CS 332
Spring 2018

CS 332 - Homework 4 - First Draft

Due: Thursday, March 29

Reading: Chapter 5.1 and 5.3 in the Sipser textbook, pages 213 - 216 and 234-238. So skip section 5.2 and also those results in 5.1 which refer back to Section 1 of the book. Chapter 7, pages 275 - 298.

PROBLEMS

Part 1: These 5 problems are to be turned in and each will be graded (10 points each).

1. Recall Dijkstra's algorithm. This algorithm takes as input an undirected, weighted graph G with N vertices and a specific vertex s of G. It outputs the length of the shortest path from s to any other vertex of G.

If you never learned this algorithm you should have. Look it up in your favorite algorithm book and see how it works. It’s pretty easy to understand, but go over it and make sure you see how it works.

i. What is the complexity (of big-O notation) of Dijkstra’s algorithm? You need not justify your answer but you should write where you found your version of the algorithm.

ii. Now consider the “all pairs shortest path” problem. This problem takes the same kind of input graph G as Dijkstra, but it tell you more. Namely, for each pair of vertices u, v from G it outputs the length of the shortest path from u to v.

Show how you can reduce the “all pairs shortest path” problem to Dijkstra’s algorithm.

That is, give an algorithm which solves “all pairs shortest path” and which uses Dijkstra’s algorithm as a procedure call (a subroutine).

What is the complexity of your algorithm?

iii. Give a lower bound for the “all pairs shortest path” problem

That is, find a function f(n) for which it would not be possible to solve the problem in less than O(f(n)). (I’m not looking for any difficult new algorithm here, just for a “trivial” f which any algorithm A which solves the problem cannot be smaller than. Your f should be strictly smaller than the complexity of the algorithm you found in part ii.

2. Say whether each of i. - iv. is True or False. In each case briefly justify your answer.

i. There are infinitely many undecidable problems.

ii. If L is decidable and J is undecidable than L ∪ J is undecidable.

iii. If L is decidable and J is undecidable than L ∩ J is undecidable.

iv. If J is reducible to K and K is recognizable then J is also recognizable.
3. Prove that the problem of deciding if a TM $M$ ever writes the symbol 1 on its tape is undecidable.

You should do these problems using reductions.

4. Prove that it is undecidable to determine if two TM’s $M$ and $L$ accept the same languages.

You should do these problems using reductions.

5. Assume $M$ is a TM which is restricted to always working on the non-blank part of its work tape (So the tape head on its tape never moves left off its tape or right past the last input symbol.)

Prove that the language of $M$ is decidable. In particular, give an algorithm which shows that for any input $w$ to $M$ we can decide if $M(w)$ loop or halts. Use this to give an algorithm which decides $L(M) = \text{the set of strings } w \text{ that } M \text{ accepts.}$

(Hint: To get started think about how many steps the TM can make while staying on the same tape square without repeating the same state and the same symbol on the tape square.)

Part 2: These problems are good practice and you should try them. They will not be graded.

1. Note: Some of the ideas for this problem were covered in section. You can see the exact definitions you need on page 271 of our textbook.

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

   b. Give an example of a Boolean formula $F$ which contains at least 3 different variables, and where $F$ is satisfiable and has exactly 4 satisfying truth assignments.

2. Page 239, problem 5.7

3. Page 239, problem 5.8