Complexity Theory – Homework 11

Discussed on 21.07.2010.

Definition 1. A language L is in $\mathbf{P}_{/\mathbf{poly}}$ if there exist a family $\{C_n\}$ of Boolean circuits of size polynomial in n such that for all $x \in \{0,1\}^n$

$$x \in L \text{ iff } C_n(x) = 1.$$

A family of Boolean circuits $\{C_n \mid n \in \mathbb{N}\}$ is *logspace uniform* if there is a deterministic Turing machine M running in logarithmic space which on input 1^n outputs a description of C_n. Similarly for *polytime unifrom* we require M run in polynomial time.

(Note that the definition of **NC** requires the logspace uniformity together with polynomial size and polylog depth.)

Exercise 11.1

Show that $\mathbf{BPP} \subseteq \mathbf{P}_{/\mathbf{poly}}$.

Remark: Use one of the results on **BPP** which have already been shown in the lecture.

Exercise 11.2

(a) Show that for every polynomial p the following language is in **coNP**:

$L_p := \{\langle C_1, C_2, \ldots, C_n\rangle \mid C_i$ is a circuit of size at most $p(i)$ which decides SAT for every formula of length exactly $i\}$.

Remark: Assume w.l.o.g. that every formula has length at least one with 0 (false) and 1 (true) the two formulae of length 1. Now, use the circuits $C_1, \ldots, C_i$ ($i \geq 0$) to check the correctness of circuit C_{i+1}. (Recall the so-called self-reducibility of SAT.)

(b) Show that **PH** collapses to the second level if $\mathbf{NP} \subseteq \mathbf{P}_{/\mathbf{poly}}$, i.e. if there is a sequnce of polynomial sized circuits for SAT.

Remark: It suffices to show that $\Pi_2\text{SAT} \in \mathbf{\Sigma}_2^p$.

(c) What happens if there is a sequnce of polynomial sized circuits for SAT that is moreover logspace uniform? What if it is polytime uniform?

Exercise 11.3

Prove that for $n \geq 100$, most of the boolean functions on n variables require circuits of size at least $2^n/n$.

Exercise 11.4

(a) Design a circuit family for the parity problem and describe it formally. Prove that there is a logspace uniform one.

(b) Let $A[0..n]$ be an array of integers. Design a PRAM for summing numbers in an array, i.e. compute $\sum_{i=0}^{n} A[i]$. Can you compute the array-suffix-sum, i.e. $\sum_{i=j}^{n} A[i]$ for all $0 \leq j \leq n$, with the same complexity?