Computability and Computational Complexity, A.Y. 2025–2026

Written test

Friday February 20, 2026

Exercise 1

1.1) Show that the set R of recursive languages is closed with respect to:

- language union (i.e., if $L_1, L_2 \in R$ then $L_1 \cup L_2 \in R$),
- language intersection (if $L_1, L_2 \in R$ then $L_1 \cap L_2 \in R$) and
- language complementation (if $L \in R$ then $\bar{L} = \Sigma^* \setminus L \in R$).

1.2) What can be said about the set RE of recursively enumerable languages?

Exercise 2

The FACTORING language is the decision version of the prime factoring problem. It contains all integer pairs (N, k) such that N has a proper divisor not greater than k:

$$\text{FACTORING} = \{(N, k) \in \mathbb{N}^2 \mid \exists m \,:\, 1 < m \leq k \wedge N \bmod m = 0\}.$$

In other words, N is divisible by some number greater than 1 but not larger than k. For instance:

- $(36, 3) \in$ FACTORING, because $N = 36$ is divisible by $m = 2$ (which is smaller than $k = 3$);
- $(35, 4) \notin$ FACTORING, because the smallest proper divisor of $N = 35$ is 5, which is larger than $k = 4$ (i.e, no number m between 2 and $k = 4$ divides $N = 35$).

2.1) Write an algorithm (in any language or pseudocode you like) that decides FACTORING. Assuming that integers are encoded with a positional system (e.g. binary), show that your algorithm runs in exponential time with respect to the input size.

2.2) Prove that FACTORING $\in$ **NP**.

2.3) Prove that FACTORING $\in$ **coNP**.

Hint — *For point 2.1, just iterate over all values of $m = 2, \dots, k$; when you discuss the algorithm's complexity, remember that N and k are the input values, but the input size is much less.*

If, against all odds, you find a polynomial-time algorithm, you should publish your answer on a major Math or CS journal.

Exercise 3

Let $G = (V, E)$ be a directed graph (meaning that its edges are *ordered* pairs: $E \subseteq V \times V$). A *cycle* in G is a sequence of $k \geq 2$ vertices $v_1, \ldots, v_k \in V$ such that consecutive vertices are connected by an edge in the correct direction:

$$\forall i = 1, \ldots, k-1 \quad (v_i, v_{i+1}) \in E,$$

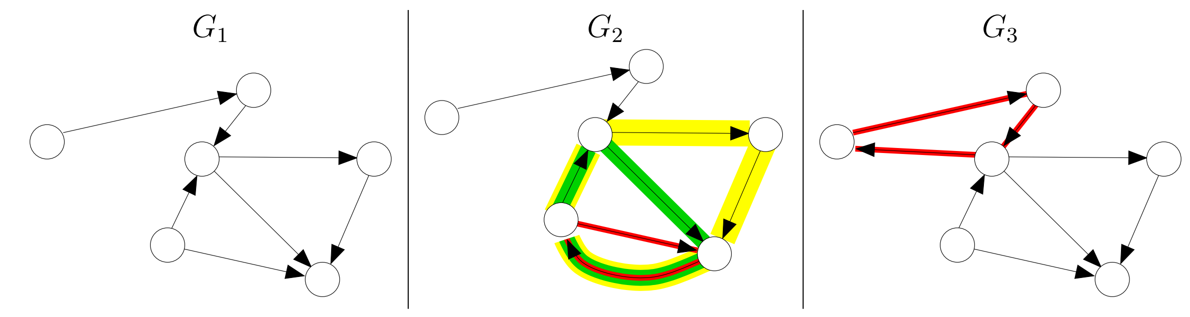
and the last vertex is connected to the first:

$$(v_k, v_1) \in E.$$

We define the language

$$\text{HAS\_CYCLES} = \{G | G \text{ is a directed graph and contains at least one cycle}\}.$$

For example, consider the three directed graphs:



Then:

- $G_1 \notin$ HAS_CYCLES (it doesn't contain cycles);
- $G_2, G_3 \in$ HAS_CYCLES because both contain at least a cycle (highlighted).

3.1) Prove that HAS_CYCLES $\in$ **P**.

3.2) Prove that HAS_CYCLES $\in$ **NL**.

Hint — *While some (pseudo-) code would be ideal, a word-only high-level description of the decision algorithms is fine.*