Complexity of Boolean and Arithmetic Circuits - 236603

Exercise 1

Lecturer: Amir Shpilka *Hand in date: April 23rd*

Please write your solution in LaTeXand submit them by email to shpilka@cs.technion.ac.il.

1. In this question we consider a special kind of circuits: circuits that can only use $\wedge$ (and) gates, and are not allowed to use negations. We call such circuits $\wedge$-circuits. Clearly such circuits can only compute functions that are and over a subset of inputs. Let $C_\wedge(f)$ denote the size of the minimal $\wedge$-circuit computing f. We will be interested in functions from n inputs to n outputs. Such functions are determined by n subsets of our input variables: for subsets $(S_1, \ldots, S_n)$ we have that the i'th output is $\wedge_{j\in S_i} x_j$. For convenience we think of every such function as given by an $n \times n$ 0/1-matrix where the 1's in the i'th row tell us which indices belong to S_i. Given a matrix A we let $f_A : \{0,1\}^n \to \{0,1\}^n$ be the function determined by A.

   (a) Prove that there exists an A such that $C_{B_2}(f_A) = \Omega(\frac{n^2}{\log n})$. (notice that the question is about circuits over the basis B_2,[†] and not about $C_\wedge$).

   (b) Prove that for every A, $C_\wedge(f_A) = O(\frac{n^2}{\log n})$. Hint: partition the n inputs to sets of size $\log n$ and for each such subset compute the "and" of every one of its subsets.

   (c) i. Let S_i be the set defined by the i'th row of the matrix A. Prove that if for every $i \neq j$ we have that $|S_i \cap S_j| \leq 1$ then $C_\wedge(A) = \sum_{i=1}^n (|S_i| - 1)$.

   ii. Find n sets of size roughly $\sqrt{n}$ satisfying the conditions of the previous section (i.e. every two sets intersect in at most 1 element). Hint: every two lines intersect in only 1 point.

   iii. Prove that we cannot get a lower bound better than $\Omega(n^{3/2})$ using such an argument.

2. In this question we consider circuits over the basis $\{\wedge, \vee, \mathrm{NOT}\}$ (if you want you can think of circuits over the basis U_2). Remember that you get negations for free.

   (a) Let $count : \{0,1\}^n \to \{0,1\}^{\log n}$ be the function that writes in binary the number of 1-s in the input. Prove that $C(count) = O(n)$.

   (b) Let $sort : \{0,1\}^n \to \{0,1\}^n$ be the function that sorts the input bits (i.e. it first outputs the 1-s and then the 0-s). Prove that $C(sort) = O(n)$.

3. In class we proved that for any function f, $D(f) = \theta(\log L(f))$. I.e. we proved that $\log L(f) \leq D(f) \leq 2\log_{3/2} L(f)$ (this is what we get from the recursion that we wrote in class). We thus have that $D(f) \leq 3.42 \cdot \log_2 L(f)$.

   Improve the constant 3.42 as much as you can.

[†] Recall that this is the basis consisting of all functions of 2 variables