computable, since every step is implementable. In particular, constructing $\langle M'\rangle$ is easy given $\langle M, w\rangle$, and constructing a DFA that accepts a single string is also implementable (we saw in the first part of class how to do that).

Furthermore, $\langle M, w\rangle \in A_{TM} \iff f(\langle M, w\rangle \in L$ because:

1. If $\langle M, w\rangle \in A_{TM}$, then M' accepts w and rejects all other strings, so $L(M') = \{w\} = L(D)$ and so $\langle M', D\rangle \in L$.
2. If $\langle M, w\rangle \notin A_{TM}$, then M' does not accept any string, so $L(M') = \emptyset \neq L(D)$ and so $\langle M', D\rangle \notin L$.

Thus, we have a computable function f such that $\langle M, w\rangle \in A_{TM} \iff f(\langle M, w\rangle) \in L$, and so $A_{TM} \leq_m L$. We conclude that $\overline{L}$ is unrecognizable.

Remark. Note that in our function, we did not include logic to deal with improperly formed input strings. To implement this, we could pick a string not in L, then explicitly tell our TM to map all invalid inputs to that string. In general, when we give a TM that reduces A to B, we can assume that our TM will map improperly formed inputs to strings not in B without saying so.

Problem 2

Prove that $L = \{\langle M \rangle \mid M \text{ is a TM which does not accept strings of length } \geq 50\}$ is not recognizable.

Answer:

Proof using mapping reduction. We will show that $E_{TM} \leq_m L$. Consider the computable function f defined as follows:

Algorithm 2 Function f

Input: $\langle M \rangle$ where M is a TM

1: Build a TM M' as follows:

- M': "On input x
  - If $|x| < 50$, reject x.
  - If $|x| \geq 50$
  - Let x' be x without the first 50 characters.
  - Run M on x' and output the same."

2: Return $\langle M' \rangle$.

This f is computable, since every step is implementable.

If $\langle M \rangle \in E_{TM}$, M will never accept any string as $L(M) = \emptyset$. But the only time M' accepts a string is if M accepts a (different) string. Thus, M' will never accept any string, and so will not accept any string of length ≥ 50. Thus, $f(\langle M \rangle) = \langle M' \rangle \in L$.

If $\langle M \rangle \notin E_{TM}$, then there exists a w such that M accepts w. Let $a \in \Sigma$. Note that M' will accept $a^{50}w$. Thus, since $|a^{50}w| \geq 50$, $f(\langle M \rangle) = \langle M' \rangle \notin L$.

Therefore, $w \in E_{TM} \iff f(w) \in L$, and so $E_{TM} \leq_m L$. Therefore, since E_{TM} is not recognizable, neither is L.

Remark. Note that typically there are many natural reductions (and this applies to all the exercises here). For example, for the proof we gave above, it really doesn't matter what M' does on inputs of length shorter than 50. The crucial point was to create M' such that $\langle M \rangle \in E_{TM}$ if and only if $\langle M' \rangle \in L$. We could also choose a different language to reduce from.

Proof using refined Rice's theorem. We show that L satisfies all the properties required for the refined Rice's theorem. First, we show this L a property of recognizable languages. This follows directly from how L is defined: $L \subseteq \{\langle M \rangle \mid M \text{ is a TM}\}$, and if we have two TMs M_1, M_2 such that $L(M_1) = L(M_2)$, then if either of them accepts any string of length ≥ 50 then the other one does too. That is, $\langle M_1 \rangle \in L \iff \langle M_2 \rangle \in L$.

Next, we show L is non-trivial. Let $M_\emptyset$ be the TM that immediately rejects every input. In particular, it does not accept any input of length ≥ 50, so $\langle M_\emptyset \rangle \in L$. Now let M_{Σ^*} be the TM that immediately accepts every input. $\langle M_{\Sigma^*} \rangle \notin L$.

Finally, as already noted above, $\langle M_\emptyset \rangle \in L$. Thus, by Refined Rice's theorem we can conclude that L is not recognizable.

Problem 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.

Answer:
Let M_1 and M_2 be Turing machines that recognizes L_1 and L_2 respectively. We will construct a non-deterministic TM[1] M_3 that recognizes $L_3 = L_1 \cdot L_2$.

Algorithm 3 A non-deterministic recognizer M_3 for L_3

Input: w

1: Non-deterministically split w into w_1 and w_2 such that $w = w_1 w_2$
2: Run M_1 on w_1
3: **if** M_1 accepts **then**
4: Run M_2 on w_2
5: **if** M_2 accepts **then**
6: Accept w
7: **end if**
8: **end if**

We now prove that the above is a recognizer for L_3. M_3 accepts w if and only if $w \in L_3$.

If $w \in L_3$ and $L_3 = L_1 \cdot L_2$, then there is a parsing $w = xy$ where $x \in L_1$ and $y \in L_2$. So there is a branch of computation where M_3 picks $w_1 = x$ and $w_2 = y$ on line 1. Since $x \in L_1$, M_1 will halt and accept x as it is a recognizer for L_1. Similarly M_2 will halt and accept y on line 5. Hence on this branch of computation, algorithm will reach line 6 and M_3 will accept input w.

If $w \notin L_3$, then no matter what split $w = w_1 w_2$ we choose, $w_1 \notin L_1$ or $w_2 \notin L_2$. If $w_1 \notin L_1$, then M_1 will reject or run forever on line 2. If $w_1 \in L_1$ but $w_2 \notin L_2$, then M_2 will reject or run forever on line 4. Either ways M_3 will not accept and will run forever.

Remark. In fact, recognizable languages are closed under union, star and intersection too! Try to prove these as an exercise. However, recognizable languages are NOT closed under complement. We can easily provide A_{TM} as a counter example. We know that A_{TM} is recognizable but $\overline{A_{TM}}$ is not recognizable.

Problem 4

Let A be a language. Prove that $A \leq_m A$.

Answer:
Let f be the identity function. That is, for all strings w, let $f(w) = w$. This is clearly computable. We have $w \in A \iff w = f(w) \in A$. Thus, $A \leq_m A$ by definition.

[1] We could also construct a deterministic TM which checks all parsings of the input into a concatenation of two strings, but we need to use dovetailing to make sure we don't get in an infinite run on one parsing when there's another parsing that works. The non-deterministic solution we give is simpler.

Problem 5

Is it necessarily true that $A \leq_m \overline{A}$?

Answer:
No. If it were always true that $A \leq_m \overline{A}$ and $\overline{A} \leq_m A$, then this would imply that all recognizable languages are co-recognizable, which would then imply that all recognizable languages are decidable! But we know this to be false. One example is A_{TM}, as mentioned in problem 3's solution.
Remark. Note that for Turing-reductions, it IS true that for every A we have $A \leq_T \overline{A}$ (since $A \leq_T A$ is equivalent to $A \leq_T \overline{A}$).