Computability and Computational Complexity, A.Y. 2024–2025

Written test

Wednesday, June 11, 2025

Exercise 1

1.1) Prove that the two following problems belong to **NP**:

P_1: Given a finite list L of unordered pairs of persons, where $\{a, b\} \in L$ means "a and b know each other", and a positive integer k, is there an individual who knows at least k other people?

P_2: Given a finite list L of unordered pairs of persons, where $\{a, b\} \in L$ means "a and b know each other", and a positive integer k, is there a group of k people who all know each other?

1.2) Prove the two following statements:

$$\text{If } P_2 \in \mathbf{P} \text{ then } \mathbf{P} = \mathbf{NP}.$$

$$\text{If } P_1 \text{ is } \mathbf{NP}\text{-complete then } P_2 \in \mathbf{P}.$$

Hint — *Here is a list of languages that you can assume to be **NP**-complete without having to prove it: SATISFIABILITY, 3-SATISFIABILITY, CLIQUE, INDEPENDENT SET, INTEGER LINEAR PROGRAMMING, VERTEX COVER, 3-VERTEX COLORING, SUBSET SUM, KNAPSACK, HAMILTONIAN PATH, DIRECTED HAMILTONIAN CYCLE, HAMILTONIAN CYCLE, TRAVELING SALESMAN PROBLEM.*

Exercise 2

For each of the following properties of Turing machines $\mathcal{M}$, prove whether it is recursive or not. Whenever possible, use Rice's theorem.

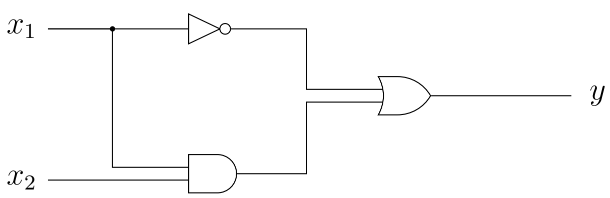
2.1) $\mathcal{M}$ either performs less than 100 steps or runs forever when executed on an empty tape;

2.2) $\mathcal{M}$ never visits any state more than ten times when executed on an empty tape;

2.3) $\mathcal{M}$ recognizes Turing machines with more states than alphabet symbols.

Exercise 3

Consider the following Boolean circuit representing a Boolean function $y = f(x_1, x_2)$:



3.1) Write the function f in terms of the Boolean operators $\wedge$ (and), $\vee$ (or) and $\neg$ (not) on the two variables x_1 and x_2.

3.2) Write a 3CNF formula on the three variables x_1, x_2 and y (and, if needed, other auxiliary variables for gate outputs) that is satisfiable if and only if $y = f(x_1, x_2)$ (i.e., if x_1, x_2 and y have values that are compatible with the given Boolean circuit).

Hint — *Point 3.2 can be solved in two ways: by directly writing the dependency as $y \Leftrightarrow f(x_1, x_2)$ and applying Boolean algebraic rules to work out a 3CNF formula, or by writing down a 3CNF formula for each gate and requiring them all to be true. The second way is the one discussed in the course.*