15-251 : Great Theoretical Ideas In Computer Science

Fall 2013

Assignment 7 (Chinese Food Homework)

Due: Thursday, Oct. 24, 2013 11:59 PM

Name: ______________________________

Andrew ID: ______________________________

Question:	1	2	3	4	5	Total
Points:	20	30	25	25	10	110
Score:						

A brief introduction to satisfiability (SAT)

A boolean formula is a statement consisting of "variables" and boolean operators on these variables (here, we'll only use AND ($\wedge$), OR ($\vee$), and NOT ($\neg$)). We assign TRUE or FALSE to each variable and evaluate the formula.

Satisfying Assignment: A satisfying assignment for a boolean formula is an assignment of truth values to the variables such that the formula evaluates to TRUE. For example, consider the formula $(\neg x_1 \vee \neg x_2) \wedge x_2$. Notice that the unique satisfying assignment for this formula is setting x_1 to FALSE and x_2 to TRUE. The formula $x \wedge \neg x$ does not have any satisfying assignment.

Literals: For each variable x, there are two "literals", x and $\neg x$.

CNF: We say that a boolean formula is in conjunctive normal form (CNF) if it is an AND of a finitely many "clauses", where each clause is an OR of finitely many literals. For example, $(x_1 \vee x_2) \wedge (\neg x_1) \wedge (x_2 \vee \neg x_3 \vee x_4 \vee \neg x_4)$ is a CNF formula, but $\neg(x_1 \vee x_2) \wedge (x_3 \vee x_4)$ is not.

***k*-CNF:** We say that a boolean formula is in k-CNF form if it is in CNF form where all the clauses use exactly k literals. For example, $(x_1 \vee x_2) \wedge (\neg x_2 \vee x_3)$ is in 2-CNF form, while $(x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_4 \vee x_4)$ is in 3-CNF form.

DNF: We say such a boolean formula is in disjunctive normal form (DNF) if it is an OR of a finitely many "clauses", where each clause is an AND of finitely many literals. For example, $(x_1 \wedge x_2) \vee (\neg x_1) \vee (x_2 \wedge \neg x_3 \wedge x_4 \wedge \neg x_4)$ is a DNF formula, but $(x_1 \wedge x_2) \vee (x_2 \wedge (x_3 \vee x_4))$ is not.

SAT: The satisfiability (SAT) problem is to check if a given a CNF formula has a satisfying assignment. SAT is known to be NP-complete (you can assume this in this problem).

***k*-SAT:** The problem of checking whether a given k-CNF formula has a satisfying assignment is known as k-SAT.

You cannot make any complexity assumptions about k-SAT or checking satisfiability of DNF formulae. Check out questions Q1(b) and Q1(c).

NP-completeness: Please review the relevant definitions carefully from the lecture slides before attempting this homework.

1. ## Pineapple Fried Rice

I just opened a Chinese food restaurant, General Tso's Kitchen, that serves delicious General Tso's chicken 24 hours a day, 365 days a year.

While I only have one item on the menu, I feel that it's all I need to satisfy my guests.

However, after being open for a few weeks, I've noticed something interesting. When two people come in, it's easy to keep them satisfied. However, it's quite hard to satisfy 3 people (although I can still verify that they're satisfied pretty easily!). Please help me understand this phenomenon.

(5) (a) For each of the following CNF formulas, give a satisfying assignment or explain why no such assignment exists.

i. $(x_1 \vee x_2) \wedge (x_3) \wedge (\neg x_1 \vee \neg x_3)$

ii. $(\neg x_2 \vee \neg x_3) \wedge (x_1 \vee x_2 \vee x_3 \vee \neg x_1) \wedge (x_3) \wedge (x_2 \vee \neg x_3)$

Solution:

(10) (b) Interestingly, 2-SAT is in P, but 3-SAT is NP-complete.

Prove that 3-SAT is NP-complete. Use a reduction from SAT.

Solution:

(5) (c) While CNFs and DNFs look duals to each other, and checking satisfiability of CNF formulae is NP-complete, show that DNF-SATISFIABILITY, the problem of checking satisfiability of a given DNF formula, is in P.

Solution:

2. Egg Rolls and Wonton Soup

(10) (a) General Tso's Kitchen is getting quite popular, so I've come up with a new idea to help expand: start delivering to customers!

There is quite a lot of competition out there, however, and I want to stand out. Because we only serve one item, I had the idea to promise delivery within one minute after the food is ordered.

To accomplish this, all I need are a bunch of delivery men waiting at various street intersections with buckets of delicious General Tso's chicken, ready to deliver. For each street segment, there should be a delivery man ready on at least one of the two end-points (street intersections) of the street segment. Formally, think of the intersections as vertices and the street segments as edges connecting the intersections.

Of course, I want to hire as few delivery men as possible to accomplish this. Define the problem TSO'S-DELIVERY as follows: Given a layout of intersections and street segments (as a graph) and an integer k, check if I can place at most k delivery men at the intersections to accomplish the delivery time I promised.

Give a polynomial-time reduction from TSO'S-DELIVERY to INDEPENDENT-SET. [Review lecture slides for the definition of INDEPENDENT-SET problem. Also, please carefully note the direction in which you have to do the reduction.]

Solution:

(20) (b) Now that my delivery men are doing all the work, I've been pretty bored during the day. Fortunately I have plenty of General Tso's chicken to eat and plenty of Minesweeper to play.

After mastering the expert level, I decided to come up with my own more general form of Minesweeper - one that can be played on any graph. This turned out to be quite hard!

Given a graph with integers on some of the vertices, MINESWEEPER determines if it is possible to put mines on a subset of the remaining vertices such that for each vertex with integer v_k, exactly v_k of its neighbors contain mines.

Prove that MINESWEEPER is NP-complete. Use 3-SAT when showing that it is NP-hard.

Solution:

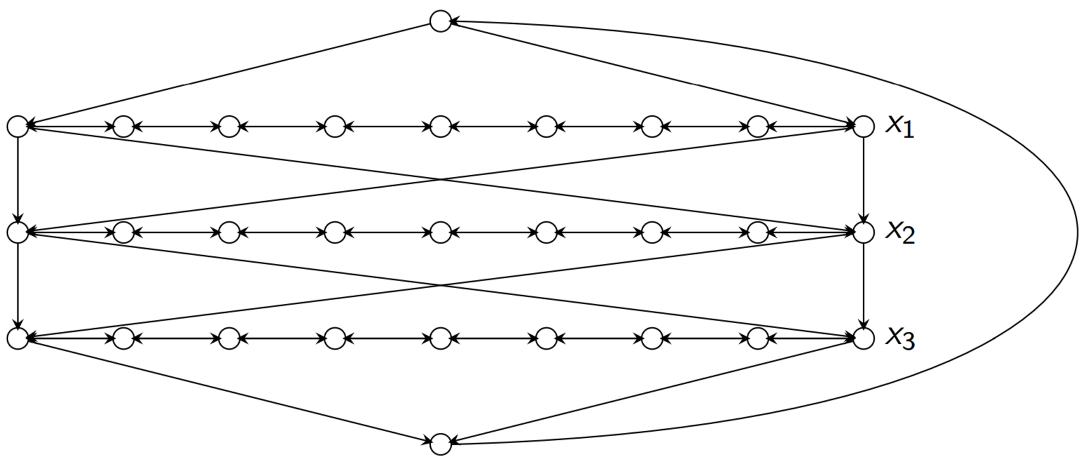
3. Soy Sauce

Wow, proving things about MINESWEEPER was even more fun than playing it! General Tso's Kitchen is doing quite well these days, but I want to come up with an even better recipe. Unfortunately, finding a better recipe is quite hard. To help me prepare to tackle this problem, I've been looking at a lot more hard problems (NP-hard, that is).

Recall 3-SAT from problem 1 and the HAMILTONIAN-CYCLE problem from lecture. Here, we'll consider a slight variant of the HAMILTONIAN-CYCLE problem, taking directed graphs as input rather than undirected graphs. I've been trying to come up with a polynomial-time reduction from 3-SAT to HAMILTONIAN-CYCLE, but need some help.

(5) (a) Given a 3-CNF formula, we'll construct a directed graph that has a Hamiltonian cycle if and only if the formula has a satisfying assignment.

First consider the following directed graph:



In general, for n variables and m clauses, imagine a similar figure with n layers, one per variable, and $3m$ vertices in each layer. The graph shown above is for 3 variables and 3 clauses. Hence, each of the 3 layers has 9 vertices.

How many Hamiltonian cycles does this graph (constructed for n variables and m clauses) have? In what natural way do these cycles correspond to assignments to variables in the corresponding 3-CNF formula?

Solution:

(20) (b) The reduction is not complete yet! (Convince yourself why...) Now, add a new vertex for each clause, and add appropriate edges to present vertices so that it restricts the Hamiltonian cycles such that a Hamiltonian cycle now exists if and only if a correspondence similar to what you gave in (a) actually leads to a satisfying assignment. Prove why this gives a polynomial-time reduction that we desired.

Solution:

4. Beef Chow Mein

As General Tso's Kitchen expands, I need to place more and more delivery men so that I can still guarantee fast delivery. However, now that the size of the input has increased, the algorithm I've created is taking way too long. I need to come up with a better way to do it!

Recall TSO'S-DELIVERY from problem 2a. Show that coming up with an efficient algorithm is extremely difficult (or likely impossible) by showing that TSO'S-DELIVERY is NP-complete.

(5) (a) First, show TSO'S-DELIVERY is in NP.

Solution:

(20) (b) Next, show that TSO'S-DELIVERY is NP-hard by giving a polynomial-time reduction from 3-SAT to TSO'S-DELIVERY.

Hint: Construct a graph with 2 vertices per variable and 3 vertices per clause, and add edges appropriately.

Solution:

5. Bonus Question

(10) (a) Now, define TSO'S-DELIVERY-EVEN as follows: Given a graph of intersections and street segments (as in problem 2a) *such that each intersection has an even number of street segments incident on it* and an integer k, is there a way to place at most k delivery men at intersections so that each street segment has a delivery man on at least one of its two ends?

Show that TSO'S-DELIVERY-EVEN is also NP-complete.

Solution: