Homework 2

Due on Monday, 10:30 am, 28 December 2020 (2020/12/28)

Name :

Student ID :

Note:

1. Write down clearly your name and student ID in the space above.
2. There are FIVE questions altogether.
3. Write your solution for each question in the space provided.
4. Submit your solution before the lesson on 28 December 2020. If you want to submit it earlier, you can slip it under the door of my office.

Question 1 (2 points). Consider the following Turing machine $\mathcal{M}_1 = \langle \Sigma, \Gamma, Q, q_0, q_{\text{acc}}, q_{\text{rej}}, \delta \rangle$.

- $\Sigma = \{0, 1\}$.
- $\Gamma = \{\lhd, 0, 1, \sqcup\}$.
- $Q = \{q_0, p_0, p_1, s, t, r_0, r_1, q', q_{\text{acc}}, q_{\text{rej}}\}$.
- $q_0, q_{\text{acc}}, q_{\text{rej}}$ are the initial, accepting and rejecting states, respectively.
- δ is defined as follows.

$(q_0, \sqcup) \to (q_{\text{rej}}, 0, \texttt{Stay})$	$(p_0, \sqcup) \to (q_{\text{rej}}, 0, \texttt{Stay})$	$(p_1, \sqcup) \to (s, 1, \texttt{Stay})$
$(q_0, 0) \to (p_0, \lhd, \texttt{Right})$	$(p_0, 0) \to (p_0, 0, \texttt{Right})$	$(p_1, 0) \to (p_0, 1, \texttt{Right})$
$(q_0, 1) \to (p_1, \lhd, \texttt{Right})$	$(p_0, 1) \to (p_1, 0, \texttt{Right})$	$(p_1, 1) \to (p_1, 1, \texttt{Right})$
$(q_0, \lhd) \to (q_{\text{rej}}, \lhd, \texttt{Stay})$	$(p_0, \lhd) \to (q_{\text{rej}}, \lhd, \texttt{Stay})$	$(p_1, \lhd) \to (q_{\text{rej}}, \lhd, \texttt{Stay})$
$(s, \sqcup) \to (q_{\text{rej}}, 0, \texttt{Stay})$	$(t, \sqcup) \to (q_{\text{rej}}, 0, \texttt{Stay})$	$(q', \sqcup) \to (q', 0, \texttt{Left})$
$(s, 0) \to (t, 1, \texttt{Left})$	$(t, 0) \to (t, 0, \texttt{Left})$	$(q', 0) \to (r_0, 0, \texttt{Left})$
$(s, 1) \to (s, 0, \texttt{Left})$	$(t, 1) \to (t, 1, \texttt{Left})$	$(q', 1) \to (r_1, 0, \texttt{Left})$
$(s, \lhd) \to (r_1, \lhd, \texttt{Right})$	$(t, \lhd) \to (q_{\text{acc}}, \lhd, \texttt{Stay})$	$(q', \lhd) \to (q_{\text{acc}}, \lhd, \texttt{Right})$
$(r_0, \sqcup) \to (t, 0, \texttt{Left})$	$(r_1, \sqcup) \to (t, 1, \texttt{Left})$	
$(r_0, 0) \to (r_0, 0, \texttt{Right})$	$(r_1, 0) \to (r_0, 1, \texttt{Right})$	
$(r_0, 1) \to (r_1, 0, \texttt{Right})$	$(r_1, 1) \to (r_1, 1, \texttt{Right})$	
$(r_0, \lhd) \to (q_{\text{rej}}, \lhd, \texttt{Stay})$	$(r_1, \lhd) \to (q_{\text{rej}}, \lhd, \texttt{Stay})$	

Determine the run of $\mathcal{M}$ on each of the following input words: ϵ, 11, 00, 01.

Solution for question 1.

Question 2 (2 points). In the following, for a Turing machine $\mathcal{M}$, we denote by $L(\mathcal{M})$ the language that consists of all the words accepted by $\mathcal{M}$. That is, $L(\mathcal{M}) = \{w \mid \mathcal{M} \text{ accepts } w\}$.

Consider the following Turing machine A that works as follows.

INPUT: $\lfloor\mathcal{M}\rfloor\$w$.

- Construct a TM $\mathcal{K}_{\mathcal{M},w}$ that works as follows.

  INPUT: $u \in \Sigma^*$.
  - Run $\mathcal{M}$ on w.
  - If $\mathcal{M}$ accepts w, ACCEPT.
  - If $\mathcal{M}$ rejects w, do the following:
    * Check if u is a string of the form 0^n1^n, for some $n \geqslant 0$.
    * If it is, ACCEPT.
    * Otherwise, REJECT.
- Output $\lfloor\mathcal{K}_{\mathcal{M},w}\rfloor$.

Answer each of the following questions

(a) If $\mathcal{M}$ accepts w, what is the language $L(\mathcal{K}_{\mathcal{M},w})$?

(b) If $\mathcal{M}$ rejects w, what is the language $L(\mathcal{K}_{\mathcal{M},w})$?

(c) If $\mathcal{M}$ does not halt on w, what is the language $L(\mathcal{K}_{\mathcal{M},w})$?

(d) Recall the language $L_5 := \{\lfloor\mathcal{M}\rfloor \mid L(\mathcal{M}) \text{ is a regular language}\}$ in Note 9. Is the following true?

$$\lfloor\mathcal{M}\rfloor\$w \in \mathsf{HALT} \quad \text{if and only if} \quad \lfloor\mathcal{K}_{\mathcal{M},w}\rfloor \in L_5$$

Justify your answer (in a few sentences).

Solution for question 2.

Question 3 (1 point). Recall that $L(\mathcal{M})$ denotes the language of all the words accepted by Turing machine $\mathcal{M}$. Consider the following language:

$$L_\infty \quad := \quad \{\lfloor \mathcal{M} \rfloor \ \mid \ L(\mathcal{M}) \text{ is infinite}\}.$$

That is, L_∞ consists of all descriptions of Turing machines that accepts infinitely many words. Prove that L_∞ is undecidable.

Note: You are *not* allowed to use Rice's theorem here. Also, don't prove Rice's theorem here and then use it as your solution. Just present a straightforward reduction from one of the undecidable languages, in the same style as algorithm A in question 2.

Solution for question 3.

Question 4 (1 point). Prove that if L is decidable and K is undecidable, then $L \leqslant_m K$.

Note: Here you should also present your reduction in the same style as algorithm A in question 2.

Solution for question 4.

Question 5 (2 points). Consider the following problem, which we denote by CFL-Reversal.

CFL-Reversal	
Input:	A CFG $\mathcal{G} = \langle \Sigma, V, R, S \rangle$.
Task:	Output `True`, if $L(\mathcal{G})$ is closed under reversal. Otherwise, output `False`.

Prove that CFL-Universality $\leqslant_m$ CFL-Reversal, and hence, CFL-Reversal is undecidable.

Note 1: A language L is closed under reversal, if for every word $w \in L(\mathcal{G})$, its reversal $w^r \in L(\mathcal{G})$.

Note 2: You should present a pseudo code that, on input a CFG $\mathcal{G}$, outputs a CFG $\mathcal{G}'$ such that $L(\mathcal{G}) = \Sigma^*$ if and only if $L(\mathcal{G}')$ is closed under reversal. It has to be explicit about the rules in $\mathcal{G}'$. At the same time you should describe intuitively (in a few sentences) the language $L(\mathcal{G}')$.

Solution for question 5.

Question 6 (2 points). A *linear inequality* (over integer) is an inequality of the form:

$$\alpha_1 z_1 + \cdots + \alpha_n z_n \quad \circledast \quad \beta,$$

where $\alpha_1, \ldots, \alpha_n, \beta$ are integers, $z_1, \ldots, z_n$ are variables and $\circledast$ is either $\leqslant$ or $\geqslant$.

Let $\mathcal{C}$ be a set of linear inequalities with variables $z_1, \ldots, z_n$. We say that $\mathcal{C}$ *has a solution* in $\mathbb{Z}$, if we can assign the variables $z_1, \ldots, z_n$ with integers so that all the inequalities in $\mathcal{C}$ are satisfied.

Consider the following problem, whcih we denote by Problem-X.

Problem-X	
Input:	A set $\mathcal{C}$ of inequalities (where numbers in the inequalities are written in binary).
Task:	Output `True`, if $\mathcal{C}$ has a solution in $\mathbb{Z}$. Otherwise, output `False`.

Prove that Problem-X is **NP**-hard.

Note: You are only asked to prove that it is **NP**-hard, *not* **NP**-complete. We can indeed show that Problem-X is **NP**-complete, but the **NP** membership is a little bit tricky to prove. The **NP**-hardness part is actually the easy part.

Solution for question 6.