HW3, CPSC 468/568, Due Mar. 3, 2016

Throughout this assignment, if a proof or step of a proof follows directly from a definition given or a theorem proven in class or in a reading assignment, then you may simply say that, *i.e.*, you need not reproduce proofs given in class or in the reading.

Problem 1 *(20 points)*:

(a) (10 points) Prove the existence of a universal TM for space-bounded computation (analogous to the deterministic universal TM for time-bounded computation of Theorem 1.9). That is, prove that there exists a TM SU with the following property: For every string α and input x, if the TM M_α represented by α halts on x before using more than t cells of its (writable) work/output tapes, then $SU(\alpha, t, x) = M_\alpha(x)$.

(b) (5 points) Show that SU uses at most $C \cdot t$ cells of its work tapes, where C is a constant depending only on M_α.

(c) (5 points) Use parts (a) and (b) to prove Theorem 4.8, the Space Hierarchy Theorem.

Problem 2 *(15 points)*:

Suppose that we define NP-completeness using logspace reductions instead of polynomial-time reductions. Starting with the proof of the Cook-Levin Theorem, show that SAT and 3SAT are NP-complete under this alternative definition. Conclude that SAT is in L if and only if NP = L.

Problem 3 *(15 points)*:

Consider the certificate definition of NL (Def. 4.19 in your textbook). Prove that, if the verifier's tape head is allowed to move back and forth on the certificate, the complexity class being defined changes from NL to NP.

Problem 4 *(15 points)*:

(a) (5 points) In class, we proved Theorem 5.12 for the case of $i = 2$. Provide a proof of the general case, *i.e.*, prove that $\Sigma_i^P = \mathrm{NP}^{\Sigma_{i-1}\mathrm{SAT}}$ for all $i \geq 2$.

(b) (5 points) In class, we proved that, if P = NP, then P = PH. Show that, if $\Sigma_i^P = \Pi_i^P$, then $\mathrm{PH} = \Sigma_i^P$ and that, if $\Sigma_i^P = \Sigma_{i+1}^P$, then $\mathrm{PH} = \Sigma_i^P$.

(c) (5 points) Prove that, if EXACT-INDSET is Σ_2^P-complete, then the PH collapses.

Problem 5 *(10 points)*:

Recall that AP = PSPACE and that we do not know whether PH = PSPACE, what is wrong with the following proof that AP = PH?

We already know that PH $\subseteq$ PSPACE = AP. In order to prove that AP $\subseteq$ PH, choose an arbitrary language $L \in$ AP and consider an alternating polynomial-time machine M that accepts L. Suppose that M makes at most $i-1$ alternations, *i.e.*, at most $i-1$ transitions from an $\exists$ state to a $\forall$ state or from a $\forall$ state to an $\exists$ state. If M's start state is an $\exists$ state, we have $L \in \Sigma_i^P$; if it is a $\forall$ state, we have $L \in \Pi_i^P$. But PH $= \bigcup_{i \in \mathbb{N}} \Sigma_i^P = \bigcup_{i \in \mathbb{N}} \Pi_i^P$. In both cases, we have $L \in$ PH.

Problem 6 *(10 points)*:

Prove Claim 5.9, *i.e.*, prove that, for every $i \in \mathbb{N}$, $\Sigma_i^P = \cup_c \Sigma_i \text{TIME}(n^c)$ and $\Pi_i^P = \cup_c \Pi_i \text{TIME}(n^c)$ (where the classes are as defined in Defs. 5.3 and 5.8).

Problem 7 *(15 points)*:

Prove Theorem 6.15, *i.e.*, prove that L is in P if and only if L has polynomial-sized, logspace-uniform circuits.