Lecture 13. A Lower Bound Proof for a Concrete Problem

It is quite difficult to show a nontrivial lower bound for a given concrete problem, and unfortunately, not only NP-complete problems but almost all known problems, we have been unable to prove show an absolute lower bound on their computational complexity. Here we review one famous example for which we could give a reasonable computational lower bound analysis. An explanation given below is basically from Section 6.6 of [1].

13.1 Circuit model and our target problem

We study a lower bound for computing "parity" by "constant depth" circuits. Here we first recall a circuit model and introduce necessary notions and notation.

We may consider a circuit as a device for computing a Boolean value at its *output gate* (the last internal gate located at the top of the circuit) from Boolean values given to its *input gates* (located at the bottom of the circuit), by computing each value of an internal gate from the bottom to the top. For internal gates, we consider the stadard AND, OR, and NOT gates[1]. For a given circuit C with n input gates, let $C(x_1, \ldots, x_n)$ denote a Boolean function computed by C in this way.

Here we consider the task of the following simple function:

$$\mathrm{par}_n(x_1, \ldots, x_n) \;=\; \sum_{1 \le i \le n} x_i \bmod 2 \;=\; x_1 \oplus x_2 \oplus \cdots \oplus x_n.$$

By "computing parity" we mean to compute par_n for all $n \ge 0$. More formally, we consider a family $C = \{C_n\}_{n \ge 0}$ of circuits such that each C_n computes each member of the family $\mathrm{par} = \{\mathrm{par}_n\}_{n \ge 0}$, which is simply said that C computes par. By $\overline{\mathrm{par}}$ we mean the negation of par; that is, $\overline{\mathrm{par}}_n(x_1, \ldots, x_n) = 1 - \mathrm{par}_n(x_1, \ldots, x_n)$.

For a circuit model, there are two essential complexity measures; namely, size and depth. For a circuit family C, these complexity measures are defined as follows:

$$\begin{aligned} \mathrm{size}(C)(n) &= \text{the number of internal gates of } C_n, \text{ and} \\ \mathrm{depth}(C)(n) &= \text{the depth of } C_n. \end{aligned}$$

By "depth of C_n" we mean the number of internal nodes on the longest path from its output gate to some of its input gate. Intuitively, the depth complexity measures the parallel computation time of the computation expressed by a given circuit. Below we often write, e.g., $\mathrm{size}(C)(n)$ as "$\mathrm{size}(C)$." (Soon later we will not count NOT gates for discussing the size and depth measures.)

It is easy to see that all NOT gates are moved to the bottom (just after the input gates) without increasing the size or depth of a circuit. Thus, we generalize the notion of an input gate so that an input gate is either a Boolean variable x_i or its negation $\overline{x_i}$, and we assume that all internal gates are either AND or OR gates. We also allow to use an *unbounded fan-in gate*; that is, we put no bound on the number of inputs of each AND or OR gate. Thus, two consequtive AND (or OR) gates can be merged to one, and we may

[1] It can be shown that these three gate types are enough for computing all computable functions.

assume that a circuit is *leveled*, that is, its bottom layer is it input gates, the first layer consists of all OR (or AND) gates, the 2nd layer consists of all AND (resp., OR) gates, and so on. Clearly, a given circuit can be restructured as a leveled one without increasing its size or depth. In the following, by a circuit we consider only such an unbounded fan-in and leveled circuit.

Note the following fact on the circuit complexity of the parity function.

Theorem 13.1 There is a $O(\log n)$-depth and polynomial-size circuit computing the parity. More formally, there is a family C of circuits that computes a family of parity functions with the following complexity bounds.

$$\text{size}(C) \;=\; n^{O(1)}, \;\; \text{and} \;\; \text{depth}(C) \;=\; O(\log n).$$

(In fact, we can show that $\text{size}(C) = O(n)$.)

Thus, the parity is relatively easy to compute. But what if we consider only constant depth circuits. Would it be possible to design circuits computing the parity in some constant parallel time? Well, it is indeed possible by using exponential number of gates. In fact, we can compute any Boolean function by depth 2 circuits. But this in general requires exponential number of gates. A problem of interest (and also very important) to us is whehter it is possible to compute the parity by constant depth *and* polynomial-size circuits. This is the question we will discuss below. Let AC_0 denote the class of Boolean functions (or decision problems) that can be computed/solved by a consntant-depth and polynomial-size circuit family. Our question is whether the parity function par (resp., $\overline{\text{par}}$) is in the class AC_0.

13.2 A lower bound result by the switching lemma

We can show that par, $\overline{\text{par}} \notin \text{AC}_0$. More precisely, we have the following lower bound result. (The fact par $\notin \text{AC}_0$ was proved first by Furst, Saxe, and Sipser in 1984. The following version is due to Håstad proved in 1986. See [2] for some more explanation on further improvements and related topics.)

Theorem 13.2 There exists some constant c_0 satisfying the following: For any integer $d \geq 1$, for any circuit family C of depth $\leq d$ that computes either par or $\overline{\text{par}}$. Then we have $\text{size}(C) \geq \exp(c_0 n^{1/(d-1)})$ for all $n \geq 1$.

For this lower bound analysis, they (i.e., Furst, Saxe, and Sipser, and later Håstad) introduced a "random restriction", and developed an important lemma that is now called a *switching lemma*. We study the version explained in [1].

Rererences:

1. D. Du and K. Ko, Theory of Computational Complexity (2nd edition), John Wiley and Sons, Inc., 2000, ISBN:978-1-118-30608-6.
2. S. Tamaki and O. Watanabe, Local restrictions from the Furst-Saxe-Sipser paper, *Theory of Computing Systems*, 60(1): 20-32, 2017.

Homework exercise from Lecture 13

Homework rule: The following is another advanced problem that you can choose to solve from three lectures from Nov. 6th. Among those problems, choose one of these problems (from six problems I will give from this and next two lectures) and submit your report by Nov. 24th (Friday) noon to Watanabe's mail box in the mail box room of the West 8E building. You can submit your report by email. You can get 3 points by submitting an OK report. Please do not solve more than one problem.

One Additional Important Rule
(This applies to all documents you will produce in your future career.)
When you use any information or any help for writing your document, you have to clearly state its source (and acknowledge it if appropriate) in your document.

Advanced Problems

1. (You can solve this problem, I think, by yourself.)
   Show that the parity can be computed by some depth d and size $\exp(O(n^{1/(d-1)}))$ circuit family. (Thus, Theorem 13.2 is in a sense optimal.)

2. Give your explanation on the proof of the switching lemma in [1]. In particular, explain the following points.
   (1) Why do we need to consider the generalized conditional probability? Is there any problem if we simply try to prove the target probability directly?
   (2) Explain the derivation of the equations (6.7) and (6.8).