Schedule A2, (Computer Science, CS and Philosophy, Maths and CS)
Hilary Term 2023

Computational Complexity

Exercise class 4: oracle TMs, space-bounded computation

1. The following definitions will be used in this question:

   Definition. An *oracle Turing Machine* (OTM) is a Turing Machine M that has a special read-write tape (the machine's oracle tape) and three special states: q_{query}, q_{yes}, q_{no}. To execute M, we specify in addition a language $\mathcal{O} \subseteq \{0,1\}^*$ that is used as the oracle for M. Whenever during the execution M enters the state q_{query}, the machine moves to the state q_{yes} if $w \in \mathcal{O}$ and q_{no} if $w \notin \mathcal{O}$, where w denotes the contents of the special oracle tape. Regardless of the choice of $\mathcal{O}$, a membership query to $\mathcal{O}$ counts only as a *single computation step*. For every $\mathcal{O} \in \{0,1\}^*$, $\mathrm{P}^{\mathcal{O}}$ is the class of problems that can be decided by a polynomial-time deterministic TM with oracle access to $\mathcal{O}$. The corresponding class of problems based on non-deterministic Turing machines is denoted as $\mathrm{NP}^{\mathcal{O}}$.

   Definition. For any 3CNF formula ϕ, let $MAX_{true}(\phi)$ be the maximum number of variables set to true in a satisfying assignment for ϕ. If ϕ is not satisfiable, $MAX_{true}(\phi) = 0$. We now define the following languages:

   $$\textsc{MaxTrue3Sat} = \{\langle \phi, n\rangle \ : \ MAX_{true}(\phi) \text{ is at least } n\}$$
   $$\textsc{OddMaxTrue3Sat} = \{\phi \ : \ MAX_{true}(\phi) \text{ is odd}\}$$

   (a) Show that if $\mathcal{O} \in \mathrm{P}$, then $\mathrm{P}^{\mathcal{O}} = \mathrm{P}$.

   (b) Show that $\mathrm{NP} \cup \text{co-NP} \subseteq \mathrm{P}^{\mathrm{SAT}}$.

   (c) Let us define $\mathrm{P}^{\mathrm{NP}} = \bigcup_{\mathcal{O} \in \mathrm{NP}} P^{\mathcal{O}}$. Show that $\mathrm{P}^{\mathrm{NP}} = \mathrm{P}^{\mathrm{SAT}}$.

   (d) Show that $\textsc{MaxTrue3Sat}$ is in NP.

   (e) Show that $\textsc{OddMaxTrue3Sat}$ is contained in $\mathrm{P}^{\mathrm{SAT}}$.

2. The following definitions will be used in this question:

   Definition. A propositional formula is minimal if there is no smaller formula equivalent to it (a formula ϕ is smaller than formula ψ if the binary representation of ϕ is smaller than the binary representation of ψ). Then, we define the following language:

   $$\textsc{NotMinFormula} = \{\phi \ : \ \phi \text{ is not minimal}\}$$

   (a) Show that $\textsc{NotMinFormula}$ is in $\mathrm{NP}^{\mathrm{SAT}}$.

   (b) Show that $\textsc{NotMinFormula}$ is in APTIME by designing an alternating algorithm that solves the problem.

3. Given a non-deterministic polynomially-time bounded Turing machine M and a word w, denote by $Acc(M,w)$ all the accepting computations of M on w and by $Rej(M,w)$ all the rejecting computations.

   Let PP be the complexity class defined as follows: a language L is in PP if there exists a non-deterministic polynomially-time bounded Turing machine M such that $w \in L$ if and only if
   $$|Acc(M,w)| \geq |Rej(M,w)|.$$
   Show that PP $\subseteq$ PSPACE and NP $\subseteq$ PP.

4. Show that the following problem is in LOGSPACE.

MATCHED PARENTHESIS	
Input:	A word w over the alphabet $\Sigma := \{), (\}$.
Question:	Is every parenthesis in w properly matched?
	(e.g. "(()(()()))" is properly matched whereas "(()))(" is not.)