Fall 2024 CSE Qualifying Exam—Theory (551)

1. Let $\Sigma := \{a, b, c\}$. For any language string $w \in \Sigma^*$, define $\mathrm{MTE}(w)$ to be the set of all strings obtained from w by moving one of its symbols to the end of the string. (MTE stands for "Move To End"). So for example,

$$\mathrm{MTE}(abcab) = \{bcaba, acabb, ababc, abcba, abcab\}$$
$$\mathrm{MTE}(\varepsilon) = \emptyset$$

(Note that $\mathrm{MTE}(w)$ always includes w because moving its last symbol to the end does not change the string.)

For any $L \subseteq \Sigma^*$, define

$$\mathrm{MTE}(L) := \bigcup_{w \in L} \mathrm{MTE}(w) .$$

So $\mathrm{MTE}(L)$ is the set of all strings obtained from strings in L by moving any symbol in the string to its end.

Show by construction that if L is regular, then $\mathrm{MTE}(L)$ is regular. If your construction works, you need not justify it. [Hint: given an n-state DFA for L, there is an NFA for $\mathrm{MTE}(L)$ with roughly $n + |\Sigma|n = 4n$ states.]

Answer: Let $D = \langle Q, \Sigma, \delta, s, F\rangle$ be an n-state DFA such that $L = L(D)$. We build a $(4n + 3)$-state NFA N such that $L(N) = \mathrm{MTE}(L)$ as follows:

- N starts with the disjoint union of D with three new disjoint copies of D that are D_a, D_b, D_c, plus three new states $\{t_a, t_b, t_c\}$. For any state $q \in Q$, we let q_a, q_b, q_c denote the corresponding copies of q in D_a, D_b, D_c, respectively.
- The start state of N is s, the start state of the D copy.
- The set of accepting states of N is $\{t_a, t_b, t_c\}$. All states in D, D_a, D_b, D_c are made rejecting.
- For every transition $q \xrightarrow{x} r$ of D, where $x \in \Sigma$ and $q, r \in Q$, add the ε-transition $q \xrightarrow{\varepsilon} r_x$ to N. (This transition allows you to pretend you read a symbol x on the input without actually reading it, but you remember x in the state r_x.)
- For every state $q \in F$ and $x \in \Sigma$, add the transition $q_x \xrightarrow{x} t_x$. (This allows you to read the same symbol at the end that you skipped earlier.)

2. We assume all languages are over the binary alphabet $\{0, 1\}$ for this problem.

   Let f be a function that, for every enumerator E and natural number $n \geq 0$ as inputs, outputs the number of strings in $L(E)$ of length n, i.e.,

   $$f(\langle E, n \rangle) = |L(E) \cap \{0, 1\}^n| \; .$$

   Show that no such f can be computable.

   Answer: Fix an enumerator E for some enumerable but undecidable language, e.g., A_{TM}. Assuming f is computable, the following procedure decides $L(E)$ (contradiction):

   $D :=$ "On input $w \in \{0, 1\}^*$:

   (a) Let $n := |w|$. (The length of w is n.)
   (b) Compute $s := f(\langle E, n \rangle)$. (Then $s = |L(E) \cap \{0, 1\}^n|$ by assumption.)
   (c) Run E until it prints s many distinct strings of length n.
   (d) If w is one of the strings printed by E in the last step, then accept; else reject."

   Explanation (optional): Step (c) eventually finishes, because every string in $L(E)$ is eventually printed by E, and there are s many of these strings of length n. If w is one of these strings, then clearly $w \in L(E)$, so accepting is correct. If w is not one of these strings, then $w \notin L(E)$ (for otherwise $L(E)$ has more than s many strings of length n), and so rejecting is correct.

3. The VC-OVERLAP problem is

   Instance: A graph G and a natural number $k \leq |G.V|$.
   Question: Is there a vertex cover C of G of size $\leq k$ such that at least one edge of G has *both* its endpoints of C?

   VC-OVERLAP is clearly in NP. Show that VC-OVERLAP is NP-hard by giving a polynomial reduction to VC-OVERLAP from some well-known NP-complete problem. If your reduction is correct, you need not justify it.

   Answer: We reduce VC to VC-OVERLAP. Given a graph G and number k (i.e., an instance of VC), construct a graph G' that is obtained from G by adding three new vertices u, v, w and connecting them all by edges, forming a 3-clique. We then return $\langle G', k' \rangle$, where $k' := k + 2$.

   Explanation (optional): This reduction is clearly polynomial-time. Given $\langle G, k \rangle$ we argue both directions of the if-and-only-if:

   ⇒: If G has a v.c. C of size $\leq k$, then $C' := C \cup \{u, v\}$ is a v.c. of G' of size $k' = k + 2$, and the edge (u, v) has both endpoints in C'.

$\Leftarrow$: If G' has a v.c. C' of size $k+2$ (that contains both endpoints of some edge), then let $C := C' \cap G.V$. Then C is clearly a v.c. of G, and C has size $\leq k$, because C' must include at least 2 elements of $\{u, v, w\}$ to cover the edges in that triangle.