KTH Computer Science
and Communication

Homework IV, Complexity Theory Fall 2011

Due on September 30 at 15.15, i.e. at the beginning of the lecture. The general rules on homework solutions available at the course home-page apply. In particular, discussions of ideas in groups of up to at most two people are allowed but solutions should be written down individually.

Some of the problems below are "classical" and hence their solutions is probably posted on the Internet. It is not allowed to use such solutions in any way. The order of the problems is "random" and hence do not expect that the lowest numbered problems are the easiest.

Any corrections or clarifications on this problem set will be posted under "homework" on the course home-page http://www.csc.kth.se/utbildning/kth/kurser/DD2446/kplx11/uppgifter.

1 (10p) This is information gathering problem and should be solved *individually*. 3-Sat is the problem of determining whether a formula in 3-CNF containing n variables is satisfiable. It is conjectured that the running time of the most efficient algorithm to solve this problem grows exponentially with n, i.e., it grows like 2^{cn} for a constant c. Find the best value of c obtained for any proposed algorithm. State the running time and give a reference to the result.

2 (10p) Normally we pose NP-problems as decision problems, i.e., given a formula φ we ask if it is satisfiable. Usually, if the formula is satisfiable we also want to find an assignment satisfying φ. This is called the "search problem". This is the problem of returning a satisfying assignment in the case when φ is satisfiable and the statement "not satisfiable" when φ is not satisfiable.

2a Prove (4p) that these two problems are equivalent in that if we can solve one in polynomial time then we can solve the other.

Is this a unique property for satisfiability or is the corresponding property true for any arbitrary NP-complete problem? In other words:

2b (6p) Is it true for any NP-complete language A that deciding $x \in A$ is polynomial time equivalent to finding a witness to this fact?

What is mean by a witness is usually intuitively clear but since we are talking about an arbitrary language A let us be specific. A is recognized by non-deterministic polynomial time Turing machine M. The witness that x belongs to A is a description of the non-deterministic choices that makes M output one on input x.

3 (15p) 2-Sat is the problem of given a formula φ in 2-CNF to decide whether it is satisfiable and an example is

$$\varphi = (x_1 \vee x_2) \wedge (\bar{x}_1 \vee x_3) \wedge (\bar{x}_2 \vee \bar{x}_3),$$

which is satisfied by making x_1 and x_3 true and x_2 false. Suppose it has m clauses and n variables where $n \leq m \leq n^2$.

3a (5p) Show that 2-Sat is in P.

Let us instead consider the maximization problem when you try to satisfy as many clauses of φ as possible. To make this a decision problem assume that you are also given a number k and the question is whether it is possible to satisfy at least k out of the m clauses. Call this problem OPT-2-Sat.

3b (10p) Prove that OPT-2-Sat is NP-complete.

In this second problem you are only allowed to assume that you know that problems discussed in class are NP-complete. These are 3-Sat, Clique, Vertex cover and Hamiltonian Circuit.

Please be formal about the part proving that OPT-2-Sat belongs to NP and in particular define a non-deterministic machine the recognizes this language.