Computational Complexity, II

1. For $A, B \subseteq \mathbf{N}$ let $A \oplus B$ denote $\{2x : x \in A\} \cup \{2x + 1 : x \in B\}$. Show: $A, B \leq_p A \oplus B$. Moreover, if $A, B \leq_p C$ then $A \oplus B \leq_p C$. (Compare with m-reducibility.)

2. Discuss nondeterministic Turing machines. Consider a restricted variant of nondeterminism, where a machine is allowed to choose between no more than two alternatives at each step. (Thus, a computation tree is binary.) Define the corresponding class of languages accepted in polynomial time. Is it the same as NP?

3. Let $L = \{\langle M, x, 1^t \rangle : M$ is a nondeterministic Turing machine that accepts x in $\leq t$ steps$\}$. Show that L is an NP-complete language.

4. (Huiswerk) Show that if $P = NP$ then a polynomial time algorithm exists that for a boolean formula finds a satisfying assignment, if it is satisfiable, and says 'NO' if it is not. (Thus, this algorithm delivers more information than just a decision algorithm for SAT.)

5. Show that the set of all boolean formulas that have at least two different satisfying assignments is NP-complete.

6. Consider the set of all graphs with a 4-clique. Is it in NP? Is it in P?