Week C. By Deng Yufan.

Prob 1. Show $L = \{(M, w) \mid M \text{ halts with } w\}$ is RE but not R.

① L is RE: We construct a TM.

1° For input (M, w), translate the code of M, w into normal form.

2° Simulate M with input w.

3° If M halts, then accept. Otherwise never accept.

② L is not R: Prove by contradiction.

If L is R, then $\bar{L}$ is R, so $\exists M_1$, $L(M_1) = \bar{L}$. Then we can construct M_2 that $L(M_2) = L_u$ is R.

1° For input (M, w), translate into normal form.

2° Simulate M_1 with input (M, w).

3° If M_1 accept, then M_2 ~~don't~~ halt without accept. ~~Otherwise~~

4° Simulate M with input w.

5° If M accept, then M_2 accept. Otherwise M_2 halt without accept. Since $(M, w) \notin \bar{L}$, then M will halt, so M_2 will halt.

But we know L_u is not R. Contradiction.

Prob 2. Let L is RE and $\bar{L}$ is not. Consider $L' = \{0w \mid w \in L\} \cup \{1w \mid w \notin L\}$. Whether L' or $\bar{L'}$ are RE, R, or not RE?

L' and $\bar{L'}$ are not RE. If L' is RE, then $\exists M_1, L(M_1) = L'$. Consider M_2:

~~1° For input (M, w), translate into normal form,~~
~~2° Simulate M_1 with input~~

1° For input w, simulate M_1 with input 1w.

2° If M_1 accept, then M_2 accept. Otherwise never accept.

Then $L(M_2) = \bar{L}$. But we know $\bar{L}$ is not RE. Construction.

Proof for $\bar{L'} = \{\varepsilon\} \cup \{0w \mid w \notin L\} \cup \{1w \mid w \in L\}$ is similar to L'.

Prob 3. Are the following ~~lang.~~ problems RE or not?

a. Does $L(M)$ contain at least 2 strings?

Is RE. We construct a TM. (non-deterministic) -

1° For input M, translate into normal form.

~~2° Check strings in alphabet order (length first), to see whether current string is in M.~~ ~~(by simulate M with in~~

~~3° If seen 2 strings accept~~

2° Guess 2 strings M might accept. (One possible way is to guess one, then guess another).

3° Simulate M with input one of the 2 strings respectively.

4°. If the 2 strings all accepted by M, then accept. Otherwise never accept.

b. Is $L(M)$ infinite?

Is not RE. Consider $L = \{M \mid L(M) \text{ is infinite}\}$, then $\bar{L} = \{M \mid L(M) \text{ is finite}\}$.

We construct a TM (non-deterministic) that accept $\bar{L}$.

1° For input M, translate into normal form.

2° Guess a number that equals to $|L(M)|$.

Is not RE. If $\exists M_1$, $L(M_1) = \{M \mid L(M) \text{ is infinite}\}$, then $\exists M_2$, $L(M_2) = \overline{L_u}$.
Construct M_2 as:

1° For input (M, w), construct M' with input n as:

1°° Simulate M with input w.

2°° If M accept w in n steps, then ~~accept. Otherwise~~, halt without accep
Otherwise. accept.

2° Simulate M_1 with input M'.

3° If M_1 accept, then accept. Otherwise never accept.

Easy to see $L(M_2) = \overline{L_u}$. But we know L_u is RE but not R. contradiction.

Prob 4. Show the following problem is decidable:

TM that never make a leftmove on any input.

We prove that we can just check input that no longer than $|Q|$ to make sure TM never make a left move.

Otherwise, if for a TM M and input $|w| > |Q|$, TM make a left move ~~in the last move, then~~ after making $|Q|$ right move, then in the $|Q|+1$ moves, we have 2 samestate. Assume that after n right move and m right move. $(0 \le n < m \le |Q|)$. the current state of head are the same. then we can consider $w' = w_1 w_2 \cdots w_n w_{m+1} w_{m+2} \cdots w_k$.

So $|w'| < |w|$ and still make left move.

Then we can construct a TM:

1° For input M, count the number of state n.

2° For every string w length no longer than n, simulate M with input w.

3° If M make a left move, then halt without accept.

4° If all string no longer than n are checked, accept.