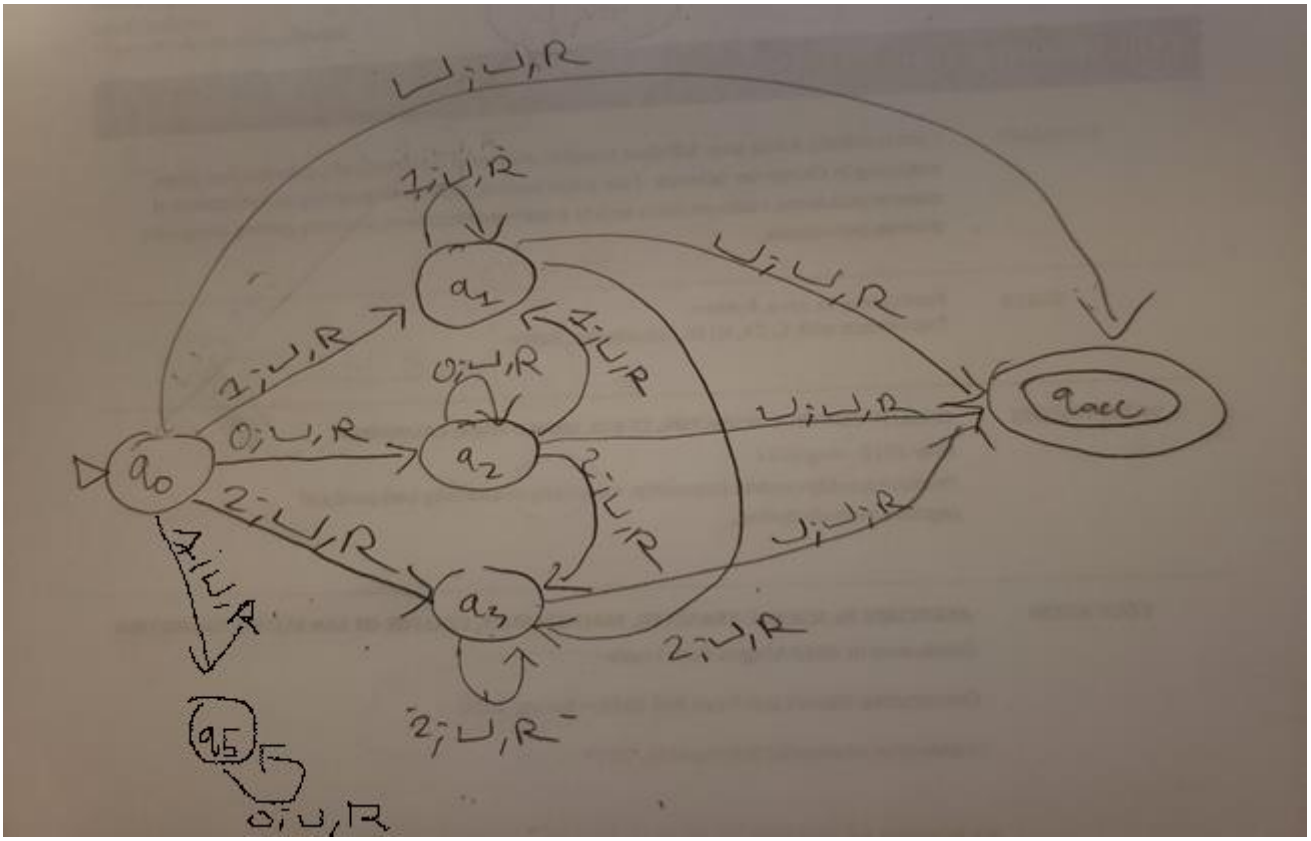
1.

a)



q0: Start state

q1: Last input read was a 1. Can read more 1s, or 2.

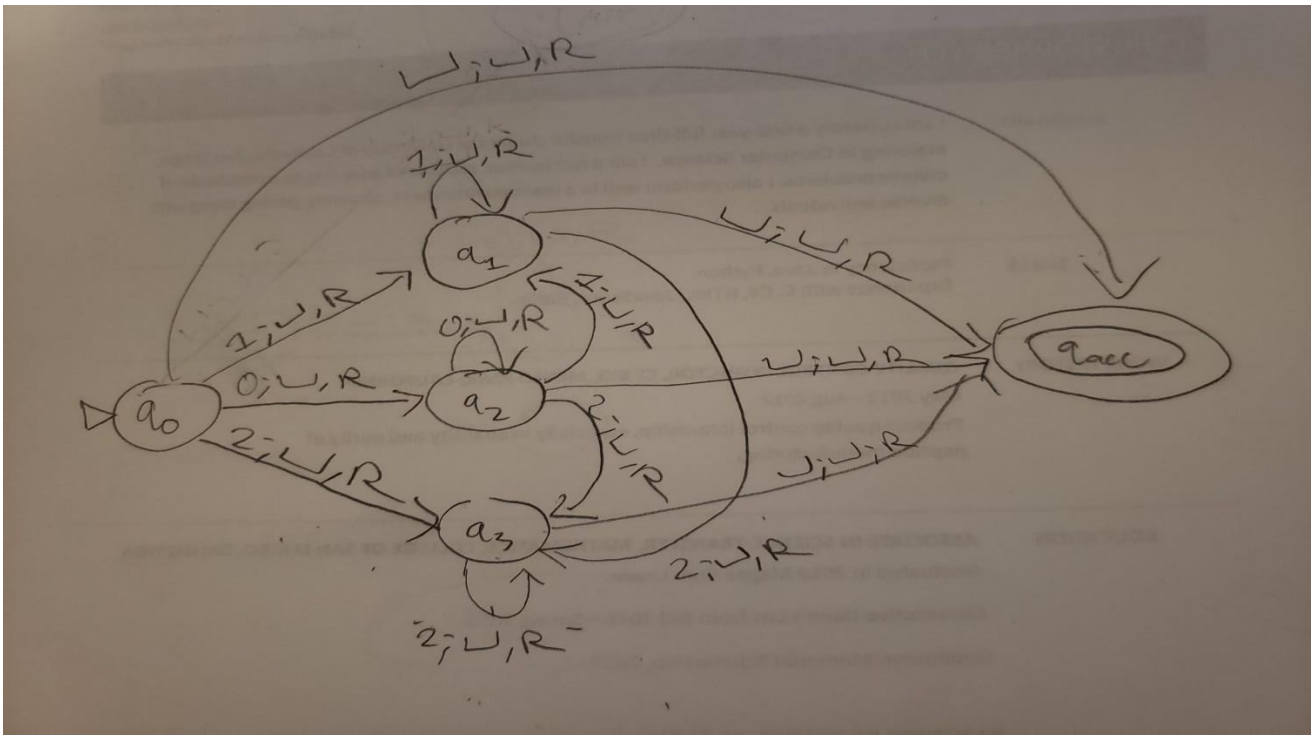
q2: Last input read was a 0. Can read more 0s, 1, or 2.

q3: Last input read was a 2. Can only read more 2s.

q5: Infinite loop to keep reading invalid strings of 100000….

q_{acc}: Tape is empty. Accept and halt.

b)



q0: Start state

q1: Last input read was a 1. Can read more 1s, or 2.

q2: Last input read was a 0. Can read more 0s, 1, or 2.

q3: Last input read was a 2. Can only read more 2s.

q_{acc}: Tape is empty. Accept and halt.

c) Yes. It can be described by the following high-level description:

Ignore input.

Have a list L, such that it is ordered in standard string order of all possible strings in {0, 1, 2}*; i.e. L_1 comes before L_2 comes before L_3 in standard string order and is within {0, 1, 2}*.

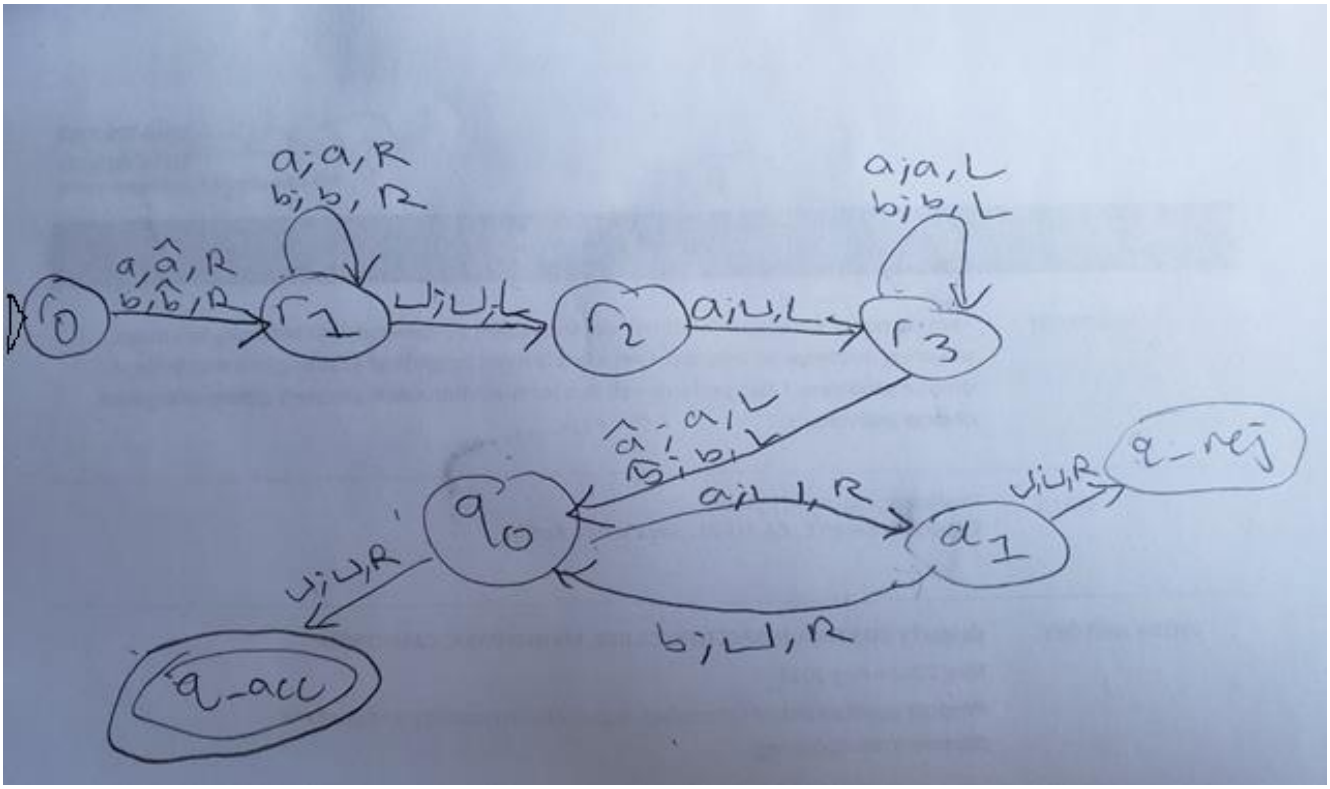
1. For i = 1, 2, …

    2. Run ND on L_i for (at most) i steps.

    3. If L_i is recognized in step 2, print L_i

2.

a)



b) No, as the construction of the transition functions do not allow for it. Our new start state, r0, only has transitions that read an a or a b. The empty string is rejected.

c) Yes. If M does not have any possible path to q_acc, M does not accept any strings.Then M_{new} will not accept any strings either. For example, from applying M_{new} to M_{sample} above, if M_{sample}'s q0 did not have a transion to q_acc, the machine would not accept any strings.

3.

a) Yes, every possible combinations of substrings x and y are considered for all possible inputs w such that their concatenation is xy, if M_1 accepts x and M_2 accepts y. The machine isn't a decider but is a recognizer, and we only need it to recognize when both machines accept their substrings.

b) Yes, as this construction has an enumerator recognizing all possible concatenations of strings from $L(E_1)$ and $L(E_2)$. All enumerators recognize TM recognizable languages. Also, every string in the language is printed out in finite time.

c) Yes, as all possible substring concatenations are considered for each possible input w. M_1 and M_2 are also limited to a finite amount of steps.