Theory of Computation Homework 2

350510 Kim Yujun

March 2022

Problem 1

$$L_1 = \{< D, D' >: D \text{ and } D' are \text{ DFA's and } L(D) \subseteq L(D')\}$$
$$L_2 = \{< M >: M \text{ is a TM such that } L(M) \text{ is infinite}\}$$

(b) I claim that L_2 is not recognizable. To prove this we find reduction from non-recognizable language $\overline{A_{TM}} = \{< T, w >: T \text{ is TM that does not accept } w\}$ to L_2. Note that $\overline{A_{TM}}$ is not recognizable as A_{TM} is recognizable but not decidable.

Given $< T, w >$, define TM M as following:
If input x is given to M,
1. Run T with w as input for $|x|$ steps.
2. If T accepts w within $|x|$ steps, loop the machine infinitely.
3. Otherwise, if T has not accept w within $|x|$ steps, accept x.

This mapping $< T, w > \mapsto < M >$ is a reduction from $\overline{A_{TM}}$ to L_2:

i) Suppose $< T, w > \in \overline{A_{TM}}$. As T never accepts w, T does not accept w within $|x|$ steps for every input x. Thus, M accepts every input x by above construction. Hence, $L(M) = \Sigma^*$ is infinite and $< M > \in L_2$.

ii) On the other hand, suppose $< T, w > \notin \overline{A_{TM}}$. Let's say T accepts w in $k \in \mathbb{Z}_+$ number of steps. If $|x| \geq k$, then T accepts w within $|x|$ steps. Hence, M loops infinitely by the construction of M. This means M accepts no string of length greater or equal to k. Therefore, $L(M) \subseteq \cup_{i=0}^{k-1} \Sigma^i$. i.e. $L(M)$ is finite and so $< M > \notin L_2$.

Overall, we construct reduction from $\overline{A_{TM}}$ to L_2, showing that L_2 is not recognizable.

Problem 2

(2a) Let Turing machine P, Q recognize $\bar{A}, \bar{B}$ respectively. Then, for $w \in \Sigma^*$, (1)P accepts w if $w \notin A$ and (2)rejects or loops if $w \in A$. Similarly, (1)Q accepts w if $w \notin B$ and (2)rejects or loops if $w \in B$.

Now construct Turing machine as following. Given $w \in \Sigma^*$, run w on both P, Q. As $\bar{A} \cup \bar{B} = \overline{A \cap B} = \Sigma^*$, either P accepts or Q accepts on each string $\cdots(*)$. If P accepts before Q accepts the input, let the new Turing machine reject the input. Otherwise, if Q accepts before P accepts the input(or even accepts simultaneously), let the new Turing machine accept the input.

By $(*)$, the new Turing machine T halts on every input. (1) If $w \in A$, then P will never accept it. Thus, Q accept the input before P. Hence, T accepts w. (2) If $w \in B$, then Q never accepts it. Thus, P accepts the input before Q. Hence, T rejects w.

Let C be the language decided by T. Then, $A \subseteq C \subseteq \bar{B}$. i.e. C is a decidable language that separates A, B.

(2b) Suppose M is a TM that decides language C which separates A, B. Define TM T(recursively) as following:
Let an input w be given. If M accepts $< T, w >$(i.e. $< T, w > \in C$), then let T reject w. Otherwise if M rejects $< T, w >$(i.e. $< T, w > \notin C$, then let T accept w. Whether $< T, w > \in C$ or $< T, w > \notin C$, we conclude $< T, w > \notin C$ or $< T, w > \in C$ respectively having a contradiction. Thus, there is no such decidable language C.