Johannes Blömer
Jan Bobolz, Gennadij Liske

02. June 2016
submission due: 13. June 2016, 15:30

Complexity Theory

SS 2016

Homework 8

Exercise 1 (6 points)**:**
Let $f : \Sigma^* \to \Sigma^*$ be a log space computable function. Show that f can be computed in polynomial time. Infer that any log space computable function has at most polynomial length output.

Exercise 2 (8 points)**:**
Consider the language

$$\mathit{SUMPAL} := \{(a, b) \in \mathbb{N}^2 \mid a + b \text{ is a palindrome}\}.$$

($a + b$ is considered the unique binary representation of $a + b$ without leading zeros.)
Show that $\mathit{SUMPAL} \in \mathbf{L}$.

Exercise 3 (10 points)**:**
In the lecture, we use the notion of log space computability from [1] (over 3-tape Turing machines) in order to define log space reductions. In [2] they use an alternative approach: They define log space reductions over *implicitly log space computable functions*.
Namely, a function $f : \{0,1\}^* \to \{0,1\}^*$ is *implicitly log space computable* if it fulfills the following requirements:

1. f is polynomially bounded (i.e. there is a polynomial p such that $|f(x)| \leq p(|x|)$ for all x).
2. $L_f = \{\langle x, i\rangle \mid f(x)_i = 1\} \in \mathbf{L}$.
3. $L'_f = \{\langle x, i\rangle \mid i \leq |f(x)|\} \in \mathbf{L}$.

Here, $f(x)_i$ denotes the i'th bit of $f(x)$.

a) Show that the two definitions are equivalent, i.e. a function f is log space computable if and only if it is *implicitly* log space computable.

b) Show that requirement 1 is necessary for the equivalence. For this, give an example of a function f that is not log space computable but that fulfills requirements 2 and 3, i.e. $L_f, L'_f \in \mathbf{L}$.

Literatur

[1] Michael Sipser, *Introduction to the Theory of Computation*, 2nd edition, 2006

[2] Sanjeev Arora and Boaz Barak, *Computational Complexity – A Modern Approach*, 2009, Draft available online: `http://theory.cs.princeton.edu/complexity/`