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

Written test

Tuesday, June 23, 2026

Exercise 1

Consider the alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ and the language

$$L = \{\mathtt{a}^n\mathtt{bba}^n | n \in \mathbb{N}\} \subseteq \Sigma^*$$

i.e., a string belongs to L if it contains zero or more a's, followed by two b's and then the same number of a's, for example:

$$\mathtt{bb} \in L, \quad \mathtt{aabbaa} \in L, \quad \mathtt{aaabbaa} \notin L, \quad \mathtt{aaabaaa} \notin L.$$

1.1) Specify a one-tape, three-symbol deterministic Turing machine (on alphabet $\Sigma \cup \{\sqcup\}$) that decides L.
What is the time complexity of your solution with respect to the input size (logarithmic, linear, quadratic, cubic, exponential...), and why?

1.2) Show that $L \in$ **L**.

1.3) Show that a two-tape DTM could decide L in linear time.

Hint — *1.1: A DTM can be specified as a state-transition diagram or as a transition table indexed by symbols and states. If you cannot do it (or you have no time), then a verbal description is fine as long as it describes elementary actions of a TM (e.g., "count the number of a's" isn't acceptable unless you specify how to do it). Remember that a one-tape DTM is allowed to modify its input (e.g., delete a's as soon as they are taken into consideration).*
1.2 and 1.3: here a high-level description is fine, as long as it allows us to agree on the conclusion.

Exercise 2

Let DOUBLE-SAT be the language of CNF Boolean formulas with *at least two* satisfying assignments.

2.1) Show that DOUBLE-SAT $\in$ **NP**.

2.2) Show that DOUBLE-SAT is **NP**-complete.

Hint — *For 2.2, try to reduce SAT to DOUBLE-SAT. Given CNF formula $f(x_1, \dots, x_n)$, add a dummy variable y to it obtaining $f'(x_1, \dots, x_n, y)$ such that f' is satisfiable if and only if f is too, but y can be either true or false, so that f' has twice the number of solutions as f.*

Exercise 3

Consider a universal Turing machine $\mathcal{U}$ able to simulate 1-tape TMs on alphabet $\{0, 1, \sqcup\}$.
Given a string $s \in \{0,1\}^*$, remember that its *Kolmogorov complexity* $K_{\mathcal{U}}(s)$ is the size of the smallest Turing machine (in $\mathcal{U}$'s notation) that, when started on an empty tape, outputs s and halts.
Show that the set

$$\left\{s \in \{0,1\}^* | K_{\mathcal{U}}(s) \leq 10^6\right\}$$

(binary strings having Kolmogorov complexity not larger than 10^6) is recursively enumerable.