Schedule S1, (Computer Science, CS and Philosophy, Maths and CS)
Schedule B, MSc in Computer Science

Hilary Term 2019

Computational Complexity

Exercise class 5: log space

1. The following definitions (hopefully mostly familiar to you) will be used in this question.

   Definition. A finite automaton is a tuple $(Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of states, Σ is a finite set of symbols called the alphabet, δ is the transition function, which maps pairs in $Q \times \Sigma$ to a set of states in $\mathcal{P}(Q)$ (with $\mathcal{P}$ denoting powerset), $q_0 \in Q$ is the initial state and $F \subseteq Q$ is the set of accepting states. We say that a finite automaton is deterministic if the transition function is of the form $Q \times \Sigma \mapsto Q$; that is, if pairs of state and symbol are mapped to single states (rather than to sets of states).

   Definition. Let A be a finite automaton. A run r of A over a word $w = a_0 \ldots a_{n-1} \in \Sigma^n$ is a sequence of states $s_0 \ldots s_n \in Q^{n+1}$ such that $s_0 = q_0$ and $s_{i+1} \in \delta(s_i, a_i)$ for $0 \leq i < n$. A run r is accepting if $s_n \in F$. A word w is accepted by A if there exists an accepting run of A over w. The language of A, denoted $L(A)$, is the set of words accepted by A.

   Definition. We define the following languages for finite automata:

$$
\begin{aligned}
\textsc{Acceptance} &= \{\langle A, w\rangle \mid A \text{ a finite automaton, } w \text{ a word, and } w \in L(A)\} \\
\text{D-}\textsc{Acceptance} &= \{\langle A, w\rangle \mid A \text{ a deterministic finite automaton, } w \text{ a word, and } w \in L(A)\} \\
\textsc{Emptyness} &= \{A \mid A \text{ a finite automaton and } L(A) = \emptyset\} \\
\text{D-}\textsc{Emptyness} &= \{A \mid A \text{ a deterministic finite automaton and } L(A) = \emptyset\} \\
\textsc{Equivalence} &= \{\langle A_1, A_2\rangle \mid \text{ finite automata over the same alphabet, } L(A_1) = L(A_2)\} \\
\text{D-}\textsc{Equivalence} &= \{\langle A_1, A_2\rangle \mid \text{ deterministic finite automata over same alphabet, } L(A_1) = L(A_2)\}
\end{aligned}
$$

   (a) Show that D-$\textsc{Acceptance}$ is in $\textsc{LogSpace}$.

   (b) Show that $\textsc{Acceptance}$ is in $\text{N}\textsc{LogSpace}$.

   (c) Show that $\overline{\textsc{Emptyness}}$ is in $\text{N}\textsc{LogSpace}$.

   (d) Show that $\overline{\text{D-}\textsc{Emptyness}}$ is $\text{N}\textsc{LogSpace}$-hard.

   (e) Given the aforementioned results, establish tight complexity bounds for the following problems (that is, argue for which class each of these problems is complete):

   - $\overline{\textsc{Emptyness}}$
   - $\overline{\text{D-}\textsc{Emptyness}}$
   - $\textsc{Emptyness}$
   - D-$\textsc{Emptyness}$

   (f) Show that D-$\textsc{Equivalence}$ is in $\text{N}\textsc{LogSpace}$. *Hint: Note that A_1 and A_2 are non-equivalent if and only if either the intersection of A_1 and the complement of A_2 is non-empty, or the intersection of A_2 and the complement of A_1 is non-empty.*

(g) Show that EQUIVALENCE is in PSPACE. *Hint: Note that A_1 and A_2 are non-equivalent if and only if there exists a word that is accepted by one automaton but not the other. Furthermore, if such a word exists, there exists one of length at most $2^{|Q_1|+|Q_2|}$ with Q_1 and Q_2 the states of A_1 and A_2 respectively.*

2. (a) Let $f : \Sigma^* \to \Sigma^*$ be a function computed by a logarithmic space bounded deterministic Turing machine. Show that there is a constant d so that for all $w \in \Sigma^*$, $|w| > 1$, $|f(w)| \leq |w|^d$.

   (b) Show that LOGSPACE-reductions are transitive, i.e. show that if f and g are LOGSPACE-computable functions then so is $f \circ g$, the composition of f and g. $((f \circ g)(w) = f(g(w)))$

3. Let $\Sigma := \{0, 1\}$ and let PAL $\subseteq \Sigma^*$ be the language of palindromes over the alphabet Σ.

   (a) Show that PAL cannot be decided in logarithmic space by a Turing-machine that can only move its head on the input tape to the right. (That is, at every time step, the machine moves its input head to the right, similar to the operation of a finite deterministic automaton. It can operate arbitrarily on its work tape, though, but can only use $d \log n$ tape cells, for some constant d, where n is the input size.)

   (b) Show that the language $\mathcal{L} := \{w \in \Sigma^* : w \text{ is not a palindrome}\}$ can be decided in non-deterministic logarithmic space by a Turing-machine as above, that can only move its read head to the right.

4. A directed graph G is *strongly connected* if every two nodes in G are connected by a directed path in each direction. The language SC is the language of strongly connected graphs, i.e.:

$$SC = \{\langle G \rangle \mid G \text{ is a strongly connected graph}\}.$$

   Show that SC is NL-complete.