CS 332 - Homework 5 - short answers

Due: Thursday, April 18 at 12 noon

At the top of your homework paper write your name, the class number (CS 332) and also the homework assignment number (HW 5), and write the number of the problem at the top of your problem solution for each problem.

Reading: Chapter 7: Pages 292-311.

PROBLEMS:

1. i. Show that P is closed under intersection.
   (This means that if languages J and L are both in P, then so is J ∩ L.)

   Proof:
   Let \( A \) be an algorithm for deciding J which runs in time \( O(n^j) \) for some fixed \( j \), and let \( B \) be an algorithm for deciding L which runs in time \( O(n^l) \).

   We define an algorithm \( C \) be an algorithm for deciding J ∩ L
   On input \( x \) of length \( n \), \( C(x) \) carries out the following steps.
   1. Compute \( A(x) \). If \( A(x) \) rejects then halt and reject, else go on
   2. Compute \( B(x) \). If \( B(x) \) rejects then halt and reject, else accept.

   The algorithm \( C \) clearly accepts J ∩ L, and does so in time \( O(n^j) + O(n^l) \) which is \( O(n^{\max(j,l)}) \) and so in P.

   ii. Show that JL (the concatenation of J and L) is in P.
   (Here J and L are in P as in part (i), and JL = \( \{ xy | x \in J \text{ and } y \in L \} \). )
   Here you should calculate the complexity of your P-time algorithm for JL.

   Proof: Let \( A \) be an algorithm for deciding J which runs in time \( O(n^j) \), and let \( B \) be an algorithm for deciding L which runs in time \( O(n^l) \).

   We define an algorithm \( C \) be an algorithm for deciding JL
   On input \( x = x_1, x_2, x_3, ..., x_n \), \( C(x) \) carries out the following steps.
   (Here we briefly describe how \( C \) computes on input \( x \) rather than the full detailed pseudocode.)

   Split \( x \) into \( n \) pairs of string \( (y,z) \) where \( y = x_1, ..., x_i \) and \( z = x_{i+1}, ..., x_n \) for each \( i=1,...,n \).
   (There are \( n \) such pairs \( (y,z) \), you get the idea).

   For each such \( (y,z) \) pair, use \( A \) on input \( y \) to test if \( y \) is in J and \( B \) on input \( z \) to test if \( z \) is in L. If you ever find that both tests accept for some \( y \) and \( z \) the accept \( x \), if not the reject \( x \).
It’s not hard to check that $C$ run on $x$ in polynomial time in $nO(n^4)O(n^4)$ which a polynomial in $|x| = n$, though we omit the this time calculation details of this here.

iii. Show that NP is closed under union.

Proof (sketch): Given that $J$ and $L$ in NP, we know that there is a verifier $V(x,c)$ for $J$ a verifier $V'(y,d)$ for $L$. We need to use $V$ and $V'$ to define a verifier $W$ for $J \cup L$.

Algorithm $W$ takes as input $(x,y,c,d)$ where $x$ is an instance of problem $J$, $y$ is an instance of problem $L$, $c$ is a certificate for problem $J$, and $d$ is an instance for problem $L$.

We compute $W(x,y,c,d)$ by running $V(x,c)$ until it halts, and then running $V'(y,d)$ until it halts. It accepts if either of $V$ or $V'$ accepts, otherwise $W$ reject.

Clearly $W$ decides $J \cup L$ and does so in polynomial time since both $V$ and $V'$ run in polynomial time, and the sum of two polynomials is a polynomial.

2. The Graph accessibility Problem (GAP) = \{(G,s,t) | G is a directed graph where $s$ and $t$ are two vertices in $G$ and there exists a path from $s$ to $t$ in $G$\}.

Show that gap is in P. You need not use a TM just give a pseudo-code algorithm. You should do this using an algorithm of your choosing but need to do a detailed analysis of your algorithm to show its polynomial time bound.

Answer (very brief): The correct answer is to first give a pseudo-code algorithm such as depth-first search or bfs of a graph $G$ to test if $s$ and $t$ are connected in $G$. (This was done in cs 330.) Now analyze your graph to show that it runs in P. (The algorithm should be shown to be $O(n^2)$ or even $O(n)$.)

3. Note: Some of the definitions and ideas for this problem can be found on page 299 of our textbook.

i. Give an example of a Boolean formula $F$ which contains at least 2 different variables, and where $F$ is satisfiable and its negation ($\text{not}F$) is not satisfiable.

Answer: One such $F$ is $F(x,y) = (x \lor \overline{x}) \lor (y)$

ii. Give an example of a Boolean formula $F$ which contains at least 3 different variables, and where $F$ is satisfiable and has exactly 5 satisfying truth assignments.

Answer: One such $F$ is $F(x,y,z) = (\overline{x} \land y \land z) \lor (x \land y \land z) \lor (x \land \overline{y} \land z) \lor (x \land y \land \overline{z}) \lor (\overline{x} \land \overline{y} \land z)$.

4. Show that the Hamiltonian path problem (which was discussed in lab section) is in NP.

Answer: Here is brief sketch of a verifier $V$ for the HP problem showing it is in NP. We went over several verifiers for NP problems in class so not all details are given here.

The input to $V$ is a directed graph $G$ and two vertices of $G$, $s$ and $t$, and a certificate $p$ which is a candidate for a Hamiltonian path from $s$ to $t$ in $G$.

The verifier $V$’s computation is to check if $p$ is a correct Hamiltonian Path for $G$, in which case it accepts input $(G,s,t,p)$, else it rejects on this input.
5. Page 323, problem 7.12
   Answer: Here is brief sketch of a verifier for the ISO (graph isomorphism) problem showing it is in NP. We went over this idea of the verifier in class so not all details are given here.
   The input to $V$ is a pair of graphs $(G,H)$, the certificate $f$ for $V$ is a function. $V(G,H,f)$ checks if $f$ is a 1-1 and onto function from the vertices of $G$ to those of $H$ which preserves the edges of $G$ and $H$.

6. (Just so you won’t forget about the first 1/2 of the class. No partial credit, no need to justify your answer.)
   Say whether each of i. - v. is True or False.
   i. Any language $J$ with $\text{HALT} - J$ finite is undecidable. ($\text{HALT} = \text{the halting problem}$.)
   Answer: F. Let $J = \text{HALT} \cup$ some finite set from a different alphabet than $\text{HALT}$'s alphabet.
   ii. If $A$ is recognizable and $B \subseteq A$ then $B$ is recognizable.
   Answer: F
   iii. The disjoint union of a finite set and an infinite set is always infinite.
   Answer: T
   iv. If $J$ is reducible to $K$ and $K$ is recognizable then $J$ is also recognizable.
   Answer: T. This requires a short proof.
   v. If $L$ is any undecidable problem, then any language $J$ where $J - L$ is finite is also undecidable.
   Answer: F. Let $J$ be any finite set.