CS2200: Languages, Machines, and Computation
Class test 4

May 1, 2024

Max Marks: 15
Duration: 1 hour

1. **[4 marks]** Recall the following languages that we saw in class.

$$\begin{aligned}\mathsf{MP}_\varepsilon &= \{\langle M\rangle \mid \varepsilon \in L(M)\}.\\ \mathsf{HP} &= \{(\langle M\rangle, x) \mid M \text{ halts on input } x\},\\ \overline{\mathsf{HP}} &= \{(\langle M\rangle, x) \mid M \text{ does not halt on } x\}.\end{aligned}$$

Consider the following statements about these languages.

(S1) $\mathsf{MP}_\varepsilon \leq_m \mathsf{HP}$

(S2) $\mathsf{HP} \leq_m \overline{\mathsf{HP}}$

Which one of the following is correct? Justify your answer.

A. Both (S1) and (S2) are true

B. (S1) is true and (S2) is false

C. (S1) is false and (S2) is true

D. Both (S1) and (S2) are false

Solution: Correct option is (B).

We know that for every r.e language L, $L \leq_m \mathsf{MP}$. Therefore, $\mathsf{MP}_\varepsilon \leq_m \mathsf{MP}$. We saw in class that $\mathsf{MP} \leq_m \mathsf{HP}$, and that $\leq_m$ is transitive. Therefore, $\mathsf{MP}_\varepsilon \leq_m \mathsf{HP}$.

If $\mathsf{HP} \leq_m \overline{\mathsf{HP}}$, then $\overline{\mathsf{HP}} \leq_m \mathsf{HP}$. This would mean that $\overline{\mathsf{HP}}$ is r.e. But then that would imply that HP is recursive, which is a contradiction.

2. **[6 marks]** For a Turing machine M, let $h(M)$ be the number of strings w such that M halts on w. Consider the following two languages.

$$\begin{aligned}L_1 &= \{\langle M\rangle \mid h(M) \geq 2200\},\\ L_2 &= \{\langle M\rangle \mid h(M) < 2200\}.\end{aligned}$$

Which of the following statement(s) is/are true? Justify your answer.

A. L_1 is recursive

B. L_2 is recursive

C. L_1 is r.e, but not recursive

D. L_2 is r.e, but not recursive

E. L_1 is not r.e

F. L_2 is not r.e

Solution: Options (C) and (F) are correct.

We will show that L_1 is not recursive by showing that $\mathsf{HP} \leq_m L_1$. Given $\langle M \rangle$ and a string x, construct a TM M' as follows:

$M'(y)$: Simulate M on x. If M halts on x, accept y

- If $(\langle M \rangle, x) \in \mathsf{HP}$, then $L(M') = \Sigma^*$. Therefore $h(M') \geq 2200$ and hence $\langle M' \rangle \in L_1$.
- If $(\langle M \rangle, x) \notin \mathsf{HP}$, then $L(M') = \emptyset$ and M' does not halt on any string. Therefore, $h(M') < 2200$ and hence $\langle M' \rangle \notin L_1$.

L_1 is r.e because we can do the timesharing simulation of M that we saw in class on all inputs, and accept if M halts on at least 2200 strings. In other words, we have a TM N that computes in phases. In phase i, N simulates M with the first i strings (lexicographically ordered) for i steps. Now, if $h(M) \geq 2200$, let k be the time step at which the first 2200 strings are accepted by M. Let ℓ be the smallest index such that the there are at least 2200 strings accepted by M within the first ℓ strings. Then in Phase $j = \max(\ell, k)$, N will observe that 2200 strings are accepted by M, and will halt and accept.

Observe that $L_1 = \overline{L}_2$ and this shows that $\overline{L}_2$ is not recursive, but r.e. Therefore, L_2 cannot be r.e since that would mean that L_2 and $\overline{L}_2$ are both r.e, leading to the contradiction that L_2 is recursive.

3. **[5 marks]** Let $L = \{(\langle M_1 \rangle, \langle M_2 \rangle) \mid L(M_1) \subseteq L(M_2)\}$. Show that L is not recursive.

Hint: Force $L(M_2)$ to be the empty set.

Solution: We will show that $\overline{\mathsf{HP}} \leq_m L$. Given $\langle M \rangle$ and a string x, we construct M_1 and M_2 as follows:

$M_2(y)$: Reject y and halt

$M_1(y)$: Simulate M on x. If M halts on x, then accept y.

- If $(\langle M \rangle, x) \in \overline{\mathsf{HP}}$, then $L(M_1) = \emptyset$ and $L(M_2) = \emptyset$. Therefore, $L(M_1) \subseteq L(M_2)$ and hence $(\langle M_1 \rangle, \langle M_2 \rangle) \in L$.
- If $(\langle M \rangle, x) \notin \overline{\mathsf{HP}}$, then $L(M_1) = \Sigma^*$, and $L(M_2) = \emptyset$. Therefore, $L(M_1) \nsubseteq L(M_2)$ and hence $(\langle M_1 \rangle, \langle M_2 \rangle) \notin L$.