CS 351: Problem Set 4
(assigned 10/25/2002; due 10/31/2002)

On each problem, you may work with at most one other person, but you must write up the solutions independently. Please turn in each problem on a separate sheet of paper that also contains your name, the name of your collaborator (if any), and other sources (eg., papers, books) that you consulted.

(1) *Addition of two n-bit numbers in* NC^1*—how is that for a wicked sense of humor?*

Let A be a set and let $\odot : A^2 \longrightarrow A$ be an associative operator, that is, for all $a, b \in A$, $a \odot b \in A$, and for all $a, b, c \in A$, $(a \odot b) \odot c = a \odot (b \odot c)$. (Thus, $(A, \odot)$ is a semigroup; some examples of semigroups include $(\mathbf{N}, +)$, $(\{0,1\}^*, \cdot)$, where $\cdot$ denotes string concatenation, $(\mathcal{M}_{n\times n}, \star)$, where $\mathcal{M}_{n\times n}$ denotes the set of all $n \times n$ matrices over some field and $\star$ denotes matrix multiplication.)

The *prefix sum function for* $(A, \odot)$ is defined (for $n \geq 1$) by $\textsc{pref}_{\odot,n}(x_1, x_2, \ldots, x_n) = (y_1, y_2, \ldots, y_n)$, where $y_i = x_1 \odot x_2 \odot \ldots \odot x_i$, for $1 \leq i \leq n$.

(1a) Let A be a finite set, and let $\odot$ be an associative operator that can be computed by a single $\odot$-gate whose inputs are elements from A. Show that $\textsc{pref}_{\odot,n}$ can be computed by a circuit of $\odot$-gates of size $O(n^2)$ and depth $O(\log n)$.

For n-bit numbers $u = (u_{n-1}, \ldots, u_0)$ and $v = (v_{n-1}, \ldots, v_0)$ (where u_0 and v_0 are the least significant bits), define the following quantities, for $0 \leq i < n$:

$c_0 = 0$;
$p_i = u_i \oplus v_i$, where $\oplus$ is the XOR of two bits;
$g_i = u_i \wedge v_i$, where $\wedge$ is the AND of two bits;
$c_{i+1} = (p_i \wedge c_i) \vee g_i$, where $\vee$ is the OR of two bits;
$s_i = p_i \oplus c_i$;
$s_n = c_n$.

The bit c_i indicates the carry coming into position i (so c_{i+1} is the carry leaving position i), s_i is the bit of the sum in position i. The bits p_i and g_i are auxiliary variables—p_i indicates whether the carry from the previous stage (if any) is *propagated* through the i-th stage, and g_i indicates whether a carry is *generated* at the i-th stage (regardless of the previous carry).

Let $B = \{0, 1\}$. For $0 \leq i < n$, define $x_i = (p_i, g_i)$, and define the operator $\diamond : B^2 \longrightarrow B^2$ by $(a, b) \diamond (c, d) = (a \wedge c, (b \wedge c) \vee d)$. Let $(y_0, \ldots, y_{n-1}) = \textsc{pref}_{\diamond,n}(x_0, \ldots, x_{n-1})$, and for each i, let $y_i = (\pi_i, \chi_i)$.

(1b) Prove (by induction) that for all i, $\chi_i = c_{i+1}$.

(1c) Use (1b) to show that adding two n-bit numbers can be carried out by a circuit of polynomial size and $O(\log n)$ depth.

(1d) Returning to the computation of $\textsc{pref}_{\odot,n}$, show that there is an $O(\log n)$ depth circuit of $O(n)$ size (using $\odot$-gates) that computes this function. (*Hint.* Assume that n is a power of two, and recursively compute $\textsc{pref}_{\odot,n/2}(x_1 \odot x_2, x_3 \odot x_4, \ldots, x_{n-1} \odot x_n)$.)

(2) *Exercise.* (You don't have to turn this in—I'll spare you the torture—but do attempt it seriously, there is also an opportunity for extra credit.)

Take your solution to (1d), and describe a logspace algorithm that, on input n, prints a description of the circuit for $\textsc{pref}_{\odot,n}$.

If you think your logspace algorithm has a particularly concise and elegant description, do turn it in for extra credit.

(3) Show that counting the number of s–t paths in a directed, acyclic, layered (or leveled) graph can be done in NC^2.

(4) Let $f : \{0,1\}^n \longrightarrow \{0,1\}$. Consider the following communication complexity problem: Alice is given x such that $f(x) = 0$ and Bob is given y such that $f(y) = 1$ (obviously, therefore, $x \neq y$). Their goal is to find a position i such that $x_i \neq y_i$ (which exists since $x \neq y$).

Let R_f denote the above (promise) problem (it's called a promise problem because there is some additional promise about the input pair, namely that $f(x) \neq f(y)$), and let $cc(R_f)$ denote the communication complexity of R_f (that is, the communication cost of the best protocol for this problem, see Problem Set (1)).

Show that for any circuit C that computes f, $\mathit{depth}(C) \geq cc(R_f)$, that is, the communication complexity of R_f is a lower bound on the depth of any circuit that computes f. (*Hint.* Assume that Alice and Bob know the details of the circuit C, and derive a protocol for R_f.)

Notes. Indeed, it can be shown that $\mathit{depth}(C) = \Theta(cc(R_f))$! This is another instance where communication complexity touches upon some of the central questions in complexity theory.