Handout 9b: Solutions to Exercises (Reductions, Undecidability, Unrecognizability)

Ananya Gandhi and Nicolas Hortiguera
Credit to Eli Goldin and Alan Du (Fall 2020 TAs)

COMS 3261 Fall 2022

1 Countability

(No exercises)

2 Turing Reductions and Undecidability

1. Prove that $HALT_{TM} \leq_T A_{TM}$.

   Answer:
   Suppose that there were a decider $\mathcal{O}$ for A_{TM}. We will construct a decider R for $HALT_{TM}$ using $\mathcal{O}$ as follows:

   R: -On input $\langle M, w\rangle$
   -Run $\mathcal{O}$ on $\langle M, w\rangle$. If $\mathcal{O}$ accepts, accept.
   -Create an encoding of a new TM $\langle M'\rangle$ as follows:
   M': "-On input x
   -Run M on x
   -If M accepts, reject. If M rejects, accept.
   "
   -Run $\mathcal{O}$ on $\langle M', w\rangle$. If $\mathcal{O}$ accepts, accept.
   -Reject.

   If $\langle M, w\rangle \in HALT_{TM}$, then either M accepts w or M rejects w. In the former case, $\mathcal{O}$ accepts $\langle M, w\rangle$. In the latter case, M' accepts w and so $\mathcal{O}$ accepts $\langle M', w\rangle$. Either way, R accepts $\langle M, w\rangle$.
   If $\langle M, w\rangle \notin HALT_{TM}$, then M runs forever on w. Thus, M' also runs forever on w. Therefore, $\langle M, w\rangle \notin A_{TM}$ and $\langle M', w\rangle \notin A_{TM}$ and so $\mathcal{O}$ rejects both cases. Thus, R rejects $\langle M, w\rangle$.

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

   Answer:
   We will prove this by showing that $A_{TM} \leq_T L$. Suppose that there were a decider $\mathcal{O}$ for L. We will

use $\mathcal{O}$ to construct a decider R for A_{TM} as follows:

R: -On input $\langle M, w\rangle$
-Create an encoding of a new TM $\langle M'\rangle$ (or we could say $\langle M'_w\rangle$) as follows:
M': "-On input x
-If $x \neq w$ reject.
-If $x = w$, run M on w. If M accepts, accept. Otherwise, reject.
"
-Create an encoding of a new DFA $\langle D\rangle$ such that $L(D) = L(w) = \{w\}$ (this is ok as we know an algorithm to construct DFAs from regular expressions).
-Run $\mathcal{O}$ on $\langle M', D\rangle$ and output same.

If $\langle M, w\rangle \in A_{TM}$, then M accepts w. Thus, M' accepts w and rejects everything else, so $L(M') = \{w\}$. Therefore, $L(M') = L(D)$, and so $\mathcal{O}$ accepts $\langle M', D\rangle$. Thus, R accepts $\langle M, w\rangle$.
If $\langle M, w\rangle \notin A_{TM}$, then M does not accept w. Thus, $L(M') = \emptyset$. Therefore, $L(M') \neq L(D)$ since $L(D) = \{w\}$. Therefore, $\mathcal{O}$ rejects $\langle M', D\rangle$ and so R rejects x.

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) $A \leq_T \overline{B}$

Answer:
1)⇒ 2): Let $A \leq_T B$. Thus, if there exists a decider $\mathcal{O}$ for B, we can create a decider R for A. Let R' run R and return the opposite. R' is a decider for $\overline{A}$ using $\mathcal{O}$. Thus, $\overline{A} \leq_T B$.
2)⇒ 3): Let $\overline{A} \leq_T B$. If there were a decider $\mathcal{O}$ for $\overline{B}$, then we could create a decider $\mathcal{O}'$ for B by running $\mathcal{O}$ and returning the opposite. But since $\overline{A} \leq_T B$, we could use $\mathcal{O}'$ to create a decider for $\overline{A}$. Thus, $\overline{A} \leq_T \overline{B}$.
3)⇒ 4): Let $\overline{A} \leq_T \overline{B}$. Thus, if there exists a decider $\mathcal{O}$ for $\overline{B}$, we can create a decider R for $\overline{A}$. Let R' run R and return the opposite. R' is a decider for $A = \overline{\overline{A}}$ using $\mathcal{O}$. Thus, $\overline{A} \leq_T B$.
4)⇒ 1): Let $A \leq_T \overline{B}$. If there were a decider $\mathcal{O}$ for B, then we could create a decider $\mathcal{O}'$ for $\overline{B}$ by running $\mathcal{O}$ and returning the opposite. But since $A \leq_T \overline{B}$, we could use $\mathcal{O}'$ to create a decider for A. Thus, $A \leq_T B$.

3 Using Rice's Theorem to prove undecidability

1. Does Rice's theorem apply to $L = \{\langle M\rangle \,|M \text{ is a TM and } M \text{ accepts } 0\}$?

Answer: Yes.
Clearly $L \subseteq \{\langle M\rangle \,|M \text{ is a TM}\}$. If M_1, M_2 are TMs and $L(M_1) = L(M_2)$, then M_1 accepts 0 $\iff$ M_2 accepts 0. Thus, $\langle M_1\rangle \in L \iff \langle M_2\rangle \in L$.
Now, take M accepting all strings, M' rejecting all strings. $M \in L$, $M' \notin L$. Thus, $L \neq \emptyset$ and $L \neq \{\langle M\rangle \,|M \text{ is a TM}\}$.
Therefore, L is undecidable.

2. Does Rice's theorem apply to $L = \{\langle M\rangle \,|M \text{ is a TM and } M \text{ has exactly two states}\}$?

Answer: No.
L is not a property of recognizable languages. Consider any TM M with two states. We can always add useless states which can not be reached to create M' with the same language. Thus, $L(M) = L(M')$ and $\langle M\rangle \in L$ while $\langle M'\rangle \notin L$.
In fact, L is decidable. We could create a Turing machine which simply counts the number of states and accepts if there are two, and rejects otherwise.

3. Does Rice's theorem apply to $L = \{\langle M \rangle \, | M \text{ is a TM and } M \text{ rejects } 0\}$?

   Answer: No.
   L is not a property of recognizable languages. Consider M_1 a TM which rejects all strings, M_2 a TM which runs forever on all strings. $L(M_1) = L(M_2) = \emptyset$. M_1 rejects 0, so $\langle M_1 \rangle \in L$. However, M_2 runs forever on 0, and specifically does not reject 0. Thus, $\langle M_2 \rangle \notin L$.

   Despite the fact that Rice's theorem does not apply, L is undecidable. We can prove this e.g. by a reduction from the language in 3.1.

4. Does Rice's theorem apply to $E_{TM} = \{\langle M \rangle \, | M \text{ is a TM and } L(M) = \emptyset\}$?

   Answer: Yes.
   Clearly $E_{TM} \subseteq \{\langle M \rangle \, | M \text{ is a TM}\}$. If M_1, M_2 are TMs and $L(M_1) = L(M_2)$, then $L(M_1) = \emptyset \iff L(M_2) = \emptyset$. Thus, $\langle M_1 \rangle \in E_{TM} \iff \langle M_2 \rangle \in E_{TM}$.
   Now, take M accepting all strings, M' rejecting all strings. We have $L(M) = \emptyset$, $L(M') = \Sigma^*$. $M \in E_{TM}$, $M' \notin E_{TM}$. Thus, $E_{TM} \neq \emptyset$ and $E_{TM} \neq \{\langle M \rangle \, | M \text{ is a TM}\}$.
   Therefore, E_{TM} is undecidable.

5. Does Rice's theorem apply to $L = \{\langle M \rangle \, | M \text{ is a TM and } L(M) = \overline{A_{TM}}\}$?

   Answer: No.
   Here, we have that L is indeed a property of recognizable languages. However, L is trivial. We know that $\overline{A_{TM}}$ is unrecognizable, and so there exists no TM M such that $L(M) = \overline{A_{TM}}$. Therefore, $L = \emptyset$. Note that as $\emptyset$ is a decidable language, so is L. (For a decider, consider the TM: "on input x, reject.").

6. Does Rice's theorem apply to $L = \{\langle M \rangle \, | M \text{ is a TM and } L(M) \text{ is recognizable}\}$?

   Answer: No.
   Note that for every TM M, by definition $L(M)$ is recognizable. Thus, $L = \{\langle M \rangle \, | M \text{ is a TM}\}$ and so L is trivial.
   Note that $\{\langle M \rangle \, | M \text{ is a TM}\}$ is a decidable language, and so L is as well. (For a decider, consider the TM: "on input $\langle M \rangle$ where M is a TM, accept.")

7. Does Rice's theorem apply to $L = \{\langle M \rangle \, | M \text{ is a TM and } L(M) \text{ is decidable}\}$?

   Answer: Yes.
   Clearly $L \subseteq \{\langle M \rangle \, | M \text{ is a TM}\}$. If M_1, M_2 are TMs and $L(M_1) = L(M_2)$, then $L(M_1)$ is decidable $\iff L(M_2)$ is decidable. Thus, $\langle M_1 \rangle \in L \iff \langle M_2 \rangle \in L$.
   Let M reject all strings, and let U be a recognizer for A_{TM}. We know that M is a decider (and $L(\langle M \rangle) = \emptyset$) is a decidable language) , and so $\langle M \rangle \in L$. However, $L(U) = A_{TM}$ is not decidable, and so $\langle U \rangle \notin L$. Thus, L is non-trivial.

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:
$INFINITE_{TM} = \{\langle M \rangle \, | M \text{ is a TM and } L(M) \text{ is an infinite language}\}$

Solution: $INFINITE_{TM}$ is a language of TM descriptions. It satisfies the conditions of Rice's theorem. First, it depends only on the language: if two TMs M_1, M_2 recognize the same language, either both have descriptions in $INFINITE_{TM}$ or neither do. Second, it is nontrivial because some TMs have infinite languages and others do not. For a specific example, take M a TM that accepts all inputs, and M' a TM that rejects all inputs, then $\langle M \rangle \in INVINITE_{TM}$ while $\langle M' \rangle \notin INVINITE_{TM}$. Thus, $INFINITE_{TM}$ is a non-trivial property of recognizable languages, and so Rice's theorem implies that it is undecidable.

4 Proving L is unrecognizable - Overview

(No exercises)

5 Using complements and undecidability to prove unrecognizability

(No exercises)

6 Mapping Reductions for unrecognizability

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.

   Answer:
   Note that the Turing-reduction given in the solution for 2.2 is actually a mapping reduction! Thus, $A_{TM} \leq_m L$, and so $\overline{A_{TM}} \leq \overline{L}$. Therefore, $\overline{L}$ is not recognizable. To see this more formally, consider the computable function f as follows:

   f: -On input $\langle M, w\rangle$
   -Create an encoding of a new TM $\langle M'\rangle$ (or we could say $\langle M'_w\rangle$) as follows:
   M': "-On input x
   -If $x \neq w$ reject.
   -If $x = w$, run M on w. If M accepts, accept. Otherwise, reject.
   "
   -Create an encoding of a new DFA $\langle D\rangle$ such that $L(D) = L(w) = \{w\}$ (this is ok as we know an algorithm to construct DFAs from regular expressions).
   -Return $\langle M', D\rangle$.

   This f is computable, since every step is implementable.
   If $\langle M, w\rangle \in A_{TM}$, then $L(M') = L(D)$ and so $\langle M', D\rangle \in L$.
   If $\langle M, w\rangle \notin A_{TM}$, then $L(M') = \emptyset \neq L(D)$ and so $\langle M', D\rangle \notin L$.
   Thus, $\langle M, w\rangle \in A_{TM} \iff f(\langle M, w\rangle) \in L$, and so $A_{TM} \leq_m L$.

2. Prove that $L = \{\langle M\rangle \,|M \text{ does not accept strings of length } \geq 50\}$ is not recognizable.

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

   f: -On input $\langle M\rangle$.
   -Create an encoding of a new TM $\langle M'\rangle$ as follows:
   M': "-On input w"
   -If $|w| < 50$, reject.
   -If $|w| \geq 50$, let w' be w without the first 50 characters. Run M on w' and output the same.
   -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, $f(M) = \langle M' \rangle \in L$.

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

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.

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

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

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

   Answer: No. Consider A_{TM}. We know that A_{TM} is recognizable, while $\overline{A_{TM}}$ is not. Thus, we cannot possibly have $\overline{A_{TM}} \leq_m A_{TM} = \overline{\overline{A_{TM}}}$.

   Note that for Turing-reductions, it IS true that for every A we have $A \leq_T olA$, as follows from exercise 2.3.