Theory of Computation

Midterm Examination on November 28, 2017
Fall Semester, 2017

Problem 1 (20 points)

1. (10 points) Let $\mathcal{C}$ be a complexity class that is closed under reductions. Prove that if L is $\mathcal{C}$-complete and L is reducible to $L' \in \mathcal{C}$, then L' is also $\mathcal{C}$-complete.

2. (10 points) Prove that L is NP-complete if and only if its complement $\bar{L} = \Sigma^* - L$ is coNP-complete.

Proof:

1. Since L is $\mathcal{C}$-complete, any $L'' \in \mathcal{C}$ can be reduced to L which is reducible to L'. By Proposition 28 (see p. 285 of the slides), L'' is reducible to L'. So L' is $\mathcal{C}$-complete.

2. See pp. 457–458 of the slides.

■

Problem 2 (20 points) Recall that the **depth** of a gate g is the length of the longest path in a circuit from g to an input gate. A circuit is **leveled** if every input of a gate in depth k comes from one in depth $k-1$. LEVELED CIRCUIT asks if a leveled circuit is satisfiable. Prove that LEVELED CIRCUIT is NP-Complete. (You do not have to show it is in NP.)

Proof: We can obtain a leveled circuit from any circuit C by increasing the number of gates by a polynomial factor. This holds for the input gates. Inductively, suppose all gates of depth $k-1$ have length $k-1$ for the shortest paths to the input gates. Now consider gates of depth k. Pick any gate g with a shorter shortest path to the input gates, say length $l < k$. Insert a series of $k-l$ $\vee$ gates on the edge between g and its predecessor gate on one such path. Note that $k - l = O(|\,C\,|)$. Note that these $\vee$ gates have their two identical inputs. So they act as the identity function. The new circuit has size $O(|\,C\,|^2)$. Finally, recall that CIRCUIT SAT is NP-complete by Cook's Theorem. ■

Problem 3 (20 points) Let $G(V, E)$ be a directed graph with vertices V and edges E. HAMILTONIAN CYCLE asks if there is a cycle through a graph G which visits each vertex exactly once. It is known that HAMILTONIAN CYCLE is NP-complete. BIGCYCLE asks if G has a cycle of length equal or larger than $|V|/2$. Prove that BIGCYCLE is NP-complete. (You need to show BIGCYCLE is in NP.)

Proof: We first prove that BIGCYCLE is in NP. Given a graph G, one can guess a cycle and accept G if the length of the cycle is equal or larger than $|V|/2$. It can be done in polynomial time.

Now we reduce HAMILTONIAN CYCLE to BIGCYCLE. Let N be an NTM which decides BIGCYCLE. Construct an NTM M which decides HAMILTONIAN CYCLE as follows:

1: On input $G(V, E)$ with $|V|$.

2: Add exactly $|V|$ isolated vertices to G to obtain G'.

3: Run N(G').

4: If N accepts, halt and accept.

5: Otherwise, halt and reject.

Hence, $G \in$ HAMILTONIAN CYCLE if and only if $G' \in$ BIGCYCLE. M runs in polynomial time. This completes the proof. ■

Problem 4 (20 points) Show that VALIDITY is coNP-complete.

Proof: First, we construct a TM which verifies the input x and accepts if $x \in$ VALIDITY. It takes polynomial time. So VALIDITY $\in$ coNP. Now we proceed to show that L can be reduced to VALIDITY for all $L \in$ coNP. It is known that SAT is NP-complete. By Proposition 54 (see p. 457 of the slides), $\overline{\text{SAT}}$ is coNP-complete. So it suffices to show that $\overline{\text{SAT}}$ can be reduced to VALIDITY. Let N be an NTM which decides VALIDITY. Construct an NTM M which decides $\overline{\text{SAT}}$ as follows:

1: On input x, let $x' = \neg x$.

2: Run N(x')

3: If N accepts, halt and accept.

4: Otherwise, halt and reject.

M clearly runs in polynomial time. Hence, VALIDITY is coNP-complete. ■

Problem 5 (20 points) Let N be a polynomial-time Monte Carlo algorithm that uses $O(\log n)$ random bits, where n is the input length. (It can give false-negative answers with probability less than 50%. It has no false positives, however.) Turn it into a polynomial-time deterministic algorithm that makes no errors.

Proof: Try all $2^{O(\log n)}$ possible coin flips. Accept the input if and only if there is at least one computation path (i.e., one sequence of $O(\log n)$ coin flips) that ends up with "yes." Reject the input otherwise. As there are only polynomially many coin flips, the total running time remains polynomial. This algorithm is correct because the input is a yes instance if at least 50% of the coin flips lead to "yes" and the input is a no instance if none of the coin flips lead to "no." And there are no other cases to consider. ■