University of Nevada, Las Vegas Computer Science 456/656 Fall 2023

Answers to Assignment 5: Due Sunday October 22, 2023

1. True or False. T = true, F = false, and O = open, meaning that the answer is not known science at this time.

   (a) **T** Suppose L is a binary language. Suppose there is a number k such that, for every $w \in L$ there is an $O(n^k)$ time proof that $w \in L$, where $n = |w|$. Then L is $\mathcal{NP}$.

   (b) **T** SAT is known to be $\mathcal{NP}$–complete.

   (c) **F** 2-SAT is known to be $\mathcal{NP}$–complete.

   (d) **T** 3-SAT is known to be $\mathcal{NP}$–complete.

   (e) **T** 4-SAT is known to be $\mathcal{NP}$–complete.

   (f) **F** The game RUSH HOUR is known to be $\mathcal{NP}$–complete.

   (g) **T** The traveling salesman problem is known to be $\mathcal{NP}$–complete.

   (h) **T** The independent set problem is known to be $\mathcal{NP}$–complete.

   (i) **O** The factorization problem for binary numerals is $\mathcal{P}$–TIME.

   (j) **T** Every regulor language is $\mathcal{NC}$.

   (k) **T** Every context-free language is $\mathcal{NC}$.

   (l) **O** The Boolean circuit problem is $\mathcal{NC}$.

   (m) **T** The subset sum problem is known to be $\mathcal{NP}$–complete.

   (n) **T** The block sorting problem is known to be $\mathcal{NP}$–complete.

   (o) **T** If G_1 and G_2 are contex-free grammars that are not equivalent, there is a proof that they are not equivalent.

   (p) **F** Recall that a DPDA is a deterministic machine with finite memory plus a stack. A DPDA can emulate any deterministic machine.

   (q) **T** A 2-DPDA is a deterministic machine with finite memory plus two stacks. A 2-DPDA can emulate any deterministic machine.

2. State the pumping lemma for regular languages accurately. If you have all the right words, but the statement does not have the correct logical structure, you might get no credit.

For any regular language L
There exists an integer p such that
For any string $w \in L$ of length at least p
There exist strings x, y, z such that

1. $w = xyz$,
2. the length of xy is less than or equal to p,
3. y is not the empty string,
4. for any integer $i \geq 0$, $xy^iz \in L$.

3. (a) State the Church-Turing thesis.

Any machine is emulated by some Turing machine.

(b) Why is the Church-Turing thesis important?

If no Turing machine can do a given computational task, then no machine can. Turing machines are very simple, making a proof that a given computation is impossible easier.

4. Give a polynomial time reduction of 3-SAT to the Independent Set problem.

I will give a proof without a diagram, just to show you how to do it. Of course, I will accept a diagram based proof.

For each instance E of 3-SAT, we construct an instance of the independent set problem which has a solution if and only if E is satisfiable.

Let $E = C_1 * C_2 * C_3 * \cdots * C_k$ where each C_i is a clause of the form $(t_{i,1} + t_{i,2} + t_{i,3})$ where each term $t_{i,j}$ is either a Boolean variable or the negation of a Boolean variable.

Our instance of the independent set problem is $R(E) = (G, k)$ where G is a graph. Let the vertices of G be $V = \{v_{i,u} : 1 \leq i \leq k \text{ and } 1 \leq u \leq 3\}$. Let the edges of G be all pairs $\{v_{i,u}, v_{j,v}\}$ such that either $i = j$ or $t_{i,u} * t_{j,v}$ is a contradiction.

Note that G contains a 3-clique for each clause of E, namely $Q_i = \{v_{i,1}, v_{i,2}, v_{i,3}\}$ for each $1 \leq i \leq k$.

This is all you need to write to get credit on this problem, but I will now give the proof that R is a polynomial time reduction of 3-SAT to independent set.

Let n be the number of symbols it takes to write E. E has $3k$ terms, hence $n = \Theta(k)$. G has $3k$ vertices hence $O(n^2)$ edges. Thus $R(E) = (G, k)$ is computable in $O(n^2)$ time.

Suppose $I \subseteq V$ is an independent set of order k. Then I must contain exactly one vertex in each clique Q_i. Let $T = \{t_{i,u} : v_{i,u} \in I\}$. Choose an assignment of E such that each member of T is true. This is possible since no two members of T can contradict each other. Since T contains one term in each clause, E is satisfiable.

Conversely, suppose E is satisfiable. Pick a satisfying assignment of E. For each clause C_i, pick a term $t_{i,u[i]}$ which is true under that assignment, and let T be the set of those terms. Then $I = \{v_{i,u[i]}\}$ has order k and is independent since no two members of T can contradict each other.

5. Give a polynomial time reduction of the Subset Sum problem to Partition.

   If $X = x_1, x_2, \ldots x_n, K$ is an instance of the subset sum problem, we need to find an instance $Y = y_1, y_2, \ldots y_m$ of the partition problem such that X has a solution if and only if Y has a solution.

   Let $S = \sum_{i=1}^{n} x_i$. We know $S \geq K$. We let $m = n + 2$, $y_i = x_i$ for all $1 \leq i \leq n$, let $y_{n+1} = K + 1$, and $y_{n+2} = S - K + 1$. Note that $\sum Y = 2S + 2$.

   Suppose X has a solution $T \subseteq \{x_1, \ldots x_n\}$ such that $\sum T = K$. Then the sum of $T \cup \{y_{n+2}\}$ is $S + 1$, exactly half $\sum Y$, hence Y has a solution.

   Conversely, suppose $U \subseteq Y$ is a solution to Y, that is, $\sum U = S + 1$. y_{n+1} and y_{n+2} cannot both be members of U, since their sum is $S + 2$. Furthermore U must contain one of those terms, since the sum of the remaining terms is S. Without loss of generality, $\{y_{n+2}\} \in U$. Then $T = U - \{y_{n+2}\}$ has sum $S + 1 - (S - K + 2) = K$, and hence is a solution to X.