CS 332: *Theory of Computation* | Prof. Mark Bun
Boston University | October 6, 2021

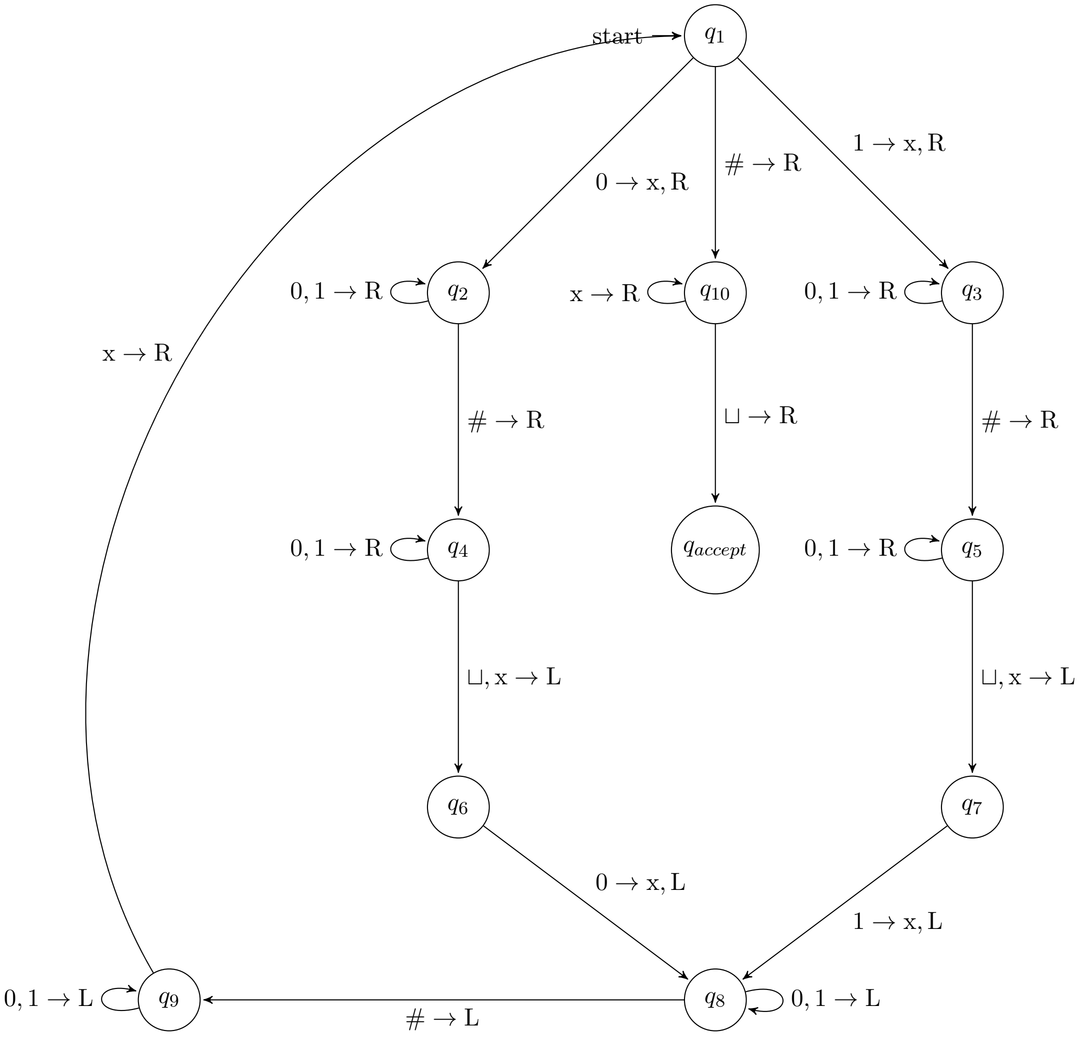
Homework 4 – Due Tuesday, October 12, 2021 at 11:59 PM

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.

Note You may use various generalizations of the Turing machine model we have seen in class, such as TMs with two-way infinite tapes, stay-put, or multiple tapes. If you choose to use such a generalization, state clearly and precisely what model you are using.

Problems There are 4 required problems.

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



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 tries to read 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) 010#100, (ii) 10#01, and (iii) 0##0. Use the same representation for your configurations as we did in lecture 9.

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

(c) What is the language decided by M?

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 $\{a, b, \text{x}, \sqcup\}$.

Input : String w

1. Read the first symbol on the input string. If it is a blank symbol, *accept*. If it is an a, then erase this a (i.e., replace it with a $\sqcup$), move the head right, and go to step 5. If it is a b, then erase this b, move the head right, and go on to the next step.
2. Continue moving the head right until a blank symbol is found. Replace this blank symbol with a b, move the head left, and go to step 6.
3. Move the head right until either an a or a blank symbol is found. If it is a blank symbol, *accept*. If it is a then cross out this a (i.e., replace it with an x), move the head left, and go on to the next step.
4. Repeatedly move the head left until the blank symbol is found. After it is found move the head one cell to the right and go on to the next step.
5. Continue moving the head right until a b or a blank symbol is found. If a blank symbol is found, *reject*. Otherwise, cross out the b that was found, move the head one cell left, and go on to the next step.
6. Continue moving the head left until the blank symbol is found. When it is found, move the head right and go back to step 3.

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

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

3. (**Programming TMs**)

Construct 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 your code as readable as possible.)

(iii) The sequences of configurations your TM enters on test inputs 00, 001111, and 010. (We will also run your solution on additional test inputs to check its correctness.)

Your TM should enter a state that is clearly marked "accept" or "reject" when it is done. It does not need to draw an emoji or anything else on its tape.

(a) $L_1 = \{w \in \{0,1\}^* \mid w \text{ contains the substring } 001\}$.

(b) $L_2 = \{0^n 1^{n^2} \mid n \geq 0\}$.

4. (**Closure properties**)

(a) For a language $L \in \Sigma^*$, define the *symmetrization* operator $SYM(L) = \{yx \mid x, y \in \Sigma^*, xy \in L\}$. For example, $SYM(\{010\}) = \{010, 100, 001\}$ and $SYM(\{010, 1100\}) = \{010, 100, 001, 1100, 1001, 0011, 0110\}$.

Suppose L is decided by a Turing machine M. Using M as a subroutine, give an implementation-level description of a Turing machine N that decides $SYM(L)$. Briefly explain in English what N does and why it works. It may be useful to use a multi-tape TM to solve this problem.

(b) Are the decidable languages closed under the SYM operator? Are the Turing-recognizable languages closed under the SYM operator? Explain your answers.