Handout 9b - Countability, Turing Reductions, Proving Undecidability

Owen Keith and Owen Terry

COMS 3261 Fall 2024

1 Countability

No exercises.

2 Turing Reductions and Undecidability

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

Proof. Note, here and in all later TM descriptions it is implied that the input should correspond to a valid encoding, and the TM will reject otherwise. For this problem, that means we are omitting the implied first step of "If $\langle M, w \rangle$ is an invalid encoding, reject."

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.
  Prepare $\langle M' \rangle$ where
    $M' =$ "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 $\langle M, w \rangle$. □

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

Proof. 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 $\langle M' \rangle$ 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 okay because 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. □

Exercise 3. Prove that the following are equivalent: $A \leq_T B, \overline{A} \leq_T B, A \leq_T \overline{B}, \overline{A} \leq_T \overline{B}$.

Proof. $1 \implies 2$: Let $A \leq_T B$. Thus, if there exists a decider $\mathcal{O}$ for B, we can create a decider R for A. On input x, let R' run R on x and return the opposite. R' is a decider for $\overline{A}$ using $\mathcal{O}$ since $x \in \overline{A} \iff R$ rejects $\iff R'$ accepts, and $x \notin \overline{A} \iff R$ accepts $\iff R'$ rejects. Thus, $\overline{A} \leq_T B$.
$2 \implies 3$: Let $\overline{A} \leq_T B$. Thus, if there exists a decider for B, then there exists a decider for $\overline{A}$. Let's say we have a decider $\mathcal{O}'$ for $\overline{B}$, then we must have a decider $\mathcal{O}$ for B, i.e., on input x, $\mathcal{O}$ runs $\mathcal{O}'$ on x and returns the opposite. Now, since we have a decider for B and since we assumed $\overline{A} \leq_T B$, we must have a decider R' for $\overline{A}$. So, we must have a decider R for A, i.e., on input x, R runs R' on x and returns the opposite value.
$3 \implies 4$: 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 using $\mathcal{O}$. Thus, $A \leq_T \overline{B}$.
$4 \implies 1$: 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

Does Rice's theorem apply to the following languages? If it does not, prove whether or not the language is decidable. To apply Rice's theorem to show a language P is undecidable, P must meet the following criteria

1. $P \subset \{\langle M \rangle | M \text{ is a TM}\}$ (strict subset)
2. P is nontrivial, i.e., $P \neq \emptyset$ and $P \neq \{\langle M \rangle | M \text{ is a TM}\}$
3. P is a property of the TM's language, i.e., whenever $L(M_1) = L(M_2)$ we have $\langle M_1 \rangle \in P$ if and only if $\langle M_2 \rangle \in P$.

1. $L = \{\langle M \rangle \mid M \text{ is a TM and } M \text{ accepts } 0\}$

   Yes.

   Clearly $L \subset \{\langle M \rangle \mid M \text{ is a TM}\}$. 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 $\langle M \rangle$ accepting all strings, $\langle M' \rangle$ rejecting all strings. $\langle M \rangle \in L$, $\langle M' \rangle \notin L$. Thus, $L \neq \emptyset$ and $L \neq \{\langle M \rangle \mid M \text{ is a TM}\}$. Therefore, L is undecidable.

2. $L = \{\langle M \rangle \mid M \text{ is a TM and } M \text{ has exactly two states}\}$

   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. $L = \{\langle M \rangle \mid M \text{ is a TM and } M \text{ rejects } 0\}$

   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, $\langle M_2 \rangle \in L$.

   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 (proof omitted).

4. $E_{TM} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) = \emptyset\}$

   Yes.

   Clearly $E_{TM} \subset \{\langle M \rangle \mid M \text{ is a TM}\}$. If M_1, M_2 are TMs and $L(M1) = L(M2)$, 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 $E_{TM} \neq \{\langle M \rangle \mid M \text{ is a TM}\}$. Therefore, E_{TM} is undecidable.

5. $L = \{\langle M \rangle \mid M \text{ is a TM and } L(M) = \overline{A_{TM}}\}$

   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. $L = \{\langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is recognizable}\}$

   No.

For every TM M, by definition $L(M)$ is recognizable. Thus, $L = \{\langle M \rangle \mid M \text{ is a TM}\}$ and so L is trivial.

Note that $\{\langle M \rangle \mid M \text{ is a TM}\}$ 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. $L = \{\langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is decidable}\}$

   Yes.

   Clearly $L \subset \{\langle M \rangle \mid M \text{ is a TM}\}$. 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.