CSE 355: Intro to Theoretical Computer Science
Recitation #13 (20 pts)

1. [8 pts] A Turing machine with double infinite tape is like an ordinary Turing machine, but its tape is infinite in both directions, to the left and to the right. Prove that Turing machine with double infinite tape is equivalent to the ordinary Turing machine. (Note: you can assume that the tape is initially filled with blanks except for the portion that contains the input)

In order to show equivalence, both machines has to simulate each other.

Let infinite tape TM = D, and ordinary TM = M

D simulates M:

D marks the left-hand end of the input. Which will prevent D by moving its head to the left of the mark. In this case, D simulates M.

M simulates D:

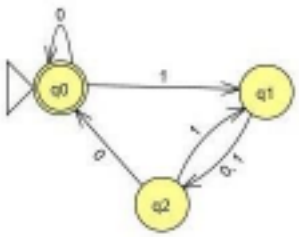
For M simulates D we let M be a two-tape TM, which has an equivalent power of an ordinary TM. Let M1 = first tape and M2 = second tape.

M1 contains input strings and all the blank spaces to its right side.

M2 contains left of the input string in reverse order.

In this case, D simulates M.

2. [6 pts] Answer questions for the following DFA *M* and give reasons for your answers.



2.1) Is $< M, 0100 > \in A_{DFA}$?

Yes, $q_0 0100 \rightarrow 0q_0 100 \rightarrow 01q_1 00 \rightarrow 010q_2 0 \rightarrow 0100q_0$ (accept)

2.2) Is $< M, 011 > \in A_{DFA}$?

No, $q_0011 \rightarrow 0q_011 \rightarrow 01q_11 \rightarrow 011q_2$ (reject)

2.3) Is $< M > \in A_{DFA}$?

No, since w is not given in proper formate, thus, M does not accept the input string

2.4) Is $< M, 0100 > \in A_{REX}$?

No, since M is a DFA and A_{REX} needs it to be a regular expression to check whether 0100 can be generated. Therefore M cannot generate string 0100

2.5) Is $< M> \in E_{DFA}$?

No, because the language accepted by M is not empty.

2.6) Is $< M, M > \in EQ_{DFA}$?

Yes, because $< M, N > \in EQ_{DFA}$ is in EQ_{DFA} if and only if the language accepted by M, N are the same. In this case, input <M, M> = (M = N). Therefore, $< M, M > \in EQ_{DFA}$.

3. [6 pts] Consider the problem of determining whether a DFA and a regular expression are equivalent. Express this problem as a language and show it is decidable (Hints: build a Turing decider).

Assume a Turing machine decide a language L = <D, R> , where D = DFA and R = regular expression.

Now, we convert R into DFA (N_R). Then, operate a TM M as a decider on input <D, N_R>

If M accepts, accept language L.

If M rejects, reject language L.