University of Nevada, Las Vegas Computer Science 456/656 Spring 2020

Answers to Assignment 5: Due Thursday April 9, 2020

Before working these problems, watch my videos. For some problems, you may need to search the internet..

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

   (a) **T** The regular language membership problem is known to be in Nick's class.

   (b) **T** The context-free language membership problem is known to be in Nick's class.

   (c) **T** The set of all recursively enumerable languages is countable.

   (d) **T** Every context-sensitive language is in $\mathcal{P}$-SPACE.

   (e) **F** $\mathcal{NC} = \mathcal{P}$-SPACE.

   (f) **O or F** If the integer factoring problem is in $\mathcal{P}$, there can be no secure one-way coding system. There are two ways to interpret this statement. If it is taken literally, the anser is **O**. If the sentence means, "If someone proves that the factoring problem is decidable, we will know that there is no secure one-way encryption system," the answer is **F**.

   (g) **F** The decidability problem is decidable.

   The decidability problem is: given an encoding of Turing machine, $\langle M \rangle$, is $L(M)$ decidable?

2. Give a polynomial time reduction of the subset sum problem to the partition problem.

   This problem is a replacement for Problem 3, which is much more difficult.

   The subset sum problem is to determine whether a given set of weighted items has a subset whose total weight is a given amount. For example, given a pocket of coins, can you select exactly one dollar's worth of those coins?

   The partition problem is a special case of the subset sum problem, where the desired sum is half the total of the items. For example, if you have a dollars worth of coins in your pocket, can you find a subset of those coins worth exactly fifty cents?

   We define a ruduction R from the subset sum problem to the partition problem. If $w = (K, x_1, x_2, \ldots x_n)$ is an instance of the subset sum problem, where all $x_i > 0$ then w is correct if and only if there is some subsequence of $x_1, x_2, \ldots x_n$ whose sum is K. Let $S = \sum_{i=1}^{n}$. Without loss of generality, $K \leq S$. Define $x_{n+1} = K + 1$ and $x_{n+2} = S - K + 1$. Let $R(w) = (x_1, x_2, \ldots x_n, x_{n+1}, x_{n+2})$ an instance of the partition problem. Note that $\sum_{i=1}^{n+2} = 2S + 2$. $R(w)$ is correct if and only if there is a subsequence of $R(w)$ whose sum is exactly half that, namely $S + 1$. Clearly, it takes polynomial time to compute $R(w)$ from w. We need to show that $R(w)$ is correct if and only if w is correct.

   Assume w is correct. Then there is a subsequence of $x_1, x_2, \ldots x_n$ whose sum is K. Take that subsequence and append x_{n+2}, and we have a solution to $R(w)$.

   Conversely, assume that $R(w)$ is correct. There are two disjoint subsequences of $R(w)$ each of which has total $S + 1$. Neither of those two subsequences can contain both x_{n+1} and x_{n+2}, since the sum of those two numers is $S + 2$, which is larger than the sum we need. Thus one of the subsequences contains x_{n+2} and does not contain x_{n+1}. The remaining members of that subsequence must have total K, hence W is correct.

3. Give a polynomial time reduction of the knapsack problem to the subset sum problem.

   The knapsack problem is to determine whether a given set of items, each of which has both a value and a size, has a subset whose total size is at most a given amount and whose total value is at least a given amount. For example, can a burglar find a set of items which fit into his knapsack whose total value is at least five hundred dollars?

   I actually assigned this problem by accident, but didn't want to remove it because someone might have started working on it. The problem is beyond the scope of what we have covered in class, and I will not attempt to give a solution.

4. Prove that the context-free grammar equivalence problem is in the class co-R.E. (Hint: CYK) The problem is, given encodings $\langle G_1 \rangle$ and $\langle G_2 \rangle$ of CF grammars, is $L(G_1) = L(G_2)$?

   For every string $w \in \Sigma^\star$, in topological order, use the CYK algorithm to determine whether $w \in L(G_1)$ and also whether $w \in L(G_2)$. If those two questions have different answers, we have determined that the grammars are not equivalent. If the grammars are not equivalent, we will eventually find a string which is in one language but not the other, and we answer, "No." If the answer is "Yes," we will test strings forever.

5. Classify each of these languages/problems, by giving one of these answers.
   1. Known to be $\mathcal{NC}$
   2. Known to be $\mathcal{P}$-TIME, but not known to be in $\mathcal{NC}$ and not known to be $\mathcal{NP}$-complete.
   3. $\mathcal{NP}$, but not known to be $\mathcal{P}$-TIME, and not known to be $\mathcal{NP}$-complete.
   4. Known to be $\mathcal{NP}$-complete.
   5. Known to be $\mathcal{P}$-SPACE, but not known to be $\mathcal{NP}$.
   6. Not known to be $\mathcal{P}$-SPACE

   (a) **2** The Boolean circuit problem.

   (b) **5** The puzzle Rush Hour described at `http://www.mathsonline.org/game/jam.html` The decision problem is, given a configuration, is there a way to win?

   (c) **3** The integer factoring problem, using binary notation.

   (d) **4** Boolean satisfiability.

   (e) **1** The context-free grammar membership problem.

   (f) **6** Generalized chess. The decision problem is, given a position, can White force a win?