Reminder: Lab problems are graded and must be submitted along with homework when lab writeups are due. Your name should not appear anywhere on your handin; each individual page of the homework should include your Banner ID only. For your digital submission, each page should include work for only one problem (i.e., make a new page/new pages for each problem).

While collaboration is encouraged in this class, please remember not to take away notes from any labs or collaboration sessions.

Please monitor Piazza, as we will post clarifications of questions there. Hand in your solutions by 10:20am to Gradescope. Late homeworks are not accepted.

Problem 1

Ginger’s friend Ginny Tseng convinces her to try online dating—after all, it’s how Ginny met Cole Robby. Ginger decides to look for a date on JicaMate. She doesn’t want to let her family know anything about her search, though, as they think she should be focusing on her farmwork. Luckily, they’re bored to tears when they see text that’s just 0, 1, and \( \omega \). It doesn’t help that many of the O’Nions have chronically watery eyes and generally poor eyesight.

a. Ginger has come up with a clever plan to protect her messages: instead of using the English alphabet, she will use the alphabet \( \{0,1\} \). To put her plan into action, she needs to convince herself that she can encode any message into an encrypted form. She knows that two Turing machines \( T_1, T_2 \) with different input alphabets \( \Sigma_1, \Sigma_2 \) are equivalent if and only if there is some function \( f : \Sigma_1 \rightarrow \Sigma_2 \) such that \( f \) is invertible (essentially, \( f \) is a lookup table where each input has a unique output) and for every string \( x \) that is accepted, rejected, or looped forever on by \( T_1, T_2 \) similarly accepts, rejects, or loops forever on input \( f(x) \). Help Ginger by showing how any Turing machine (with original input alphabet \( \Sigma \) ) can be converted into an equivalent Turing machine with input alphabet \( \{0,1\} \).

**Hint:** the tape alphabet does not need to be \( \{0,1,\omega\} \).
b. Ginger has found a promising match in a guy named Rudy, but she needs
to keep her thoughts on him secret from her family. Describe how any
Turing machine (with original tape alphabet $\Gamma$) can be converted to a
Turing machine with tape alphabet \{0, 1, \_\} such that the languages of
both are the same. You may assume that a state transition can move the
head to left or right (as in a standard TM) and can also choose to keep
the head stationary on the tape.

c. All this planning has gotten Ginger interested in what else can be done
with only a binary choice. She’s especially intrigued by JicaMate’s algo-
rithms, which somehow condense the problem of finding a soulmate to
answering yes/no questions. Describe how any nondeterministic Turing
machine (such as finding a soulmate) can be converted into a nondeter-
ministic Turing machine where any \((\text{state}, \text{symbol})\) tuple has at most two
transitions. You may assume that a state transition can choose to keep
the head stationary on the tape.

d. Justify why the choices made by a nondeterministic TM can be encoded
as a binary string.

Problem 2

In this problem, you will give a high level description of a Turing machine
that recognizes the following language. Let $\Sigma = \{0, 1, \times, =\}$. Then:

$L = \{w \mid w \text{ is of the form } "x \times y = z" \text{ with } x, y, z \in (0 \cup 1)^* \text{ such that the product of } x \text{ and } y \text{ is } z \text{ when interpreted as binary integers}\}$

a. First, explain the operation of incrementing and decrementing. That
is, if there is a string $x \in (0 \cup 1)^*$ on the tape of your TM, give a high
level description of how you would increment and decrement $x$ by one
(for decrementing, ignore the case where $x = 0$).

b. Now give a high level description of the TM that recognizes $L$.

**Note:** There are multiple ways of doing binary multiplication, some
of which have better runtimes than others. You will not be graded on
the runtime of your TM—only worry about correctness. You can just
say ‘increment’ and ‘decrement’ without re-explaining the process. It
may also be useful to consider a multi-tape Turing machine.
Problem 3

Consider the following Turing machine variant: a two-layered Turing machine is a Turing machine whose input tape has two symbols in each cell, a top symbol and a bottom symbol. A two-layered Turing machine transitions like a regular Turing machine, except that at each step, it simultaneously reads both the top and bottom symbols in the current cell, and likewise simultaneously updates both the top and bottom symbols in the current cell. (Different values can be written to the top and bottom symbols.) The input to a two-layered Turing machine appears on the bottom of cells on the tape starting from the left; all the top symbols (as well as the bottom symbols beyond the input cells) are initially empty.

a. Show that any language recognized by a two-layered Turing machine can also be recognized by a regular Turing machine.

b. How could you generalize part (a) for a Turing machine variant with an $n$-layered input tape (which has $n$ symbols in each cell)?