Homework 2
(* denotes a hard problem and ** denotes an even harder problem)

§1 Recall that NC^k is the family of languages computed by Boolean circuits of fanin 2, polynomial size and depth $(\log(n))^k$. Also, NC^k is the family of languages computed by Boolean circuits of fanin 2, polynomial size and depth $(\log(n))^{O(1)}$. Recall that NL = NSPACE(log(n)) and L = DSPACE(log(n)).

(1.1) Describe an NC circuit for the problem of computing the product of two given $n \times n$ matrices *A, B*. (Should be easy; exploits parallelism)

(1.2) Describe an NC circuit for computing, given an $n \times n$ matrix, the nth powered matrix A^n. (Use recursive powering and repeat 1.1)

(1.3) Use (a) and (b) to show that the PATH problem (given a digraph, with vertices s and t, is there a directed path from s to t?). Explain why you can then conclude that every NL language is in NC. (Hint: recall PATH is a log-space-complete problem for NL)

***(1.4)** Prove that $NC^1 = L$. (Hint: Prove inclusion in each direction; think of the circuits of NC^1 evaluated sidewise in L, and think of L simulated by O(log(n) depth circuits.) Explain why you can thus conclude that PSPACE is not equal to NC^1.

§2 Recall that PH is the polynomial hierarchy: PH is the union of Σ_k^P for all k.

***(2.1)** Show for every $k > 0$ that PH contains languages whose circuit size complexity is $\Omega(n^k)$. (Hint: First show that such a language exists in $DSPACE(2^{poly(n)})$.)

***(2.2)** Show that Σ_2^P contains languages whose circuit size complexity is at least $\Omega(n^k)$ for every $k > 0$. (Hint: Keep in mind the lecture's proof of the existence of functions of high circuit size complexity.)

***(2.3)** Show for each $k > 0$ there is a language in PH that is not decidable by circuits of size (n^k). (Hint: Use diagonalization)

*§*3 Recall that P*/poly* is the family of languages of Boolean circuits of polynomial size. Also recall that EXP is the family of languages of deterministic TMs running in exponential time.

(3.1) Show that if EXP $\subseteq$ P*/poly* then EXP = Σ_2^P.
(Hint: Recall the lecture's proof that if NP $\subseteq$ P*/poly* then PH = Σ_2^P.)

***(3.2)** Show that if P = NP then there is a language in EXP that requires circuits of size at least $2^n/n$. (Hint: Again, keep in mind the lecture's proof of the existence of functions of high circuit size complexity.)

§4 A binary language $L \subseteq \{0, 1\}^n$ is *sparse* if there is a polynomial p such that $p(n)$ upper bounds the number of strings of L of length $\leq$ n for every $n \in$ N.

(4.1) Show that every sparse language is in P/poly. (Should be easy; exploits fact that there is a separate circuit for each input length n.)

****(4.2)** Show that if a sparse language L is NP-complete then P = NP.
(Hint: show a recursive exponential-time algorithm A such that on input of a n-variable Boolean formula F and a binary string x of length n, outputs 1 iff F has a satisfying assignment y of its Boolean variables such that x < y where both x and y are interpreted as the binary representation of a number between 0 and 2^n-1 . Use the polynomial time reduction from SAT to L to prune possibilities in the recursion tree of A.)