CS5371 Theory of Computation

Homework 5 (Solution)

1. Let S be a set and let C be a collection of subsets of S. A set S', with $S' \subseteq S$, is a called a *hitting set* for C if every subset in C contains at least an element in S'. Let *HITSET* be the language

$$\{\langle C, k\rangle \mid C \text{ has a hitting set of size } k\}.$$

Prove that *HITSET* is NP-complete.

Answer: It is easy to check that *HITSET* is in NP (why?). To see why *HITSET* is NP-complete, we observe that we can reduce *VERTEX-COVER* to *HITSET*: Given a graph $G = (V, E)$, we set $S = V$ and $C = E$, then immediately we have G has a vertex cover of size k if and only if C has a hitting set of size k. As *VERTEX-COVER* is NP-complete, the reduction (obviously) takes polynomial time, and *HITSET* is in NP, we have proved that *HITSET* is NP-complete.

2. Let U be the language

$$\{\langle M, x, \#^t\rangle \mid \text{TM } M \text{ accepts input } x \text{ within } t \text{ steps on at least one branch}\}.$$

Show that U is NP-complete. (For this problem, you are required to prove it without using reduction from any known NP-complete problems.)

Answer: To see why U is in NP, we observe that there is an NTM N that recognizes U in polynomial time, such that for any $\langle M, x, \#^t\rangle \in U$, N guesses the t choices of the branch for M to accept x within t steps.

To see why U is NP-complete, let A be any language in NP. Since A is in NP, there exists an NTM N_A that accepts any string y in A within $|y|^k$ steps, for some k. Then, we can reduce A to U as follows: Given any input string y, we set $M = N_A$, $x = y$ and $t = k$; immediately, we have y in A if and only if $\langle M, x, \#^t\rangle$ in U. As the reduction is polynomial time, we have shown that any language in NP is polynomial time reducible to U. As U is in NP, so by definition U is NP-complete.

Further Question: Will the above proof still be okay when U is replaced by the following language U':

$$\{\langle M, x, t\rangle \mid \text{TM } M \text{ accepts input } x \text{ within } t \text{ steps on at least one branch}\}.$$

3. We say a language A is in coNP if its complement, $\overline{A}$, is in NP. We call a regular expression *star-free* if it does not contain any star operations. Let EQ_{SF_RFX} be the language

$$\{\langle R, S\rangle \mid R, S \text{ are equivalent star-free regular expressions}\}.$$

Show that EQ_{SF_RFX} is in coNP. Why does your argument fail for general regular expressions?

Answer: To show that $\overline{\text{EQ}_{SF\_RFX}}$ is in NP, we observe that a string x is in $\overline{\text{EQ}_{SF\_RFX}}$ if and only if it is one of the following forms:

   (a) x does not represent a valid encoding of two regular expressions;

(b) x is of the correct form $\langle R, S \rangle$, but either R, or S, or both are not star-free;

(c) x is of the correct form $\langle R, S \rangle$, R and S are both star-free, but $L(R) \neq L(S)$.

If x is in the first and the second form, x can be accepted by a DTM easily in polynomial time. If x is in the third form, there exists a string y that is in exactly one of the $L(R)$ or $L(S)$. As R and S are star-free, the length of y must be polynomial in the size of $|R| + |S|$ (why?). Thus, there is an NTM N that guesses such a string y in polynomial time whenever x is of the third form (but will never find such a string y when $L(R) = L(S)$). Thus, there exists an NTM that recognizes $\overline{\text{EQ}_{SF\_RFX}}$ in polynomial time. This completes the proof.

4. (Choose either Q4 or Q5.) Show that the following problem is NP-complete. You are given a set of states $Q = \{q_0, q_1, \ldots, q_\ell\}$ and a collection of pairs $\Pi = \{(s_1, r_1), \ldots, (s_k, r_k)\}$ where the s_i are distinct strings over $\Sigma = \{\texttt{0}, \texttt{1}\}$, and the r_i are (not necessarily distinct) members of Q. Determine whether a DFA $M = (Q, \Sigma, \delta, q_0, F)$ exists where $\delta(q_0, s_i) = r_i$ for each i. Here, the notation $\delta(q, s)$ stands for the state that M enters after reading s, starting at state q. (Note that F is irrelevant here).

   Answer: (sketch) To show that the above problem is in NP, we observe that an NTM can guess the correct DFA satisfying the constraints Q and Π in polynomial time if and only if such a DFA exists.

   To show that the above problem is NP-complete, we reduce the NP-complete problem *3SAT* to it: Given a 3cnf-formula F, say, $F = \bigwedge_{i=1}^{k} C_i$ and $C_i = (x_i \vee y_i \vee z_i)$, we construct the following constraints Q and Π:

   (a) $Q = \{q_T, q_F, q_1, q_2\}$;

   (b) Create a pair (ε, q_F) in Π to enforce q_F to be the start state;

   (c) For each variable x in F, create the following two pairs in Π: $(x\overline{x}, q_T)$ and $(\overline{x}x, q_T)$;

   (d) For each clause C_i in F, create a pair $(x_i y_i z_i, q_T)$ in Π;

   (e) For each variable x in F, create the following two pairs in Π: $(x\#_x, q_1)$ and $(\overline{x}\#_x, q_2)$, where these two pairs enforce that after reading x and after reading $\overline{x}$, DFA must be in different states;

   (f) Pick any variable x in F. Then, for each variable y, create the following three pairs in Π: $(x\overline{x}y, q_T)$, $(x\#_x y, q_1)$, $(\overline{x}\#_x y, q_2)$.

   We claim that F is satisfiable if and only if there exists a DFA satisfying the constraints Q and C. (The proof of the claim is left as a further exercise.) As the reduction takes polynomial time, this completes the proof.

5. (Choose either Q4 or Q5.) Consider the algorithm *MINIMIZE*, which takes a DFA M as input and outputs DFA M'.

   MINIMIZE = "On input $\langle M \rangle$, where $M = (Q, \Sigma, \delta, q_0, A)$ is a DFA:

   1. Remove all states of M that are unreachable from the start state.
   2. Construct the following undirected graph G whose nodes are the states of M.
   3. Place an edge in G connecting every accept state with every nonaccept state. Add additional edges as follows.
   4. Repeat until no new edges are added to G:

5. For every pair of distinct states q and r of M and every $a \in \Sigma$:
6. Add the edge (q, r) to G if $(\delta(q, a), \delta(r, a))$ is an edge of G.
7. For each state q, let $[q]$ be the collection of states

$$[q] = \{r \in Q \mid \text{ no edge joins } q \text{ and } r \text{ in } G\}.$$

8. Form a new DFA $M' = (Q', \Sigma, \delta', q_0', A')$ where
   - $Q' = \{[q] \mid q \in Q\}$, (if $[q] = [r]$, only one of them is in Q'),
   - $\delta'([q], a) = [\delta(q, a)]$, for every $q \in Q$ and $a \in \Sigma$,
   - $q_0' = [q_0]$, and
   - $A' = \{[q] \mid q \in A\}$.
9. Output $\langle M' \rangle$."

A. Show that M and M' are equivalent.

B. Show that M' is minimal—that is, no DFA with fewer states recognizes the same language. You may use the Myhill-Nerode Theorem.

C. Show that *MINIMIZE* operates in polynomial time.

Answer:

A. Firstly, if a string $x = x_1 x_2 \cdots x_t$ of length t is accepted by M, there exists a sequence of states, $q_{i_0}, q_{i_1}, \ldots, q_{i_t}$ such that $q_{i_0} = q_0$, $q_{i_j} = \delta(q_{i_{j-1}}, x_j)$, and $q_{i_t} \in F$. This implies that x can be accepted by M' based on the the sequence of states $[q_{i_0}], [q_{i_1}], \ldots, [q_{i_t}]$ such that $[q_{i_0}] = [q_0]$, $[q_{i_j}] = \delta'([q_{i_{j-1}}], x_j)$, and $[q_{i_t}] \in F'$. Thus, $L(M) \subseteq L(M')$.

   Secondly, if a string $y = y_1 y_2 \cdots x_t$ of length t is accepted by M', let $[q_{i_0}], [q_{i_1}], \ldots, [q_{i_t}]$ be the set of states such that $[q_{i_0}] = [q_0]$, $[q_{i_j}] = \delta'([q_{i_{j-1}}], x_j)$, and $[q_{i_t}] \in F'$. By induction, we can show that when y is input to M, the corresponding sequence of states visited by M, say $r_0, r_1, r_2, \ldots, r_t$, will satisfy $r_0 = q_0$, and $r_j \in [q_{i_j}]$ for all j. Now, as $[q_{i_t}] \in F'$, we know that $[q_{i_t}] \subseteq F$ (why?). Thus, $r_t \in F$, so that $L(M') \subseteq L(M)$.

   In conclusion, $L(M) = L(M')$, so that M and M' are equivalent.

B. Let $\delta(q_0, x)$ denote the state of M after reading x when M starts from q_0. By induction, we can show that for two distinct states q and r in the undirected graph G, q and r are connected by an edge if and only if there exists strings x and y such that $\delta(q_0, x) = q$, $\delta(q_0, y) = r$, and x, y are distinguishable by $L(M)$.

   Based on the above result, $[q]$ will store the all states q' such that for all x, y with $\delta(q_0, x) = q$ and $\delta(q_0, y) = q'$, x and y are indistinguishable. Also, for any $q' \in [q]$, we observe that $[q'] = [q]$ (why?).

   In other words, the distinct set of states $[q]$ in Q' forms a partition of Q. By picking one string x with $\delta(q_0, x) \in q$ for each distinct $[q]$, the resulting $|Q'|$ strings are pairwise distinguishable by $L(M)$ (why?). By Myhill-Nerode theorem, any DFA recognizing $L(M)$ must have at least $|Q'|$ states.

   As M' has $|Q'|$ states and $L(M) = L(M')$, M' is a minimal.

C. Let $|Q| = n$. Step 1 takes at most $O(n^3 + n^2|\Sigma|)$ time, by using the brute force connectivity algorithm. For Step 3, it takes $O(n^2)$ time. For Step 4, it repeats Steps 5 and 6 for at most $O(n^2)$ times, each repetition takes at most $O(n^2|\Sigma|)$ time. For Step 7, it is done in $O(n^2)$ time. For Step 8, we first check if $[q] = [r]$ for each pair of q and r, which requires $O(n^3)$ time; this naturally gives a partition of Q's states, and the partition can be stored in a table; then, we construct the final DFA M', which takes an additional $O(n^2|\Sigma|)$ time.

In total, the time required for constructing M' is $O(n^4|\Sigma|)$, which is polynomial in the length of the input $\langle M \rangle$.