Computability and Computational Complexity, A.Y. 2023–2024

Written test

Thursday, August 29, 2024

Exercise 1

1.1) Define the space complexity class **L**.

1.2) Consider the following language of strings in $\{\mathtt{a},\mathtt{b}\}^*$:

$$L = \{\mathtt{a}^{n_1}\mathtt{b}\mathtt{a}^{n_2}\mathtt{b}\mathtt{a}^{n_3}\mathtt{b}\cdots\mathtt{b}\mathtt{a}^{n_k} \,:\, k \in \mathbb{N} \wedge k \geq 1 \wedge n_1 > n_2 > n_3 > \cdots > n_k > 0\}.$$

Namely, a string is in L iff it is composed of non-empty sequences of a's of decreasing length separated by single b's, e.g.:

aaaaabaaaba $\in L$, aaba $\in L$,
aaabaaa $\notin L$ (two subsequences with the same number of a's: $n_1 = n_2 = 3$),
aaabaaba $\in L$, aaaaaaaaaaaaaaba $\in L$,
aaabbaa $\notin L$ (two consecutive b's),
aabaaa $\notin L$ (the sequences of b's are not of decreasing length).

Prove that $L \in$ **L**.

Hint — *For 1.2: start by sketching down a (pseudocode) program that decides L.*

Exercise 2

A teacher wants to divide a large class of N students into a small number $k \leq N$ of groups, each to be assigned a different project. She sets a constraint on how groups are formed:

No-past-collaboration constraint*: Every group must be composed of students who never collaborated before — i.e., if two students already collaborated in a previous group project, then they must be placed in different groups.*

The teacher wants to know if she has prepared enough projects to be able to satisfy the constraint. Luckily for her, the College's Statistical Service maintains a comprehensive list of student groups formed in the past. Thus, she can formally define the following decision problem.

Given (1) the number N of students, (2) the list of past student collaborations (e.g., in the form of a list pairs of students) and (3) the number k of available projects, is it possible to split the N students into k groups so that the ***no-past-collaboration*** *constraint is satisfied?*

2.1) What complexity class does the above defined decision problem belong to, and why?

2.2) In particular, for what values of k does an efficient decision procedure exist? Among them, for what values of k is the decision trivial?
Note that group sizes need not be balanced, nor are we required to actually create the groups: we only need to decide if the number k of projects is enough, or if the teacher needs to devise more of them.

Exercise 3

3.1) State Rice's Theorem about the (un-) decidability of Turing Machine properties as precisely as possible.

3.2) Outline a proof of the Theorem.