CS 121: Introduction to Theoretical Computer Science Fall 2019

Section 8

Prof. Boaz Barak

0.1 Problems

1. Give a simple argument for why $\mathsf{NP} \subseteq \mathsf{EXP}$ — consider how you can use the existence of the verifier G.
2. For each of the following, say whether the problem is in P, NP, is undecidable, or whether we don't know.
   (a) Given an integer x, determine if x has a prime factor that is at most k.
   (b) Given an undirected graph graph, determine whether it is possible to partition its vertices into two sets, with at least k edges crossing between sets.
   (c) Given a program Q, an input x, and a string 1^t, determine whether Q halts on x within t steps.
3. Define $F \in \mathsf{coNP}$ iff $\overline{F} \in \mathsf{NP}$, where $\overline{F}$ denotes the negation of the output of F (for example, if $F(00) = 1$, then $\overline{F}(00) = 0$). Prove that if $\mathsf{P} = \mathsf{NP}$, then $\mathsf{coNP} = \mathsf{NP}$.
4. Let $V : \{0,1\}^* \to \{0,1\}$ be defined as taking two inputs x, w such that there exists $a, b \in \mathbb{N}$ such that $w \in \{0,1\}^{a|x|^b}$. $V \in P$. Prove that $V \in TIME(|x|^c)$ for some c.

Solution 1: For every possible certificate w, we can check whether $G(x, w) = 1$. We need to try all possible w of length an^b (there are 2^{an^b} of these), and evaluating G can be done in polynomial time, so we make take most an exponential number of steps.

Solution 2: (a) in NP (the certificate is a prime factor that is at most k) (b) in NP (the certificate is the partition) (c) in P (just stimulate the program).

Solution 3: For every $F \in \mathsf{NP}$, we have a $NAND-TM$ program W which computes F in polynomial time. Thus $\overline{W}$ (W which negates its output) computes $\overline{F}$ in polynomial time. Since this holds for every $F \in \mathsf{NP}$, we have $\mathsf{coNP} \subset \mathsf{P} = \mathsf{NP}$. But $P \subset \mathsf{coNP}$, so we have equality.

Solution 4: Since $V \in P$ there exists some c such that $V \in TIME(n^c)$. We can rewrite this as $V \in TIME((|x| + a|x|^b)^c)$. $(|x|+a|x|^b)^c$ is polynomial in $|x|$, so in particular, there exists some c' such that for large enough $|x|$, $(|x|+a|x|^b)^c \leq |x|^{c'}$, so $V \in TIME(|x|^{x'})$.

0.2 Problems

1. Given an undirected graph $G = (V, E)$, a clique is a subset $C \subseteq V$ such that $(v_1, v_2) \in E$ for all $v_1, v_2 \in C$. Consider the function $CLIQUE(G, k) = 1$ iff G has a clique of size k, and 0 otherwise. Show that $3SAT \leq_{\mathrm{p}} CLIQUE$, and that $CLIQUE$ is NP-complete.
2. Define $F \in \mathsf{coNP}$ iff $\overline{F} \in \mathsf{NP}$, where $\overline{F}$ denotes the negation of the output of F (for example, if $F(00) = 1$, then $\overline{F}(00) = 0$). Consider the following function $TAUTOLOGY$: if ϕ is a 3DNF formula (clauses of three 'and'ed variables, 'or'ed together), $TAUTOLOGY(\phi) = 1$ iff for all assignments x of the variables of ϕ, we have $\phi(x) = 1$. Otherwise $TAUTOLOGY(\phi) = 0$. Prove that $TAUTOLOGY$ is coNP-complete.

   We say $TAUTOLOGY$ is coNPcomplete if $TAUTOLOGY \in \mathsf{coNP}$ and $\forall F \in \mathsf{coNP}, TAUTOLOGY \leq_p F$. Hint: $3SAT$ is NP-complete. Try to relate the $3SAT$ problem to $TAUTOLOGY$.
3. Given n sets $S_1, S_2, \ldots, S_n$ such that

   $$\bigcup_{i=1}^{n} S_i = A$$

   the set cover of size k over these sets is a collection C of k of these sets such that

   $$\bigcup_{i \in C} S_i = A$$

   Given a collection of sets and an integer k, SET-COVER returns if there exists a valid set cover of a most size k over the given collection of sets. Prove that SET-COVER is NP-complete.

Solution 1: Suppose we're given a $3SAT$ formula $\varphi = \varphi_1 \wedge \cdots \wedge \varphi_l$, where each φ_i is a clause. We construct a graph G as follows: for every clause c and variable v in c, we create a vertex (c, v) (so we end up with $3l$ clauses). For example, if clause 1 is $(x_1 \vee \neg x_2 \vee x_3)$, we create the vertices $(1, x_1)$, $(1, \neg x_2)$ and $(1, x_3)$. For every two vertices $(c, v), (c', v')$, we add an edge between these two vertices iff $c \neq c'$ and v is not the negation of v'. Clearly we can do all of these steps in polynomial time. We claim that $CLIQUE(G, l) = 1$ iff φ is satisfiable.

First suppose that φ is satisfiable with assignment x. Construct a clique C as follows: for each clause φ_i, look for a variable which is set to 1 via the assignment x, and add (i, v) to C. For example, if our clause is $\varphi_1 = x_1 \vee \neg x_2 \vee x_3$, and $x = 000$, we can add the vertex $(1, \neg x_2)$. Clearly $|C| = l$. Moreover, C is a clique, because there is only not an edge between $(c, v), (c', v')$ if $c = c'$ or v is the negation of v'. The first case cannot happen by construction. The second case cannot happen because if v evaluates to 1, then $\neg v$ cannot also evaluate to 1.

For the other direction, suppose we have a clique C of size l. Then our clique is of the form $(1, v_1), (2, v_2), \ldots, (l, v_l)$. We create an assignment x which satisfies φ by setting x such that each v_i evaluates to 1. For example, if $v_1 = x_5$, we set $x_5 = 1$, and if $v_2 = \neg x_3$, we set x_3 to 0. Notice that we will never be in the case where we set x_j to both 1 and 0; this would imply that we have edges (c, v), $(c', v') \in C$ such that $v = \neg v'$, which contradicts our construction of G. Moreover, x constructed in this way satisfies φ, because for each i, φ_i evaluates to 1, because at least one variable in clause i evaluates to 1.

Lastly, notice that $CLIQUE$ is in NP, because we can always "guess" a clique of size k and determine whether this guess is indeed a clique in polynomial time (just check all possible pairs of edges).

Solution 2: We will prove that $TAUTOLOGY$ is $coNP$-complete by proving that for any $F \in coNP$, $F \leq_p TAUTOLOGY$. Let $F \in$ coNP. Then $\overline{F} \in$ NP, so for every $x \in \{0,1\}^*$ we can in polynomial time compute a $3SAT$ formula ϕ_x for $\overline{F}$ such that $\overline{F}(x) = 1$ iff ϕ_x has a satisfying assignment (i.e. there exists an x' such that $\phi_x(x') = 1$). But this means that $F(x) = 1$ iff ϕ_x has no satisfying assignment, i.e. $\overline{F}(x) = 1$ iff $\overline{\phi}_x$ is equal to 1 for every assignment of variables. But $\overline{\phi_x}$ is a $3DNF$, so this is exactly the problem $TAUTOLOGY$.

$$TAUTOLOGY(\overline{\phi_x}) = F(x)$$

Thus we have given a reduction $F \leq_{\mathrm{p}} TAUTOLOGY$.

Now we show that $TAUTOLOGY \in$ coNP. Let $G(x)$ be a function taking in x, a 3DNF, that returns 1 if there exists a non-satisfying solution for x. We see this is in NP, because the solution is a binary string and the verifier just runs through the clauses, checking them in linear time. We see that $\overline{G}(x) = TAUTOLOGY(x)$ exactly because there being no non-satisfying solutions is exactly the condition for every solution satisfying. Thus $TAUTOLOGY$ is coNP-complete.

Solution 3: We know that $SET-COVER$ is in NP because we can use a set of sets as the certificate, and can verify by checking that each element is a member of at least one set.

We now reduce from $VERTEXCOVER$. Suppose we have an instance of $VERTEXCOVER$ (so a graph and a number k). Label the edges in the graph from 1 to m. For each vertex v create a set S_v that is composed of the edges of which v is a part. This transformation is polynomial time since it must run over the edges and then runs over the vertices (going over each edge twice more).

First suppose that there is a valid $VERTEXCOVER$. I claim that the sets associated with the vertices in the $VERTEXCOVER$ (call these vertices V') form a valid $SET - COVER$. For any number associated with an edge $e = (u, v)$, either $u \in V'$ or $v \in V'$, which implies the number associated with e is either in S_u or S_v.

Now suppose there is a valid $SET - COVER$ of size k. I claim the vertices associated with the sets in this $SET - COVER$ (denote this set of vertices V') form a $VERTEXCOVER$. Suppose towards a contradiction there was an edge $(u, v) \in E$ such that $u, v \notin V'$. This implies the number associated with that edge would not be in the $SET - COVER$, so it would not be a $SET - COVER$, hence a contradictoin.

0.3 Problems

1. Prove that for $V \in P$ $STARTSWITH_V$ is in NP.
2. Using the optimization and search-to-decision results, prove that for any $F \in P$, we can compute $OPTARG(x, 1^m) = \text{argmax}_{y \in \{0,1\}^m} F(x, y)$ (again identifying the output of F with a natural number via the binary representation).

Solution 1: The certificate is the remaining $a|x|^b - l$ bits. The verifier is exactly V.

Solution 2: By the optimization result, we know that given $F \in P$ we can compute $OPT(x, 1^m) = \max_{y \in \{0,1\}^m} F(x, y)$ in polynomial time. Let $k_{x,m}$ denote $OPT(x, 1^m)$. Then we see we can compute the function $G(x, y)$ which returns 1 if and only if $F(x, y) = k_{x,m}$ in polynomial time (because $F \in P$). Applying the search to decision on result on the polynomial time algorithm for G thus yields a solution y such that $F(x, y) = k_{x,m}$.