CS3230: Design and Analysis of Algorithms (Fall 2014)

Tutorial Set #7

[For discussion during Week 9]

S-Problems are due (outside Prof. Leong's office): Friday, 10-Oct, before noon.

OUT: 29-Sep-2014 **Tutorials:** Tue & Wed, 14, 15 Oct 2014

IMPORTANT: Read "Remarks about Homework".

Submit solutions to S-Problem(s) by deadline given above.
Prepare your answers to all the D-Problems in every tutorial set.

When preparing to present your answers,

- Think of a CLEAR EXPLANATION
- Illustrate with a good worked example;
- Describe the main ideas,
- Can you sketch why the solution works;
- Give analysis of running time, if appropriate
- Can you think of other (perhaps simpler) solutions?

Helpful Hints Series: Think simple ☺

Please assume everywhere below that $\leq_P$ represents polynomial time Karp reduction (unless otherwise specified).
Please recall that decision problem are also known as languages.
Please recall that $\vee$ represent logical OR, $\wedge$ represents logical AND and $\neg$ represents logical NOT operation.

Routine Practice Problems *-- do not turn these in -- but make sure you know how to do them.*

R1. Can a decision problem A in P be NP-complete if P not equal to NP?

R2. NP stands for [No-problem/Non-polynomial/Non-deterministic-polynomial time]?

R3. Do we know this for sure: A polynomial time solution does not exist for CIRCUIT-SAT?

R4. Suppose A $\leq_P$ B, what are the implications?

1) If B is solvable in polynomial time, then A is solvable in polynomial time?
2) If A is not solvable in polynomial time, then B is not solvable in polynomial time?
3) If B is not solvable in polynomial time, then A is not solvable in polynomial time?
4) If A is solvable in polynomial time, then B is solvable in polynomial time?

S-Problems: (To do and submit by due date given in page 1)
Solve this S-problem(s) and submit for grading.

IMPORTANT: Write your NAME, Matric No, Tutorial Group in your Answer Sheet.

S1. [Understanding transformations]

An independent set in a graph G = (V,E) (V is the set of vertices and E is the set of edges of G) is a subset $I \subseteq V$ such that for all $u, v \in I:\ (u, v) \notin E$.

A clique in a graph G = (V,E) is a subset $J \subseteq V$ such that for all $u, v \in J:\ (u, v) \in E$.

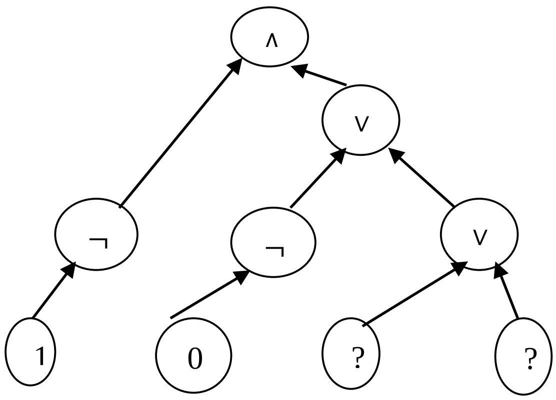
Given a graph G and a number k, transform it to a graph H such that G has an independent set of size of size at least k if and only if H has a clique of size at most k.

D-Problems: Solve these D-problems and prepare to discuss them in tutorial class. You may be called upon to present your solution *or your best attempt at a solution*. Your solution presentation does NOT need to be fully correct, given your best attempt. The TA will help clarify and correct any issues or errors.

D1. [Transformations]

a) Transform the CNF formula D= $(x1 \vee x2) \wedge (\overline{x3} \vee \overline{x2})$ to a circuit C with AND/OR/NOT gates such that C is satisfiable if and only if (iff) D is satisfiable.

b) Transform the following circuit C to a CNF formula D such that D is satisfiable iff C is satisfiable.



c) Transform a CNF formula C, in which each clause has at most 3 literals to a 3-CNF formula D in which each clause has exactly three literals, with the condition that D is satisfiable if and only if C is satisfiable.

D2. [Transitivity of reductions]

Show that if $X \leq_P Y$ and $Y \leq_P Z$ then $X \leq_P Z$.

D3. [Closure properties of P]

Let $A, B \in P$. Show that

1) $A \cup B \in P$,
2) $A \cap B \in P$,
3) $\overline{A} \in P$.

D4. [Cook v/s Karp reductions]

Show that every decision problem (language) L has a Cook reduction to $\overline{L}$ (the complement of L).
Can you show the same with Karp reduction (assume neither L nor $\overline{L}$ is empty)?

Advanced Problems – Try these for challenge and fun. There is no deadline for A-problems. *Turn in your attempts DIRECTLY to Prof. Leong. Do not combine it with your HW solutions.*)

A1. [A tricky transformation]

A 3-CNF formula is Not-All-Equal-Satisfiable if there a truth assignment to the variables so that each clause has at least one true literal and at least one false literal.

Given a 3-CNF formula C transform it to a 3-CNF formula D such that D is Not-All-Equal-Satisfiable iff C is satisfiable.