CSE681, Spring 2010 **Problem Set 2** **Due Thu. 3/4**

Reading and Exercises:

In addition to ALR chapter 29 section 4 and the Barrington-Maciel "Advanced" lectures, please read the following entry in Richard Lipton's weblog: `http://rjlipton.wordpress.com/2009/07/18/graphs-permutations-characters-and-logspace/`

(1) Given a Boolean function $f(x_1, \ldots, x_n)$, define its "Boolean partial derivatives" for all i by

$$f_i = f[x_i = 0] \oplus f[x_i = 1].$$

Here $\oplus$ is exclusive-or, and $f[x_i = 0](a_1, \ldots, a_n)$ evaluates to the result of substituting 0 for a_i in the argument assignment. If $f_i(a_1, \ldots, a_n) = 0$, where we take 0 to mean false, then there is no sensitivity to a_i for the particular assignment $a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n$ to the other variables. For a particular assignment this is not unusual. But if $f_i(a_1, \ldots, a_n) = 0$ for *all* assignments, then there is no sensitivity to the variable x_i at all. (In communication complexity we will attend to an intermediate notion where the space of assignments is divided into "rectangles" of sensitivity.)

(a) Show that the Boolean analogue to the addition rule $\partial(f + g) = \partial f + \partial g$ holds, i.e. that for all i,

$$(f \oplus g)_i = f_i \oplus g_i.$$

(b) Show, however, that the corresponding analogue to the multiplicative rule $\partial(f \cdot g) = g\partial f + f\partial g$ *fails*. That is, give an example of f and g and i for which

$$(f \wedge g)_i \neq (g \wedge f_i) \oplus (f \wedge g_i).$$

(An identity that may help you is $f = (x_i \wedge f[x_i = 1]) \vee (\bar{x}_i \wedge f[x_i = 0])$, where you can also put $\oplus$ in place of the $\vee$. 3 + 12 = 15 pts.)

Meta-Open Problems: This is motivated by the question of associating to each n-ary Boolean function f an ensemble of functions $f_1, \ldots, f_n$ so that whenever f has formulas or circuits of size s, there are formulas/circuits of size $O(s)$ with n output gates o_i computing the respective functions f_i. All of the super-linear size lower bounds known for *arithmetical* circuits begin with this step, where the f_i are the partial derivatives of the arithmetical function f. The failure of (b) pole-axes the direct attempt to imitate this lower-bound strategy, however, and illustrates that for all the success that "arithmetizing" Boolean functions has had in complexity theory, the conversion is far from perfect. Still, the idea is enticing enough to attempt by other means, so the meta-open problem is to find a successful re-formulation of the multiplicative rule, or more generally an alternate definition of "$f_1, \ldots, f_n$" associated to a given Boolean function f for which the s-to-$O(s)$ conversion succeeds.

(2) An inequality of the form $\sum_i w_i x_i \geq t$, where $x_1, \ldots, x_n$ are Boolean variables and $w_1, \ldots, w_n$ and t are the given real-number parameters, defines a half-space in R^n. A conjunction of such equations defines a *polytope*. An arrangement of polytopes is said to be *in general position* if no two polytopes share an edge or face or vertex. An arrangement P of polytopes in general position defines a 2-coloring of R^n. Let us arbitrarily call the color of the point $(1/2, \ldots, 1/2)$ at the center of the Boolean cube "white" and the other color "black." Then P defines a Boolean function f on 0-1 valued assignments $\vec{a} = (a_1, \ldots, a_n)$ by $f(\vec{a}) = 1$ if $\vec{a}$ is black, 0 if $\vec{a}$ is white.

Finally define a language $L \subseteq \{0, 1\}^*$ to belong to the complexity class "PALT" if $L^{=n}$ is represented by arrangements P_n of polytopes in general position, such that each P_n has $n^{O(1)}$ member

polytopes. (Recall mention in lecture of a theorem by Minsky and Papert that any half-space H can be "adjusted" to a half-space H' with the same membership of vertices of the Boolean cube, such that each w_i and t needs only about $2n \log_2 n$ bits to specify. This implies that the whole ensemble of equations defining P_n can be written down with polynomially-many bits.)

(a) Show that PALT is contained in TC^0—here uniformity is ignored. For a hint, "PALT" stands for "Parity of Ands of Linear Thresholds." (You may, but need not, quote the theorem that a general linear threshold gate can be simulated by two levels of Majority gates with polynomial size blowup. 9 pts.)

(b) Show that PALT is closed under complements, with almost no blowup when going from P_n representing $L^{=n}$ to P'_n representing its complement. (3 pts.)

(c) Show that PALT is closed under intersection. (12 pts.)

(d) What is the blowup in the number of polytopes you get in (c) when going from P_n representing $A^{=n}$ and Q_n representing $B^{=n}$ to R_n representing $A^{=n} \cap B^{=n}$? Show that if the blowup were linear, i.e. $|R_n| = O(|P_n| + |Q_n|)$, then NC^1 would equal TC^0! But is it...? (21 pts., for 45 total on the problem)

(3) Define a list $B_n = (p_1, \ldots, p_s)$ of polynomials in $F[x_1, \ldots, x_n]$, where F is some field (or ring), to *represent* $L^{=n}$ if

$$L^{=n} = \{ \vec{a} \in \{0,1\}^n : \wedge_i p_i(a_1, \ldots, a_n) = 0 \}.$$

Here B_n is said to be a *basis* for and to *generate* the *polynomial ideal*

$$I_n = \{ \sum_i \alpha_i p_i : \alpha_1, \ldots, \alpha_s \in F[x_1, \ldots, x_n] \}.$$

Indeed, $L^{=n}$ consists of the vertices of the Boolean cube that lie in the *algebraic set* $V(I_n) = V(B_n)$ defined by the common zeroes of all polynomials in I_n. The letter V actually stands for "variety," a term older sources reserve for the case where V is *irreducible*, meaning that V cannot be written as a proper union of two other varieties in F^n. Naturally there is a notion of a sequence $[I_n]_{n=1}^{\infty}$ representing a language L, and complexity concerns about both the sizes $s = s_n$ and number of bits b_n needed to encode formulas for the polynomials in bases B_n for the respective ideals.

(a) Assuming $b_n = poly(n)$, show that the resulting language belongs to AC^1. It is OK to ignore uniformity, but you should pay attention to issues of numerical precision. (12 pts. In fact it belongs to NC^1, by a very hard theorem of Buss-Cook-Gupta-Ramachandran.)

(b) Suppose in addition that each p_i in bases B_n is given by a formula that is a sum-of-products-of-sums. Now show that the language is in TC^0. You may use the fact that n-fold multiplication of n-bit numbers is in TC^0. (12 pts.)

(c) Given B_n representing $L^{=n}$, show how to construct B'_n with a single polynomial representing $L^{=n}$, at most doubling the bit-size. If your B_n is given by $\Sigma\Pi\Sigma$ formulas as defined in part (b), can you get a $\Sigma\Pi\Sigma$ formula with linear blowup in size? (12 pts.)

(d) Noting that closure under union is easy, ditto taking negations of variables, show that if there is a construction for intersection that preserves $\Sigma\Pi\Sigma$ formulas with linear blowup in bit-size, then $NC^1 = TC^0$. (6 pts., for 42 total)

(4) Exercises 5. and 6. on page 7 of Barrington-Maciel lecture A1. (6 + 12 = 18 pts. total, for 120 total on the set)