Fall 2021 CSE Qualifying Exam
CSCE 551 Answer Key

1. Let Σ be any alphabet. For any languages $L_1, L_2 \subseteq \Sigma^*$, define $L_1 \diamondsuit L_2$ to be the set of all strings obtained by overlapping a string from L_1 followed by a string from L_2. (The "overlap" could be the empty string.). Formally,

$$L_1 \diamondsuit L_2 := \{xyz \mid x, y, z \in \Sigma^* \text{ and } xy \in L_1 \text{ and } yz \in L_2\} .$$

   Show that if L_1 and L_2 are regular, then $L_1 \diamondsuit L_2$ is regular. If your proof involves a correct construction, then you do not need to prove that it is correct.

   Answer: (No one did this problem.)

2. Let Σ be some alphabet. Say that a language $L \subseteq \Sigma^*$ is *length-dependent* iff, for all strings $x, y \in \Sigma^*$, if $x \in L$ and $|x| = |y|$, then $y \in L$. Define the language $\mathrm{LD}_{\mathsf{TM}}$ as follows:

$$\mathrm{LD}_{\mathsf{TM}} := \{\langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is length-dependent}\} .$$

   (a) Show that $\mathrm{LD}_{\mathsf{TM}}$ is undecidable by giving a mapping reduction from A_{TM} to $\mathrm{LD}_{\mathsf{TM}}$ (i.e., $A_{\mathsf{TM}} \leq_m \mathrm{LD}_{\mathsf{TM}}$).

   (b) Show that $\mathrm{LD}_{\mathsf{TM}}$ is not Turing-recognizable by giving a mapping reduction from $\overline{A_{\mathsf{TM}}}$ to $\mathrm{LD}_{\mathsf{TM}}$ (i.e., $\overline{A_{\mathsf{TM}}} \leq_m \mathrm{LD}_{\mathsf{TM}}$, or equivalently, $A_{\mathsf{TM}} \leq_m \overline{\mathrm{LD}_{\mathsf{TM}}}$).

   If you give a correct reduction, then you do not need to prove that it is correct. Getting either (a) or (b) correct is worth 70%. Getting both correct is worth 100%.

   Answer: There are several ways to reduce A_{TM} (respectively, $\overline{A_{\mathsf{TM}}}$) to $\mathrm{LD}_{\mathsf{TM}}$. Both assume wlog that all Turing machines (TMs) use as their input alphabet a set Σ that contains at least two symbols, say, $\mathbf{a}$ and $\mathbf{b}$. Also note the convention that $\langle \mathcal{O} \rangle$ denotes a string encoding the mathematical object(s) $\mathcal{O}$ in some reasonable way.

   (a) Let $f : \Sigma^* \to \Sigma^*$ be defined as follows:

   $f :=$ "On input $\langle M, w \rangle$ where M is a TM and w a string:

   i. Let $R :=$ 'On input x:

      A. If $x = \mathbf{a}$ then accept.

      B. Run M on input w (and do what M does).'

   ii. Output $\langle R \rangle$."

   Justification (optional): f is clearly computable. Let $\langle R \rangle$ be $f(\langle M, w \rangle)$. If $\langle M, w \rangle \in A_{\mathsf{TM}}$, then M accepts w; so then R accepts all strings, i.e., $L(R) = \Sigma^*$, which is length-dependent, so $\langle R \rangle \in \mathrm{LD}_{\mathsf{TM}}$. Conversely, if $\langle M, w \rangle \notin A_{\mathsf{TM}}$, then M does not accept w; so then $L(R) = \{\mathbf{a}\}$, which is not length-dependent, meaning $\langle R \rangle \notin \mathrm{LD}_{\mathsf{TM}}$. Thus f m-reduces A_{TM} to $\mathrm{LD}_{\mathsf{TM}}$.

(b) Let $g : \Sigma^* \to \Sigma^*$ be defined as follows:

$g :=$ "On input $\langle M, w\rangle$ where M is a TM and w a string:

i. Let $S :=$ 'On input x:

A. Run M on input w. // Note: If M loops on w, then S will loop on x.

B. If M rejects w, then reject.

C. If $x = \mathbf{a}$ then accept. // We get here iff M accepts w.

D. Else, reject.'

ii. Output $\langle S\rangle$."

Justification (optional): g is clearly computable. Let $\langle S\rangle$ be $g(\langle M, w\rangle)$. If $\langle M, w\rangle \in \overline{A_{\mathsf{TM}}}$, then M does not accept w; so then S accepts no strings, i.e., $L(S) = \emptyset$, which is length-dependent, so $\langle S\rangle \in \mathrm{LD}_{\mathsf{TM}}$. Conversely, if $\langle M, w\rangle \notin \overline{A_{\mathsf{TM}}}$, then M *does* accept w; so then $L(S) = \{\mathbf{a}\}$, which is not length-dependent, meaning $\langle S\rangle \notin \mathrm{LD}_{\mathsf{TM}}$. Thus g m-reduces $\overline{A_{\mathsf{TM}}}$ to $\mathrm{LD}_{\mathsf{TM}}$.

3. Let $G = (V, E)$ be a graph. An *almost-clique* in G is a set $C \subseteq V$ of vertices such that all pairs of distinct vertices in C are adjacent except for one pair. That is, an almost-clique is a clique except with exactly one edge missing.

Let ALMOST-CLIQUE be the following decision problem:

Instance: A graph G and an integer $K \geq 2$.
Question: Does there exist an almost-clique in G with K or more vertices?

ALMOST-CLIQUE is clearly in NP. Show that ALMOST-CLIQUE is NP-complete by giving a polynomial reduction from CLIQUE to ALMOST-CLIQUE.

[If your reduction is correct, you do not need to show that it is correct.]

Answer: Our reduction f takes as input a tuple $\langle G, K\rangle$ where G is a graph and K a natural number, and constructs a graph G' by starting with G then adding two new vertices u and v and new edges connecting each of u and v with all the vertices of G (but no edge connecting u with v). The reduction f then outputs $\langle G', K + 2\rangle$.

Justification (optional): f is clearly computable in polynomial time. For correctness, first assume $\langle G, K\rangle \in$ CLIQUE. Then G has a clique $C \subseteq G.V$ of size K. But then $C \cup \{u, v\}$ is an almost-clique of size $K + 2$ in G'—the unique edge missing being (u, v). Conversely, suppose that G' has an almost-clique $A \subseteq G'.V = G.V \cup \{u, v\}$ of size $\geq K + 2$. Let $x, y \in A$ be unique such that $(x, y) \notin G'.E$. There are two cases: (1) if $\{x, y\} \cap \{u, v\} \neq \emptyset$, then $A \setminus \{x, y\}$ is a clique in G of size $\geq K$ (and a moment's reflection convinces one that $A \setminus \{x, y\}$ can only contain vertices in G); (2) otherwise, $A \setminus \{u, v, x\}$ is a clique in G of size $\geq K$ (and a moment's reflection convinces one that $A \setminus \{u, v, x\}$ must contain at least K vertices). Thus in either case, $\langle G, K\rangle \in$ CLIQUE.

There are other correct solutions to this problem.