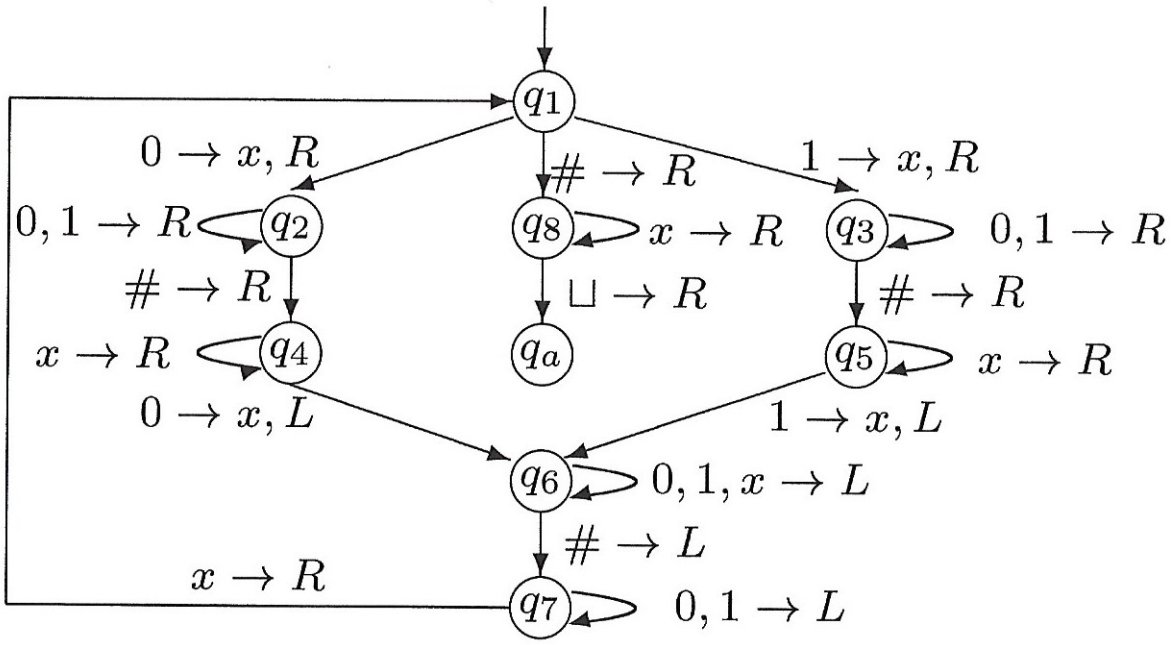
Problem Set #2

1 Theory Questions

Subjects: $CoRE$, Turing reduction, Turing machine, DFA.

1. Recall that up until now we saw one example of a language that is not even recursively enumerable: the language L^c_{Acc} (proven in the recitation). In this question we will introduce another example of such language: the language L^c_{Halt}. Follow the steps below:
   (a) Define $CoRE$.
   (b) State the connection we found between $CoRE, RE$ and R.
   (c) Use the connection in (b) and prove that $L^c_{Halt} \notin RE$.
2. Let
$$L_{Prime} = \{< P > |\ P\, is\, a\, program\, such\, that\, |L(P)|\, is\, prime\}$$
   Follow the Turing reduction examples we saw in class and recitation, and prove by Turing reduction that L_{Prime} is not decidable.
3. Consider the following Turing machine $M = (\Sigma, \Gamma, \sqcup, Q, q_1, q_a, q_r, \delta)$ where:
   - $\Sigma = \{0, 1, \#\}$
   - $\Gamma = \{0, 1, \#, x, \sqcup\}$
   - $\sqcup$ is the empty cell.
   - $Q = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_a, q_r\}$
   - q_1 is the initial state.
   - q_a is the accepting state.
   - q_r is the rejecting state.
   - the transition function δ is defined by the diagram below.
   - **Notations**:
     - As we said in class, if for a state $q \in Q$ and a symbol $c \in \Gamma$ no transition appears in the diagram - it means that we move to the rejecting state (q_r) and the input is rejected.
       For example, being in q_5 and reading # - the input is rejected.
     - When no Γ- symbol appears in the right side of an arrow, the symbol written to the cell is the same as the existing symbol.
       For example, $\# \to R$ in q_3 means that being in q_3 and reading # in the current cell - # is written to the current cell, and we move one cell to the right.
     - When more than one Γ- symbol appear in the left side of an arrow, it means that the same transition is relevant for all of those symbols.
       For example, $0, 1, x \to L$ in q_6 means that being in q_6 and reading any of $0, 1$ or x - we move one cell to the left.



(a) Let $w_1 = 10\#01, w_2 = 10\#10$. Provide the stages of running M on w_1 and on w_2.

(b) What language does M decide?

(c) Write a short description of the way that M acts given an input string $w \in \Sigma^*$, and use the description to verify your answer to (b).

4. Let $w \in \Sigma^*$ a string over an alphabet Σ.
$|w|$ denotes the length of w, and for every $c \in \Sigma$, $|c|_w$ denotes the number of occurrences of the symbol $'c'$ in w.
For example, if $w = 01101 \in \{0,1,2\}^*$ then $|w| = 5$, $|0|_w = 2$, $|1|_w = 3$, $|2|_w = 0$.

For each of the following languages, provide a DFA that accepts this language:

(a) $L_1 = \{w \in \{a,b,c\}^* \mid\mid a|_w \geq 3\}$

(b) $L_2 = \{w \in \{a,b,c\}^* \mid |w| \textit{ is not a multiply of } 4\}$

(c) $L_3 = \{w \in \{a,b\}^* \mid |a|_w \textit{is even and} |b|_w \textit{ is even}\}$

(d) $L_4 = \{a^n b^m c^k \mid n,m,k \geq 0\}$

2 Python programming

Subjects: Mutable and immutable.

Buns Buggy

Someone, and we don't want to mention any names, but someone made a mistake. An annoying one. That someone has implemented a Python library (the **buns buggy.py** file) that was supposed to help us store information about our favorite looney tunes. BUT, that someone created a bug, and one of the worst kind – a one that is hard to find. Your missions are threefold:

1. Find the problem. That is, submit the output string that is different than expected.
2. Find the bug. That is, what code made that bug happen? Provide the specific line(s) of code, and a short explanation as to what happened.
3. Fix the bug. That is, submit code that works properly.

Don't forget to submit the answers for all three tasks above.