CSC4010: Introduction to Theoretical Computer Science
Problem Set 4

A construction template for unrecognizability proofs

To prove that a language L is unrecognizable, you may construct a mapping reduction $\overline{A_{\text{TM}}} \leq_m L$. That is, define a computable function f satisfying $\langle M, w\rangle \in \overline{A_{\text{TM}}} \iff f(\langle M, w\rangle) \in L$. While the construction of f may seem mysterious, it often follows a template which describe below. Here F refers to the Turing Machine which computes f.

Machine F: On input $\langle M, w\rangle$:

1. Construct a Turing machine N as follows:

   Machine N: On input x:
   (a) **Part 1:** [Initial behavior to be specified.]
   (b) Simulate M on w.
   - If M accepts, proceed to part 2.
   - If M rejects, loop forever.
   - If M loops, continue simulating forever.

   (c) **Part 2:** [Remaining behavior to be specified.]
2. Return $f(\langle M, w\rangle) = \langle N\rangle$.

How to use the template:

- **Part 1** runs before the simulation, regardless of what M does on w. It may accept or reject immediately on selected inputs.
- **Part 2** is reached only if Part 1 continues and M accepts w.
- F **constructs a machine description; it does not run** N**.** Thus, F must halt even when N does not.

Choose Parts 1 and 2 so that:

- If M **does not accept** w (either rejecting or looping), the output belongs to L.
- If M **accepts** w, the output does not belong to L.

Your proof must describe the construction, explain why f is total and computable, establish both cases, and explain why the reduction implies that L is unrecognizable. This template is only a guide. It may not always apply. For example, if the target language concerns multiple machines or a machine-input pair, adapt the construction and its output accordingly. You are also welcome to use other proofs even if it does.

1 Homework Proper

1. Prove the following two statements:
   (a) Show that a language A is recognizable if and only if $A \leq_m A_{TM}$.
   (b) Show that a language A is decidable if and only if $A \leq_m L_{0^*1^*}$, where $L_{0^*1^*} = \{0^m 1^n \mid m, n \geq 0\}$.

2. A **useless** state in a Turing machine is one that is never entered on any input string. Consider the problem of determining whether a Turing machine has any useless states. Formulate this problem as a language and show that it is unrecognizable.

3. Let $L_{\text{finite}} = \{\langle M \rangle \mid M \text{ is a Turing Machine and } L(M) \text{ is finite}\}$, then L_{finite} is unrecognizable.

4. Given two Turing machines M_1 and M_2, we define that M_1 *lags* behind M_2 if the following condition is satisfied: For all input strings $x \in \{0,1\}^*$, if $M_2(x)$ accepts, then $M_1(x)$ loops. (If $M_2(x)$ does not accept, $M_1(x)$ may behave arbitrarily.) Define the language

$$L_{\text{LAG}} = \{\langle M_1 \rangle, \langle M_2 \rangle : M_1 \text{ lags behind } M_2\}.$$

   Prove that L_{LAG} is unrecognizable.

Solutions

1. **(a) ($\Rightarrow$)** Suppose A is recognizable, and fix a TM M with $L(M) = A$. Define $f(x) = \langle M, x\rangle$. The function f is computable, since M is fixed and f appends x to $\langle M\rangle$. Moreover, $x \in A \iff M$ accepts $x \iff \langle M, x\rangle \in A_{TM}$.

   (b) ($\Rightarrow$) Suppose A is decidable, and let D be a decider for A. Define f as follows: on input x, run D on x; if D accepts, output 01; if D rejects, output 10. Since D halts on every input, f is a computable function. By construction, $x \in A \iff f(x) \in L_{0^*1^*}$, i.e. $A \leq_m L_{0^*1^*}$.

2. Let $L_U = \{\langle M\rangle$: there is a state of M that is not reached on any input$\}$. We reduce the problem $\overline{A}_{TM}$ to L_U. Let $\langle M, w\rangle$ be an instance of $\bar{A}_{TM}$. Let M_w be the Turing machine which on input x does the following: if $x \neq w$, then it visits all its states except the "accept" state and rejects. If $x = w$, then it simulates M on w. Note that $\langle M_w\rangle \in L_U \iff \langle M, w\rangle \in \bar{A}_{TM}$. Thus, the function $f(\langle M, w\rangle) = \langle M_w\rangle$.

3. Define f by $f(\langle M, w\rangle) = \langle M_w\rangle$, where M_w is the following machine:

   $M_w =$ "On input x:

   1. Simulate M on w for $|x|$ steps.
   2. If M accepts w within those $|x|$ steps, *accept* x; otherwise *reject* x."

   Note that f is a computable function.

   1. **Case 1: M does not accept w.** Then M never accepts w, so M_w rejects *every* input x. Hence $L(M_w) = \emptyset$, which is finite, and $\langle M_w\rangle \in L_{\text{finite}}$.
   2. **Case 2: M accepts w.** Suppose M accepts w after exactly t steps. Then for every string x with $|x| \geq t$, so M_w accepts x. There are infinitely many strings of length $\geq t$, so $L(M_w)$ is infinite and $\langle M_w\rangle \notin L_{\text{finite}}$.

   Combining the two cases, $\langle M, w\rangle \in \bar{A}_{\text{TM}} \iff \langle M_w\rangle \in L_{\text{finite}}$, so $\overline{A}_{\text{TM}} \leq_m L_{\text{finite}}$.

4. We define M_2 as a machine that accepts any input x. On the other hand, define M_w as follows: on input x, first run M on w. If it accepts, then M_w accepts x. On the other hand M rejects then w, then M_w deliberately loops on x. Note that M_2 accepts all strings, and M_w lags behind M_2 if and only if M does not accept w. That is $\langle M_1\rangle, \langle M_2\rangle \in L_{\text{LAG}} \iff \langle M, w\rangle \in \bar{A}_{TM}$. Thus, we define $f(\langle M, w\rangle) = \langle M_w\rangle, \langle M_2\rangle$. This is a computable function.