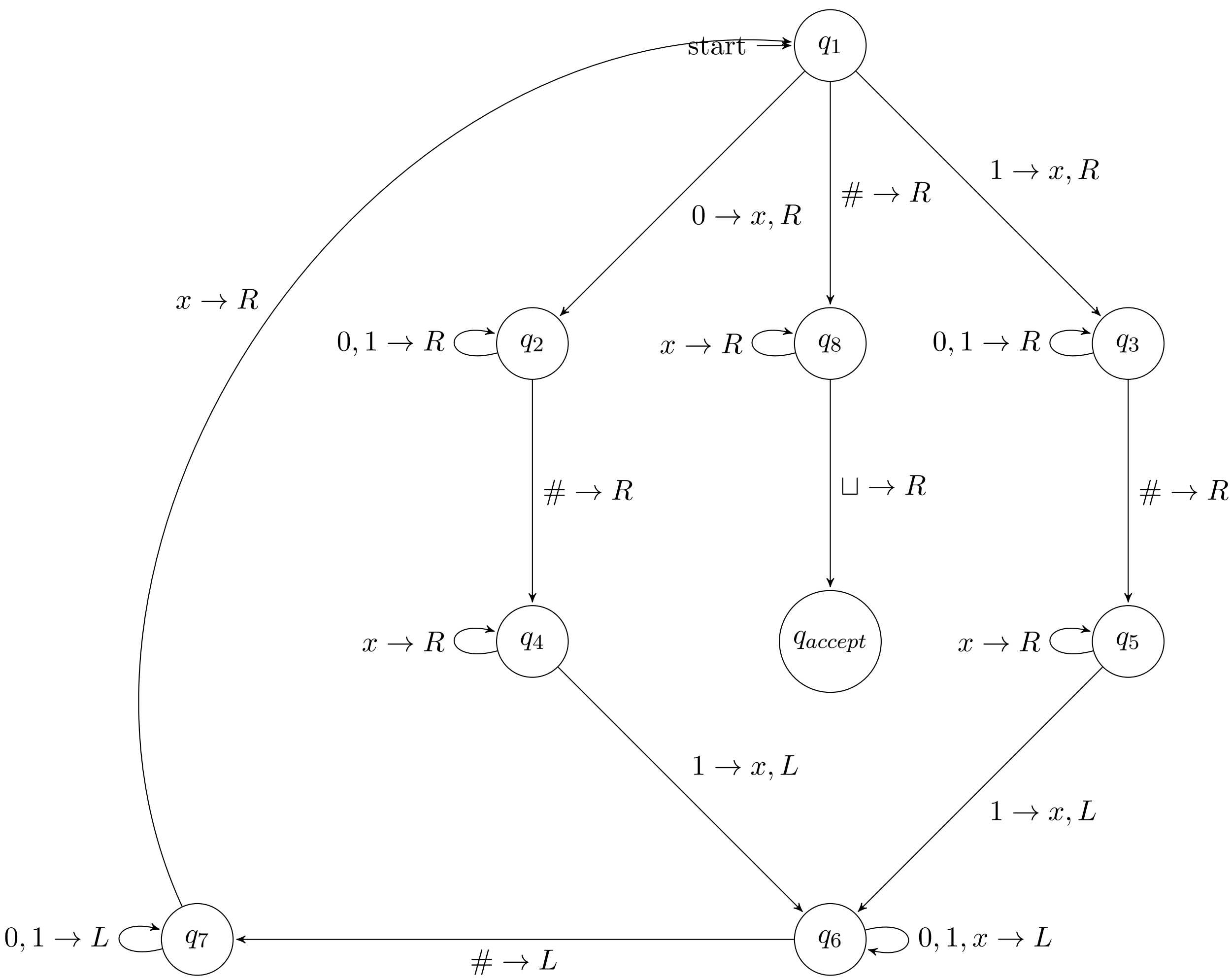
Homework 5 – Due Tuesday, March 4 at 11:59 PM

Reminder Collaboration is permitted, but you must write the solutions *by yourself without assistance*, and be ready to explain them orally to the course staff if asked. You must also identify your collaborators and write "Collaborators: none" if you worked by yourself. Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.

Problems There are 5 required problems and one bonus problem.

1. (**Low-Level to Implementation-Level**) Below is the state diagram of a Turing machine using input alphabet $\Sigma = \{0, 1, \#\}$ and tape alphabet $\Gamma = \{0, 1, \#, x, \sqcup\}$.

   The notation "$a \to R$" is shorthand for "$a \to a, R$." The reject state and transitions to the reject state are not shown. Whenever the TM encounters a character for which there is no explicit transition that means that the TM goes to the reject state. Use the convention that the head moves right in each of these transitions to the reject state.



(a) Give the sequences of configurations that this TM M enters when given as input strings (i) 01#11, (ii) 01#10, and (iii) 0##0. Use the same representation for your configurations as at the bottom of page 168 of Sipser, i.e., $q_0$01#11 means the TM is in state q_0 with its head on the left-most cell of the tape.

(b) Give an implementation-level description of the Turing machine described by this state diagram. Hint: The machine is very similar to Example 3.9 in Sipser.

(c) What is the language decided by M? That is, what is the set of strings that lead the TM M to halt and accept?

2. (**Implementation-Level to Low-Level**)

(a) Give a state diagram of a TM whose implementation-level description is below. The input alphabet of this TM is $\{a, b\}$ and the tape alphabet is $\{a, b, \sqcup\}$.

Input : String w

1. If the first symbol is blank then *accept*. If it is b then *reject*. If it is a then erase this a (i.e., replace it with a blank), move the head right, and go on to the next step.
2. Repeatedly move the head right until the blank symbol is found. After it is found move the head one cell to the left (to the last symbol of the string) and go on to the next step.
3. If this last symbol is not b then *reject*. Otherwise, erase this b, move the head one cell left, and go on to the next step.
4. If this last symbol is not (another) b then *reject*. Otherwise, erase this b, move the head one cell left, and go on to the next step.
5. Repeatedly move the head left until the blank symbol is found. After it is found move the head one cell to the right (to the first non-blank symbol of the string) and go back to step 1..

(b) Give the sequences of configurations that your TM enters when given as input strings (i) $aabbbb$, and (ii) $aabb$.

(c) What language is decided by the TM from part (a)?

3. (**Programming TMs**)

Write Turing machines that decide the following languages. That is, the machines should always halt after a finite number of steps on every input, and accept a string w if and only if w is in the given language. Implement your TMs in the following environment: http://morphett.info/turing/turing.html. Your solution should contain:

(i) An implementation-level description of your code.

(ii) Code that we can copy from your submission and run directly on that website. (Please add comments and make it as readable as possible.) There is a separate dropbox on Gradescope that will accept your code submissions.

(a) $L_1 = \{w \in \{0, 1\}^* \mid w \text{ contains an odd number of 1's}\}$.

(b) $L_2 = \{w \in \{0, 1\}^* \mid \text{ there are at least as many 1's in } w \text{ as there are 0's}\}$.

4. (**Recognizability vs. Decidability**) Recall the high-level description of a TM recognizer for the language $\{\langle p \rangle \mid p \text{ is a } k\text{-variate integer polynomial and there exists } x_1, \ldots, x_k \text{ such that } p(x_1, \ldots, x_k) = 0\}$ that we described in class.

   Input : Encoding of k-variate polynomial p
   1. For every possible setting of $x_1, \ldots, x_k$ to integer values:
   2. Evaluate $p(x_1, \ldots, x_k)$. If it equals 0, *accept*.

   Explain in a few sentences what is **wrong** about the following attempt to construct a TM decider for the same language:

   Input : Encoding of k-variate polynomial p
   1. For every possible setting of $x_1, \ldots, x_k$ to integer values:
   2. Evaluate $p(x_1, \ldots, x_k)$.
   3. If any evaluation equals 0, *accept*. Otherwise, *reject*.

5. (**Closure properties**)
   (a) Show that the class of **decidable** languages is closed under complement.
   (b) Explain why your construction from part (a) **fails** to show that the **Turing-recognizable** languages are closed under complement. (That is, if L is Turing-recognizable, explain why your construction from part (a) does not necessarily produce a TM recognizing $\overline{L}$.)

6. (**Bonus problem**) Let A be a Turing-recognizable language which is not decidable. (We will prove later in the course that such languages exist.) Consider a TM M that recognizes A. Prove that there are infinitely many input strings on which M loops forever.