CSE431: Complexity Theory March 29, 2024

Homework 1

Anup Rao Due: April 5, 2024

Read the fine print[1]. Each problem is worth 10 points:

1. Show that every function $f : \{0,1\}^n \to \{0,1\}$ can be computed by a width 3 branching program. HINT: First express the function in conjunctive normal form: eg $(x_1 \vee \neg x_2 \vee x_3 \vee \neg x_4) \wedge (\neg x_1 \vee \neg x_2 \vee x_3 \vee x_4) \wedge \dots$.

2. Show that every function $f : \{0,1\}^n \to \{0,1\}$ can be computed by a circuit of size $O(2^n/n)$. HINT: Set $t = \log n - 1$, and do the recursive construction until there are t variables left to set. Then compute all functions of t variables with 2^{2^t} circuits and connect these components up appropriately.

3. Show that there is a constant c such that for n large enough, there is a function $f : \{0,1\}^n \to \{0,1\}$ that cannot be computed by a branching program with less than $\frac{2^n}{cn}$ nodes.

4. In class we discussed the difficulty (and appeal) of showing that every small circuit can be simulated by a small-depth circuit. Here we explore how to prove this for *formulas*. A formula is a circuit where every gate has out-degree at most 1. In other words, each gate is used as an input for at most 1 gate. (We are allowed to duplicate gates that correspond to input variables in circuits, so input variables can be read more than once in a formula.)

   In this problem, we shall show that if $f : \{0,1\}^n \to \{0,1\}$ can be computed with a formula of size s, then it can be computed by a formula of depth $O(\log s)$.

   Since we are interested in showing that depth is asymptotically bounded by $O(\log s)$, it is enough to handle the case when $s > 6$, say.

   For every gate g of the formula, let $c(g)$ be the number of input gates used to compute g in the formula. First, show that there must be a gate g in the formula such that $s/3 \leq c(g) \leq 2s/3 + 1$. HINT: Define a sequence of gates $g_1, g_2, \dots$ as follows. Start by setting g_1 to be the output gate of the formula, so $c(g_1) = s$. For each i, if g_i is an input gate (so $(c(g_i) = 1)$, we end the sequence of gates. Otherwise, if h, q are the two gates that feed into g_i, we set g_{i+1} to be the gate that maximizes $c(g_{i+1})$. Since we have $c(g_i) = c(h) + c(q) + 1$, this choice ensures that $c(g_{i+1}) \geq (c(g_i) - 1)/2$. Show that some gate in this sequence has the desired property.

   Next, complete the proof by using the gate g found above and induction to give a formula of size $O(\log s)$.

[1] In solving the problem sets, you are allowed and even encouraged to collaborate with fellow students taking the class, but **each submission can have at most one author**. If you do collaborate in any way, you must acknowledge, for each problem, the people you worked with on that problem. The problems have been carefully chosen for their pedagogical value, and hence might be similar to those given in past offerings of this course at UW, or similar to other courses at other schools. Using any pre-existing solutions from these sources, for from the web, constitutes a violation of the academic integrity you are expected to exemplify, and is strictly prohibited. Most of the problems only require one or two key ideas for their solution. It will help you a lot to spell out these main ideas so that you can get most of the credit for a problem even if you err on the finer details. Please justify all answers. Some other guidelines for writing good solutions are here: `http://www.cs.washington.edu/education/courses/cse421/08wi/guidelines.pdf`.