CSE 532
Assignment 1
Due Thursday, January 22, 1997

1. This problem explores the technique of "polynomial padding". Let $L \subseteq \Sigma^*$ be a language. A *polynomial padding* of L is a language $L' = \{x\#^{|x|^k} : x \in L\}$ for some integer $k \geq 0$. The symbol $\#$ in not in the alphabet Σ.
   (a) Prove that NP is closed under polynomial padding. That is, if for all L if L' is a polynomial padding of L then $L \in NP$ if and only if $L' \in NP$.
   (b) Prove that SPACE(n) is not closed under polynomial padding. (Hint: You will need the space hierarchy theorem for this.)
   (c) Now argue that SPACE(n) $\neq$ NP.

2. Normally we would not expect that demonstrating a reduction would lead to a legitimate lower bound. In this problem you will show how this can be done in some cases. We start with the definition of a log-lin reduction. We say that L is log-lin reducible to L' if there is a function f that is computable in space $\log n$ and there is a constant $c > 0$ such that for all inputs x, $|f(x)| \leq c|x|$ and for all x, $x \in L$ if and only if $f(x) \in L'$. We say that L is *exponential time hard for log-lin reductions* if for all $L' \in \bigcup_{c>0}$ TIME(2^{cn}), L' is log-lin reducible to L. Prove that if L is exponential time hard for log-lin reductions then there is $c > 0$ such that $L \notin$ TIME(2^{cn}). (Hint: You will need the time hierarchy theorem for this.)

3. In this problem you will determine that any non-regular language requires space $\log\log n$ for infinitely many inputs. Let $L \notin \bigcup_{k\geq 1}$ SPACE(k). Prove that for any $s(n)$-space bounded deterministic Turing machine T that recognizes L, there is a constant $c > 0$ such that for infinitely many n, $s(n) > c\log_2\log_2 n$. (Hint: this is a somewhat tricky argument. It starts with assuming for each k that x is the shortest input that T uses at least k storage cells on input x. Then show $|x|$ is bounded above by a double exponential in k.)

4. There is a natural generalization of the halting problem to oracle Turing machines. Let T_x be the oracle Turing machine with binary description x. Define $L(T_x^A)$ to be the the language accepted by the oracle Turing machine T_x when supplied with the

oracle A. Define

$$K^A = \{x : x \in L(T_x^A)\}.$$

(a) Prove that K^A is not Turing reducible to A. This is the relativization of the halting problem. You may use the existence of a universal oracle Turing machine.

(b) Let ϕ_0 be the empty set and define $\phi_{k+1} = K^{\phi_k}$. Prove that the ϕ_i form an increasing sequence with respect to Turing reducibility. That is, $\phi_0 <_T \phi_1 <_T \phi_2 <_T \cdots$, where $A <_T B$ means that $A \leq_T B$ and $B \not\leq_T A$. Note that phi_1 is the normal halting problem for Turing machines.

(c) (Extra Credit) Prove that ϕ_k is many-one complete for Σ_k. For the definition of Σ_k see page 68 of Papadimitrio. A is many-one complete for a class of languages Γ if (i) $A \in \Gamma$ and (ii) for all $B \in \Gamma$, $B \leq_m A$.