COMPUTATIONAL COMPLEXITY THEORY (COMS 30126)
EXERCISE SHEET 5

(1) Let $\mathcal{L}$ be a language decided by a circuit family $\mathcal{C} = (C_0, C_1, \ldots)$ with the following property: there is a *polynomial time* computation which on input 1^n, outputs a description of the circuit C_n.
Show that $\mathcal{L}$ is also decided by a *uniform* circuit family i.e. a circuit family $(C'_0, C'_1, \ldots)$ such that there is a log space transducer which on input 1^n outputs a description of C'_n.
(Hint: use theorem C of the lecture notes).

(2) Show that NC $\subseteq$ P.

(3) (a) Show that the task of computing the parity $x_1 \oplus \ldots \oplus x_n$ of an n-bit string $x_1 \ldots x_n$ is in NC^1.

(b) Let f : n-bits$\rightarrow$ 1-bit be the function defined as follows: $f(x_1 \ldots x_n) = 1$ if the string $x_1 \ldots x_n$ contains at least one 1 and $f(x_1 \ldots x_n) = 0$ if $x_1 \ldots x_n = 0 \ldots 0$. Show that the computation of f is in NC^1.
(Hint: note that $f(x_1 \ldots x_n) = \bigvee_i x_i$).

(c) Let $A = (A)_{ij}$ and $B = (B)_{ij}$ be m by m matrices whose entries are zeroes and ones (i.e. A and B are Boolean matrices). For any matrix A, $(A)_{ij}$ denotes the ij^{th} entry of A. Consider the computational task of *computing the Boolean matrix product* of A and B: the output is the matrix $C = AB$ defined by

$$(C)_{ik} = \bigvee_j (A)_{ij} \wedge (B)_{jk}.$$

(Note: this is the natural Boolean arithmetic equivalent of the usual notion of matrix multiplication). The input size for the computation is $n = 2m^2$ (i.e. all the entries of A and B).
Show that this computation has size-depth complexity $(O(n^{3/2}), O(\log n))$.
(Hint: start by considering gates that compute $(A)_{ij} \wedge (B)_{jk}$ for all i, j, k and then use the result of (b)).

(4) In this question we'll develop a result that is used in the lecture notes to show that NL $\subseteq \mathrm{NC}^2$: we will show that the PATH problem for directed graphs (cf INSERT1)is in NC^2.
If $A = (A)_{ij}$ is an m by m Boolean matrix, the *transitive closure* A_{tr-cl} of A is defined to be the matrix:

$$A_{tr-cl} = A \vee A^2 \vee \ldots \vee A^m$$

where A^k is the Boolean matrix product of A with itself, k times, and $A \vee B$ is the bit-wise OR of the matrix elements done at each entry ij separately.

(a) Suppose A is the adjacency matrix of a directed graph G.
(i) Show that the ij^{th} entry of A^2 is 1 iff there is a path of length 2 from node i to node j in G. (Hint: recall that $(A^2)_{ik} = \bigvee_j (A)_{ij} \wedge (A)_{jk}$, and what is the condition that the RHS is 1?)

Thus A^2 is the adjacency matrix of the graph G' which has an edge from node i to node k iff G has a path of length 2 from i to k.
(ii) Similarly show that $(A^l)_{ik} = 1$ iff there is a path of length l from node i to node k in G.
(iii) Conclude that the PATH problem (with input a graph G and two nodes i and j) can be solved by computing the transitive closure of the adjacency matrix of G.

(b) (i) Show that the computation of A^k can be organised as a binary tree of size $O(k)$ and depth $O(\log k)$ (with k instances of A as inputs) where each node computes the product of the two incoming matrices.
(ii) Deduce that the circuit computing A^m has size $O(n^2)$ and depth $O(\log^2 n)$ where $n = m^2$ is the input size of A. (Hint: use the result of question 3(c) for each node of the tree in (i)).
(iii) Deduce further that the size-depth complexity of computing the transitive closure of A is $(O(n^{5/2}), O(\log^2 n))$. (Hint: we make circuits for each A^i, $i = 1, \ldots, m$ which gives another factor of m and then an extra layer of depth $\log m$ for computing the OR in the transitive closure formula).