

## 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. We recommend that you submit early drafts to Gradescope so that in case of any technical difficulties, at least some of your work is present. You may update your submission as many times as you'd like up to the deadline.

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, 3.3

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) The state diagrams of the Turing machines  $C_1$  and  $C_2$  are on the next page.
  - a. Find a string of length 3 or 4 that is accepted by **both**  $C_1$  and  $C_2$ . Write out the computation of  $C_2$  on this string, using the notation discussed in class for configurations.
  - b. Find a string of length 3 or 4 that is rejected by  $C_1$  and accepted by  $C_2$ . Write out the computation of  $C_1$  on this string, using the notation discussed in class for configurations.
  - c. Here is an implementation-level description of one of the two machines

$C_?$  = "On input  $w$ :

1. If the current tape symbol is  $b$  or  $c$ , reject.
2. Otherwise, if the current tape symbol is blank or crossed out, then scan to the right and if all symbols before the first blank are crossed out, accept. If (while scanning), read a symbol that is not crossed out, reject.
3. Otherwise, if the letter being scanned is an  $a$ :
4. Cross off this  $a$  and scan to the right reading all  $a$ 's and crossed off symbols until a  $b$  occurs, then cross this  $b$  off and scan to the right reading all  $b$ 's and crossed off symbols until a  $c$  occurs and cross this  $c$  off.
5. If in step 4. we reach the end of the string before finding the symbol(s) we need to cross off, or if we see  $a$ 's after  $bs$  or  $c$ 's, or if we see  $bs$  after  $cs$ , reject.
6. Scan left over all crossed off  $c$ 's, crossed off  $b$ 's, uncrossed off  $b$ 's, and uncrossed off  $a$ 's, and put tape head on the cell to the right of last crossed off  $a$ .
7. Go to instruction 1."

Does  $C_? = C_1$  or  $C_? = C_2$ ? Explain why the other machine does not match this description.

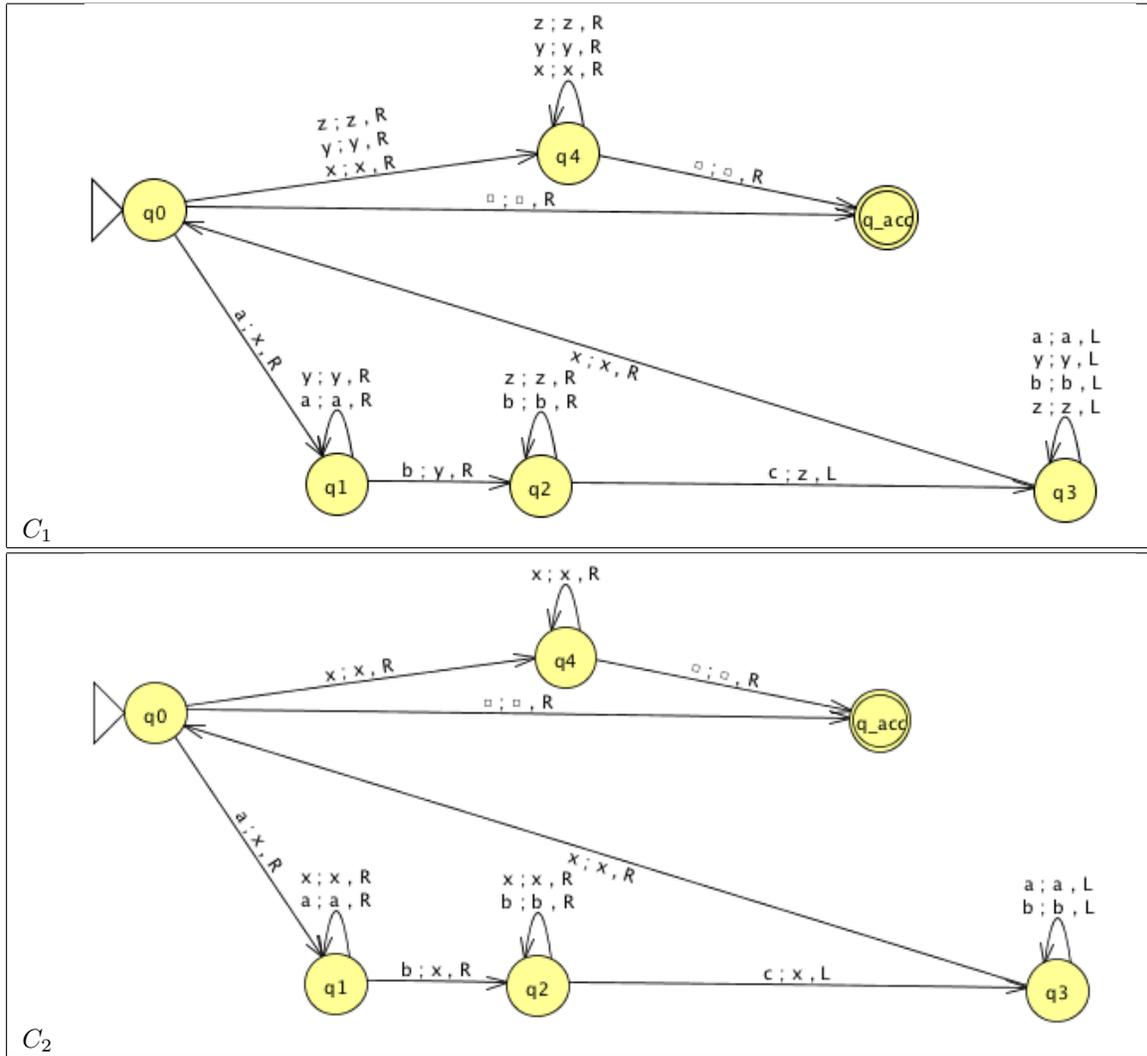
- d. Is  $C_?$  a decider? Briefly justify your answer.
- e. What is  $L(C_?)$ ? Use mathematical notation (i.e. set builder notation) or precise English.

**State diagrams of candidate Turing machines for Problem 1.**

Input alphabet is  $\{a, b, c\}$ .

Tape alphabet for  $C_1$  is  $\{a, b, c, x, y, z, \square\}$ . Tape alphabet for  $C_2$  is  $\{a, b, c, x, \square\}$ .

**Conventions:** (1) omit the reject state from the diagram, (2) any missing transitions in the state diagram have value  $(q_{reject}, \square, R)$ .



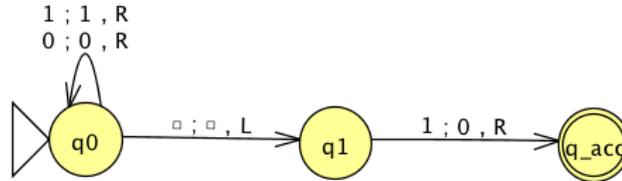
As an example of the notation of configurations, the computation of  $C_2$  on input  $ab$  is described by the sequence of configurations

$q_0 a b$   
 $x q_1 b$   
 $x x q_2 \square$   
 $x x \square q_{reject}$

**2. (10 points)** A **clean Turing machine** (cTM) is defined to be a Turing machine which satisfies the additional constraint that, at the end of each accepting computation, the tape is empty (i.e. all cells are blank).

In this problem, you'll show that every Turing machine is equivalent to a clean Turing machine. To describe your construction, you'll first demonstrate how it works on an example, and then write out the general case formally.

- a. Consider the Turing machine over the alphabet  $\Sigma = \{0, 1\}$  given by the state diagram (using the usual conventions listed in Problem 1):



Draw the state diagram of a clean Turing machine that is equivalent to this one in JFLAP, export the image as a png or jpg file, and include it as part of your submission. You may use the standard conventions (like in Problem 1) for drawing state diagrams.

You do not need to justify your construction for credit, **but** if you describe how your state diagram works by explaining the role of each state and transition, we may be able to award partial credit if your answer is incorrect.

- b. Consider an arbitrary Turing machine  $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$ . Define a clean Turing machine  $M'$  such that  $L(M) = L(M')$  by giving precise (but general) definitions for each parameter in the formal definition of  $M' = (Q', \Sigma', \Gamma', \delta', q'_0, q'_{acc}, q'_{rej})$ .

For the purpose of your construction, you may **assume** that  $M$  does not ever write a blank symbol ( $\square$ ) to a non-blank cell of the tape.

You do not need to justify your construction for credit, **but** if you describe how your construction works with an implementation-level description, we may be able to award partial credit if your answer is incorrect.

*Extension (not for credit):* how would you change your construction from part b. if you allowed  $M$  to write blank symbols to the tape?

**3. (10 points)** Fix an arbitrary alphabet  $\Sigma$ .

- Prove that the class of decidable languages over  $\Sigma$  is closed under the Kleene star operation.
- Prove that the class of recognizable languages over  $\Sigma$  is closed under the STUTTER operation.

Use high-level descriptions of Turing machines for both these proofs. You must use the model of Turing machine defined in Sipser Section 3.1 for these constructions: assume that we have a single tape, that it is one-way infinite, and that computations are deterministic.

*Extension (not for credit):* is the class of recognizable languages over  $\Sigma$  closed under the Kleene star? is the class of decidable languages over  $\Sigma$  closed under STUTTER?