You must submit your assignment through the Crowdmark system. You will receive by email an invitation through which you can submit your work in the form of separate PDF documents with your answers to each question of the assignment. To work with a partner, you and your partner must form a group on Crowdmark. Crowdmark does not enforce a limit on the size of groups. The course policy that limits the size of each group to at most two remains in effect: submissions by groups of more than two persons will not be graded.

- It is your responsibility to ensure that the PDF files you submit are legible. To this end, I encourage you to learn and use the LaTeX typesetting system, which is designed to produce high-quality documents that contain mathematical notation. You are not required to produce the PDF files you submit using LaTeX; you may produce them any way you wish, as long as the resulting document is legible.
- By virtue of submitting this assignment you (and your partner, if you have one) acknowledge that you are aware of the policy on homework collaboration for this course.\(^a\)
- For any question, you may use results previously described in class, or in prerequisites of this course, without describing them. You may also use any result that we covered in class, or is in the assigned sections of the official course textbook, by referring to it.
- Unless we explicitly state otherwise, you should justify your answers. Your paper will be graded based on the correctness and efficiency of your answers, and the clarity, precision, and conciseness of your presentation.

\(^a\) In each homework assignment you may collaborate with at most one other student who is currently taking CSCC63. If you collaborate with another student on an assignment, you and your partner must submit only one copy of your solution, with both of your names. The solution will be graded in the usual way and both partners will receive the same mark. Collaboration involving more than two students is not allowed. For help with your homework you may consult only the instructor, TAs, your homework partner (if you have one), your textbook, and your class notes. You may not consult any other source.

In your answers for this assignment you may assume (by Church’s thesis) that any algorithm described in pseudocode that involves no magical thinking (i.e., statements like “a miracle happens and . . .”) can be implemented by a Turing machine. You do not have to actually demonstrate Turing machines with detailed descriptions of their states, state transition functions, and so on.

**Question 1.** (15 marks) Recall that the Kleene closure (or Kleene star) of a language $L$, denoted $L^*$, is the language consisting of the concatenation of zero or more strings in $L$. That is, $x \in L^*$ if and only if $x = \epsilon$ or there are strings $x_1, x_2, \ldots, x_k \in L$ (not necessarily distinct) such that $x = x_1 x_2 \ldots x_k$.

Prove that for any language $L$, if $L$ is decidable then $L^*$ is also decidable.

**Hint.** You may find nondeterminism useful.
Question 2. (10 marks)

a. Let \( ZO = \{0^n1^n : n \in \mathbb{N}\} \). (Here the notation \( a^n \), where \( a \) is a symbol and \( n \in \mathbb{N} \), denotes the string consisting of \( n \) consecutive \( a \)'s; \( a^0 = \epsilon \).) Prove that, for any language \( L \), \( L \leq_m ZO \) if and only if \( L \) is decidable.

b. Let \( U = \{(M, x) : \text{Turing machine } M \text{ accepts input } x \} \). (This is the language I called “universal” in the lectures — because it is the language recognized by the universal Turing machine — and that is denoted \( A^TM \) in your textbook.)

Prove that, for any language \( L \), \( L \) is recognizable if and only if \( L \leq_m U \).  

Question 3. (15 marks)

a. Prove that the following language is decidable:

\[ L_1 = \{(M) : \text{Turing machine } M \text{ on empty tape eventually writes a non-blank symbol}\} \]

b. Prove that the following language is undecidable:

\[ L_2 = \{(M, a) : \text{Turing machine } M \text{ on empty tape eventually writes symbol } a \} \]

c. Is \( L_2 \) recognizable? Justify your answer.

Question 4. (20 marks) For each of the following languages, state whether it is (i) decidable and (ii) recognizable. Justify your answers.

a. \( L_3 = \{(M) : \text{Turing machine } M \text{ halts on exactly three inputs}\} \)

b. \( L_4 = \{(M_1, M_2) : \text{there are at least three inputs that Turing Machines } M_1 \text{ and } M_2 \text{ both accept}\} \)

c. \( L_5 = \{(M_1, M_2, k) : \text{there are exactly } k \text{ inputs that the Turing machines } M_1 \text{ and } M_2 \text{ either both accept or both do not accept}\} \)

d. \( L_6 = \{(M, x) : \text{Turing machine } M \text{ accepts } x \text{ if and only if } |\langle M \rangle| < |x|\} \)

(i.e., \( M \) accepts exactly the strings that are shorter longer than its encoding).

---

1This result can be viewed as saying that \( U \) is the “hardest” recognizable language since an algorithm for \( U \) would imply algorithms for \textit{all} recognizable languages. Languages with this property are called “recursively enumerable complete” (or RE-complete, for short). (Recall that “recursively enumerable” and “semi-decidable” are synonyms for what your textbook calls “Turing-recognizable” and that I tend to abbreviate as just “recognizable”.) Another example of an RE-complete language is the language that corresponds to the halting problem \( H = \{(M, x) : \text{Turing machine } M \text{ halts on input } x \} \).) We will study extensively an analogous concept in the second half of the course: NP-complete languages (or problems) are the “hardest” in the class NP.