Logical Complexity
Final Test 21.12.2000

1. Give a complete specification of a Turing machine computing the function:

$$Mod_3 : \{0,1\}^* \to \{0,1,2\}$$

   where $Mod_3(x)$ is the number of symbols in the word x modulo three.

   What is the time and space complexity of your algorithm? (You are encouraged to use the big O-symbol.)

2. Is $\overline{\text{HALT}} \leq_m \text{HALT}$?

3. Consider the set of codes of all Turing machines which accept a finite language. Is it decidable? (*Hint: use the Chinese equivalent of the potato.*)

4. Is there a polynomial time Turing machine computing the function 2^x, where both input and output are given in binary representation?

5. A graph is called 3-colorable if you can paint its vertices using three colors such that no edge connects two vertices of the same color. (So, every vertex has exactly one color.) Show that the problem of testing whether a graph is 3-colorable is in NP.

6. For a set $A \subset \mathbf{N}$, let $A' = \{7x \mid x \in A\}$. Show that $A \leq_p A'$ and that $A' \leq_p A$.

7. A quantified boolean formula is called existential if it is of the form

$$\exists p_1 \ldots \exists p_n \; \varphi(p_1, \ldots, p_n)$$

   where φ is quantifier free. Show that testing the truth of an existential quantfied boolean formula is an NP-complete problem.

8. Show that if P = NP then co-NP = NP.

9. Show that for any sets X and Y, one has
   $X \leq_p Y \Leftrightarrow \overline{X} \leq_p \overline{Y}$.

10. Use the above to prove:
    co-NP = NP $\Leftrightarrow$ SAT $\in$ co-NP.