9/12/2013
E 9.1

Exercise 11.1.1 b)

Consider the problem FALSE-SAT:

Given a boolean expression E that is false when all its variables are made false, is there some other truth assignment that makes E false, besides all-false?

Decide whether the problem is in NP or coNP.

Describe its complement.

If the problem or its complement is NP-complete, prove it.

Proof:

The problem is NP-complete.

- In NP: given a boolean expression E, we need to check:
  1) that E is false when all variables are assigned false
  2) that there is some other truth-assignment making E false

  (1) can be done in poly-time by a DTM
  (2) can be done in poly-time by a NTM
  guess a truth-assignment T different from all false, and answer yes if under T, E evaluates to false

- NP-hard: by a reduction from SAT
  Let E be a boolean expression with variables $x_1, \dots, x_n$.
  We construct an expression E' s.t. $E \in SAT$ iff $E' \in$ FALSE-SAT
  1) Test if E is true when all variables are false (polynomial)
  If so, $E \in SAT$, and we convert it to a fixed expression that is in FALSE-SAT, e.g. $x \wedge y$.

2) Otherwise, let E' be $\neg E \wedge (x_1 \vee x_2 \vee \cdots \vee x_n)$. E 9.2

Clearly, the reduction is poly-time.

We have that E' is false when all of $x_1, \ldots, x_n$ are false.

Notice that in case (2), E is false when all variables are false.

Hence, if $E \in SAT$, then it is satisfied by a truth assignment T different from all-false.

Thus, $\neg E$ is made false by T, and $E' \in$ FALSE-SAT.

Conversely, if $E' \in$ FALSE-SAT, then since $x_1 \vee \cdots \vee x_n$ is false only for the all-false truth assignment, there must be some other truth-assignment T that makes $\neg E$ false. Then T makes E true, and $E \in SAT$.

Exercises on problems in P, NP, and NP-complete

Exercise 1:

Consider the following optimization version of SAT:

MAXSAT: Input: a propositional formula F in CNF, and
an integer k

Output: yes, if there is a truth assignment that satisfies
at least k clauses of F

no, otherwise

What is the complexity of MAXSAT?

a) MAXSAT $\in$ NP: immediate, by the following NP algorithm

1) guess a truth assignment α (non-deterministic polynomial)
2) count the # of clauses satisfied by α, and answer yes iff it is $\geq k$ (deterministic polynomial)

b) MAXSAT is NP-hard

This follows from the fact that CSAT is a special case of MAXSAT.

Formally, we can polynomially reduce CSAT to MAXSAT, i.e.

$$\text{SAT} <_{poly} \text{MAXSAT}$$

Given an instance F of CSAT, we construct an instance (F, k) of MAXSAT, where k is the # of clauses of F.

Obviously, k can be obtained in poly time from F, and

$$F \in \text{CSAT} \iff (F, k) \in \text{MAXSAT}$$

$\square$