Homework 13

2019.05.30

Note: When the exercise asks you to "design an algorithm for…," it always means that "designs an EFFICIENT algorithm for … and ANALYZES your algorithm". You should keep this in mind when writing solutions.

1. Exercises 34.1-5

Show that if an algorithm makes at most a constant number of calls to polynomial-time subroutines and performs an additional amount of work that also takes polynomial time, then it runs in polynomial time. Also show that a polynomial number of calls to polynomial-time subroutines may result in an exponential-time algorithm.

2. Exercises 34.1-6

Show that the class **P**, viewed as a set of languages, is closed under union, intersection, concatenation, complement, and Kleene star. That is, if L_1 , $L_2 \in$ **P**, then $L_1 \bigcup L_2 \in$ **P**, $L_1 \bigcap L_2 \in$ **P**, $L_1 L_2 \in$ **P**, $(L_1)^c \in$ **P**, and $L_1{}^* \in$ **P**.

3. Exercises 34.2-3

Show that if HAM-CYCLE $\in$ **P**, then the problem of listing the vertices of a Hamiltonian cycle, in order, is polynomial-time solvable.

***Note* 1**: HAM-CYCLE is defined as " Does a graph G have a Hamiltonian cycle? "

***Note* 2**: " HAM-CYCLE $\in$ **P** " means that HAM-CYCLE is polynomial-time solvable.

4. Exercises 34.2-7

Show that the Hamiltonian-path problem can be solved in polynomial time on directed acyclic graphs. Give an efficient algorithm for the problem.

5. Exercises 34.2-9

Prove that **P** $\subseteq$ **NP** $\cap$ co-**NP**.

6. Exercises 34.2-10

Prove that if **NP** $\neq$ co-**NP**, then **P** $\neq$ **NP**.

7. Exercises 34.3-6

A language L is **complete** for a language class ***C*** with respect to polynomial-time reductions if $L \in \boldsymbol{C}$ and $L' \leq_P L$ for all $L' \in \boldsymbol{C}$. Show that $\varnothing$ and $\{0, 1\}^*$ are the only languages in **P** that are not complete for **P** with respect to polynomial-time reductions.