5. Assignment on

Computational Complexity Theory

SoSe 2016

Wolfgang Mulzer, Claudia Dieckmann

Due on 31. May 2016 in the tutorial session

Problem 1 NL *10 points*

(a) Analyze the running time of Savitch's algorithm for the *st*-path problem.

(b) Show the following properties of LOGSPACE-reductions. Let $L_1, L_2, L_3 \subseteq \{0,1\}^*$.

  (i) If $L_1 \leq_L L_2$ and $L_2 \leq_L L_3$, then $L_1 \leq_L L_3$.

  (ii) If $L_2 \in$ LOGSPACE and $L_1 \leq_L L_2$, then $L_1 \in$ LOGSPACE

Problem 2 2SAT *10 points*

In 2SAT, we are given a Boolean formula in CNF such that each clause contains at most 2 literals. Let Ψ be a 2SAT-formula, and let $(l_1 \vee l_2)$ be a clause in Ψ, where l_1 and l_2 are (not necessarily distinct!) literals. The clause $(l_1 \vee l_2)$ can also be interpreted as the *implications* $\neg l_1 \to l_2$ resp. $\neg l_2 \to l_1$.

This observation inspires the following definition. The *implication graph* G_Ψ for Ψ is a directed graph. For each variable x in Ψ, there are two vertices, namely x and $\neg x$. For each clause $(l_1 \vee l_2)$ in Ψ there are two edges, one from $\neg l_1$ to l_2 and one from $\neg l_2$ to l_1 (where we interpret $\neg\neg x$ as x).

(a) Show that Ψ is unsatisfiable if and only if there exists a variable x such that G_Ψ has a directed cycle that contains both x and $\neg x$.

(b) Show that 2SAT is NL-complete.

Problem 3 Sublogarithmic Space *10 points*

Let $S : \mathbb{N} \to \mathbb{N}$ with $S(n) = o(\log \log n)$. Show that DSPACE(S) = DSPACE(1).

Hint: Suppose that $L \in$ DSPACE(S), and let M be a TM that decides L with $\text{space}_M(n) \leq S(n)$.

Recall the definition of a *crossing sequence* from Problem 3.3. Give an appropriate definition for a crossing sequence of the ith cell *on the input tape* that also takes the state of the work-tapes into account. Show that the number of distinct crossing sequences is at most $(2^{O(S(n))})^{2^{O(S(n))}} = o(n)$.

Now let n_0 be such that for $n \geq n_0$, we have $(2^{O(S(n))})^{2^{O(S(n))}} < n/3$, and for $n \leq n_0$, we have $S(n) < n_0$. Why does n_0 always exist?

If $\mathrm{space}_M(n) < n_0$ for all $n \in \mathbb{N}$, we are done. Otherwise, let w be a shortest word for which $\mathrm{space}_M(w) \geq n_0$. Note that $|w| \geq n_0$. Conclude that during the computation on w, there must be a crossing sequence that occurs at least three times. Derive from this a contradiction to the choice of w as a shortest word with $\mathrm{space}_M(w) \geq n_0$.