CS215 ASSIGNMENT 1

Due Wednesday, January 24, 11:59PM

Ivan Neto

Note that students are NOT allowed to copy sentences without showing their references.

Problem 1: Design a Turing Machine for the language L_1 given below.

$$L_1 = \{a^i b^j c^{ij} : i, j \geq 0\}.$$

Use the Turing Machine model with 2-way infinite tape. Your solution should consist of

(a) a high-level description in plain English of the underlying algorithm (at most 100 words),

(b) the state diagram (picture) of your Turing Machine, and

(c) the transition function, in the syntax consistent with the Turing Machine simulator at https:// turingmachinesimulator.com/. (Include the transition function in your assignment using the verbatim environment of LaTeX.)

The correctness of your TM will be determined by running it on a collection of test inputs, using the simulator at https://turingmachinesimulator.com/. So make sure that your TM works correctly on all legal inputs(all strings consisting of a's, b's and c's).

Solutions:

$M = (\mathbf{Q}, \Sigma, \Gamma, \delta, q_0, q_5, q_4)$

Such that

$\mathbf{Q} = \{q_{start}, q_{founda}, q_{qcheckc}, q_{foundb}, q_{qreject}, q_{accept}, q_{foundc}, q_{findcb}, q_{findcbcheck}, q_{findcbend}, q_{goback}\}$

Or

$\mathbf{Q} = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}\}$, respectively,

$\Sigma = \{a, b, c\}$

$\Gamma = \{a, b, c, -, \#, *, +\}$

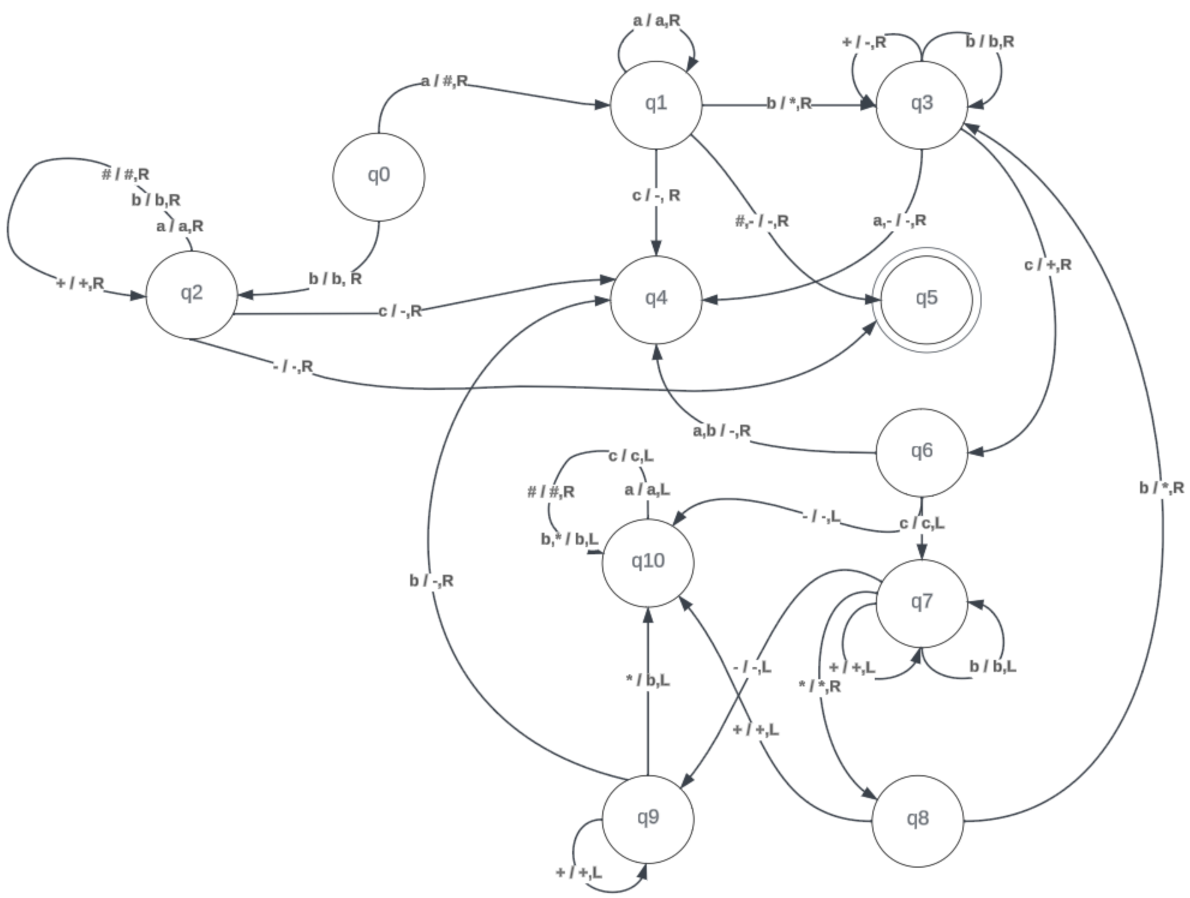
δ is subsequently defined.

Solution 1.a) High level description of my Turing Machine

M crosses out a c for every b, every time it encounters an a. More clearly:

1) Replace an a with "#".
2) Move right until you find a b. Replace the b with a "".
3) Move right until you find a c. Replace the c with a "+".
4) Go left, cross out the b next to the "".
5) Repeat step 3 and 4 until all b's and c's have been crossed out.
6) Go back to the beginning and cross out the next a. Repeat 1-5 until all a have been crossed.
7) Check if there are c's left. If yes, reject. If not, accept.

Solution 1.b) the state diagram (picture) of my Turing Machine



Solution 1.c) the transition function

```
q0,a
q1,#,>
q0,_
q5,_,>
q0,b
q2,b,>
q1,a
q1,a,>
q1,b
q3,*,>
q1,c
q4,_,>
q1,#
q5,_,>
q1,_
q5,_,>
q3,b
q3,b,>
q3,a
q4,_,>
q3,+
q3,+,>
q3,_
q4,_,>
q3,c
q6,+,>
```

```
q6,_
q7,_,<
q6,a
q4,_,>
q6,b
q4,_,>
q6,c
q7,c,<
q7,+
q7,+,<
q7,b
q7,b,<
q7,*
q8,*,>
q7,_
q9,_,<
q9,+
q9,+,<
q9,b
q4,_,>
q9,*
q10,b,<
q8,+
q10,+,<
q8,b
q3,*,>
q10,a
q10,a,<
q10,c
q10,c,<
q10,b
q10,b,<
q10,*
q10,b,<
q10,#
q0, #,>
q2,a
q2,a,>
q2,b
q2,b,>
q2,#
q2,#,>
q2,+
q2,+,>
q2,c
q4,_,>
q2,_
q5,_,>
```

Problem 2: Consider a modified Turing Machine model called a *List Turing Machine (LTM)*. A List Turing Machine, in addition to rewriting symbols, can also *delete* the current symbol, or *insert* a new symbol right before the current symbol. (Except for these new features, use the same TM convention as in Sipser's book.)

(a) Give a precise, formal definition of a List Turing Machine. Don't forget to give the definition of the language $L(M)$ accepted by a LTM M.

(b) Prove that List Turing Machines recognize only Turing recognizable languages. (In other words, you need to prove that if M is a List Turing Machine then there is a (standard) Turing Machine $M^{'}$ with $L(M^{'}) = L(M)$.)

Solutions:

Solution 2.a) formal description of a List Turing Machine

$LTM = (\mathbf{Q}, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$

Let LTM contain a 2-way infinite tape.

$\mathbf{Q}$ is the set of states.

Σ is the input alphabet minus $\{\sqcup\}$

Γ is the tape alphabet, where $\sqcup \in \Gamma$ and $\Sigma \subseteq \Gamma$.

$\delta = \mathbf{Q} \times \Gamma \rightarrow \mathbf{Q} \times \Gamma \times \{L, R, D, I\}$ where $I =$ "insert symbol" and $D =$ "delete symbol", is the transition function.

$q_0 \in \mathbf{Q}$ is the start state

$q_{accept} \in \mathbf{Q}$ is the accept state

$q_{reject} \in \mathbf{Q}$ is the reject state where $q_{reject} \neq q_{accept}$.

On input $w_1 w_2 w_3 ... w_n \in \Sigma^*$, the LTM behaves exactly as a normal Turing Machine except:

1) On tape movement "D", the current symbol is deleted and all symbols subsequent the current symbol are shifted left once. $s \in \Gamma$ does nothing for movement "D".

2) On tape movement "I", all values on and subsequent the current position are shifted right once. $s \in \Gamma$ is placed in the current position.

The language $L(LTM) = \Sigma^*$ because deleting or inserting symbols has no effect on the input language, only the behavior.

Solution 2.b) proof that a standard $M^{'}$ Turing Machine for $M = LTM$ exists such that $L(M^{'}) = L(M)$

Suppose no such machine $M^{'}$ exists. In other words, we cannot simulate M's "D" and "I" movements using "R" and "L" tape movements.

Simulating "D":

On configuration wq_0s, deleting the current symbol amounts to $wq_0s \rightarrow wq_1s'$ where $s' = s - s_0$.
On a standard Turing Machine, I can simulate this behavior by creating new transitions:

1) $\delta(q \in \mathbf{Q}, e \in \Sigma^*) = (q^{'}, \# \in \Gamma, R)$ to place a temporary symbol.

2) $\delta(q^{'} \in \mathbf{Q}, e \in \Sigma^*) = (q^{'}, e, R)$ to simulate moving "R" until the end.

3) $\delta(q \in \mathbf{Q}, \sqcup) = (s_{\sqcup} \in \mathbf{Q}, \sqcup, L)$ to move left once and be at the end of the input.

4) $\delta(s_{\sqcup}, e \in \Sigma^*) = (s_e, \sqcup, L)$ to move left and record the empty symbol.

5) $\delta(s_e, e \in \Sigma^*) = (s_e, \sqcup, L)$ to move left and record the previous symbol.

6) $\delta(s_e, \#) = (q_0^{'}, e, L)$ to move left and record the previous symbol.
7) $\delta(q_0^{'}, e \in \Sigma^*) = (q_0, e, R)$ to move right once into the correct position.

At the end, the configuration is wq_0s', so we have simulated the "D" movement using a standard Turing Machine.

Simulating "I":

On configuration wq_0s, inserting the symbol $a \in \Sigma^*$ before the current symbol amounts to $wq_0s \rightarrow waq_1s$.
On a standard Turing Machine, I can simulate this behavior by creating new transitions:

1) $\delta(q \in \mathbf{Q}, e \in \Sigma^*) = (q^e, \sqcup_b, R)$ to place the temporary symbol $\sqcup_b$, where b is the symbol we are inserting.
2) $\delta(q^e \in \mathbf{Q}, e \in \Sigma^*) = (q^e, e, R)$ to simulate shifting all elements to the right by one.
3) $\delta(q^e \in \mathbf{Q}, \sqcup) = (q^l, \sqcup, L)$ to simulate moving to the left once.
4) $\delta(q^e \in \mathbf{Q}, e \in \Sigma^*) = (q^l, e, L)$ to simulate moving to the left.
5) $\delta(q^e \in \mathbf{Q}, e \in \Sigma^*) = (q^l, e, L)$ to simulate moving to the left.
6) $\delta(q^e \in \mathbf{Q}, \sqcup_b) = (q_0, b, R)$ to simulate placing b and moving right once.

Since we can simulate both "D" and "I" movements using "R" and "L" tape movements, I reject the assumption that no machine $M^{'}$ exists for M such that $L(M^{'}) = L(M)$.

Therefore, I conclude that there exists an $M^{'}$ for a *Listing Turing Machine* $(LTM) = M$ such that $L(M) = L(M')$.

Academic integrity / collaboration statement

Resource 1 used in this homework were Michael Sipser's Theory of Computation book, in which I found a very nice definition of a Turing Machine that decides the language $C = \{a^i b^j c^k : i \times j = k \& i, j, k \geq 1\}$ (page 174).

I also used this book to aid my formal definition of M.

Resource 2 used in this homework was https://turingmachinesimulator.com/ to simulate my Turing Machine.

Resource 3 used in this homework was LucidChart to create the Turing Machine chart.

Everything else was done without any other resources.