
INSTRUCTIONS

One member of the group should upload your group submission to Gradescope. During the submission process, they will be prompted to add the names of the rest of the group members. All group members' names and PIDs should be on **each** page of the submission.

Your homework must be typed. Your assignments in this class will be evaluated not only on the correctness of your answers, but on your ability to present your ideas clearly and logically. You should always explain how you arrived at your conclusions, using mathematically sound reasoning.

READING Sipser Sections 3.1, 3.2

KEY CONCEPTS Formal definitions of Turing machines, computations of Turing machines, halting computations, implementation-level descriptions of Turing machines, recognizable languages, decidable languages.

1. (10 points) In class and in the textbook (pages 166-167, page 173), we saw a Turing machine deciding the set $B = \{w\#w \mid w \in \{0,1\}^*\}$. In this question, you will modify this machine so that it decides the set

$$A = \{w\#w^R \mid w \in \{0,1\}^*\}$$

instead.

- a) Give an implementation-level description of your machine. *Follow the style and conventions demonstrated on page 167.*
- b) Prove that the machine you gave in part (a) **decides** A by showing that for any string $w \in \{0,1,\#\}^*$,
 - if $w \in A$, then your machine accepts on input w , and
 - if $w \notin A$, then your machine rejects on input w .
- c) Draw the state diagram formalizing the machine you described in part (a). *You may use the abbreviations used on page 173 and on this week's Individual Homework: omit all outgoing transitions for the accept and reject states, and omit all incoming transitions to the reject state.*

2. (10 points) Consider the function from $\{0, 1\}$ to $\{a, b\}$ given by $h(0) = b$, $h(1) = a$. We can extend h to operate on strings by defining $h(w) = h(w_1)h(w_2)\cdots h(w_n)$ where $w = w_1\cdots w_n$ and each $w_i \in \{0, 1\}$. For example, $h(010) = bab$. Extending h to languages over $\{0, 1\}$, we define $h(L) = \{h(w) \mid w \in L\}$ for any $L \subseteq \{0, 1\}^*$.

Devise a general construction that, given a Turing machine M over the alphabet $\{0, 1\}$, builds a Turing machine M' over the alphabet $\{a, b\}$ that recognizes $h(L(M))$.

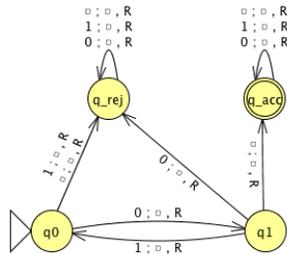
A full proof would have three stages: setup, construction, and proof of correctness. In this exercise you will focus on the setup and construction, and then apply your construction to an example.

(a) **Setup** Consider an arbitrary Turing machine $M = (Q, \{0, 1\}, \Gamma, \delta, q_0, q_{acc}, q_{rej})$, where $\{0, 1, \sqcup\} \subseteq \Gamma$ and $\{q_0, q_{acc}, q_{rej}\} \subseteq Q$ and $q_{acc} \neq q_{rej}$. Let $L = L(M)$.

Construction Build a new Turing machine whose language is $h(L)$. Specify this Turing machine formally, specifying each component of the 7-tuple formal definition.

You must supply this construction.

(b) **Application** Consider the language, L , recognized by this Turing machine:



Describe L mathematically (e.g. in set builder notation), and briefly justify your description.

Then, apply your construction from part (a) to this Turing machine and confirm that the language recognized by the resulting Turing machine is $h(L)$ by describing its computations.

Include the state diagram of the resulting machine as exported from JFLAP and your justifications in your homework submission.

3. (10 points) True or False (and justify).

- For each Turing machine M , there is some Turing machine M_1 with $L(M) = L(M_1)$ and such that M_1 never loops (i.e. its computations must accept or reject).
- For each Turing machine M , there is some Turing machine M_2 with $L(M) = L(M_2)$ and such that M_2 never rejects (i.e. its computations must accept or loop).
- For each Turing machine M , there is some Turing machine M_3 with $L(M) = L(M_3)$ and such that M_3 never accepts (i.e. its computations must reject or loop).