Notes for Lecture 26

Circuit Lower Bounds for Parity Using the Switching Lemma

In this lecture we give an alternate proof that parity $\notin \mathbf{AC}^0$ using the technique of random restrictions. This is the original method that was used to prove parity $\notin \mathbf{AC}^0$.

We will describe a proof of the following tight result.

Theorem 1 *If C is a circuit of size S and depth d that computes the parity of n inputs, then*

$$S \geq 2^{\Omega(n^{1/d-1})}$$

1 Conventions About the Circuit

Convention 1 *We count NOT gates neither towards the size nor towards the depth of the circuit. (This only makes the lower bound stronger.)*

Lemma 2 *If C is a circuit of size S and depth d, then there is a circuit C' of size at most $2S$ and depth d that computes the same function and* such that all the not gates are applied at the input level.

Proof: We prove the stronger statement that, for every gate g of C, there is a gate g' in C' whose output is the complement of g. Then we just let the output of C' be the complement of the output gate of C.

Let us order the gates of C as $g_1, \ldots, g_S$ in such a way that if the gate g_i uses the output of gate g_j as an input then $j < i$. We describe an inductive construction. Regarding g_1, if g_1 is a AND gate (respectively, an OR gate), then g'_1 is an OR gate (respectively, an AND gate), whose inputs are the complements of the inputs of g_1. It follows from De Morgan's law that the output of g'_1 is the complement of the output of g_1. Now, if we have constructed gates $g'_1, \ldots, g'_i$ whose outputs are the complement of $g_1, \ldots, g_i$, then it's easy to define g'_{i+1} using the same idea. □

Convention 2 *From now on, we restrict ourselves to circuits where the NOT gates are only applied at the input level. By Lemma 2, this only affects the lower bound by a multiplicative factor of two.*

Lemma 3 *If C is a circuit of size S and depth d, then there is a circuit C' of size at most dS and depth d that computes the same function and such that*

- *The gates are arranged in d layers, so that a gate at layer i takes inputs only from gates at layer $i-1$. The gates at layer 1 take as inputs only the inputs of the circuit.*

- *Each layer contains either only OR gates or only AND gates. Layers with OR gates and layers with AND gates alternate.*
- *If C satisfied Convention 2, then C' also satisfies Convention 2.*

Proof: Using the associativity of AND and OR, we can make sure that, in every input-output path in the circuit, we always see an alternation between AND gates and OR gates. Said another way, we can make sure that the inputs to each AND gates are coming only from OR gates (or from inputs to the circuit), and vice versa. Suppose that there is an AND gate g in the circuit one of whose inputs is coming from another AND gate g': then we can connect the inputs of g' directly to g. This does not change the size or depth of the circuit, and it reduces the number of "violations" of the above property. By repeated applications of the same rule, we eventually get a circuit of the same size and depth of C and such that each AND gate take only inputs from OR gates (and possibly from inputs of the circuit) and vice versa.

Finally, we arrange the gates in d layers so that each layer is made of either all AND or all OR gates, and wires go only from lower-numbered layers to higher-numbered layers. Finally, we replace all wires that skip many layers by a path of alternating fan-in 1 AND gates and fan-in 1 OR gates. This step increases the size of the circuit by at most a factor of d. □

Convention 3 *From now on, we restrict ourselves to ciruits as in the conclusion of Lemma 3.*

2 Random Restrictions and Sketch of the Lower Bound Proof

A *restriction* fixes some inputs of a circuit to constant values, and leaves other inputs free. More formally, a restriction is a function $\rho : \{1, \dots, n\} \to \{0, 1, *\}$. Applying a restriction ρ to a circuit C with n inputs defines a restricted circuit C_ρ as follows:

- For each $i \in \{1, \dots, n\}$:
  - If $\rho(i) = 0$, set the ith input of C to 0.
  - If $\rho(i) = 1$, set the ith input of C to 1.
  - If $\rho(i) = *$, leave the ith input of C as a free variable.

The proof of Theorem 1 has, roughly, the following structure. First, we show that parity circuits of depth 2 must have exponential size and use gates of linear width (Lemma 4). Next, we suppose by way of contradiction that we have a circuit C of some constant depth d which computes parity. We give a way of squashing C down to depth $d - 1$ while still computing parity on many variables. We can repeat the method $d - 2$ times to obtain a parity circuit of depth 2 and sublinear width, which contradicts Lemma 4. (The actual proof, indeed, will be a direct one, and we will not need to argue by contradiction.)

Restrictions enter in our method of circuit squashing. We show that, after a random restriction, the top two layers of the circuit can be replaced by an equivalent set of gates but with a switched order of AND and OR gates. That is, if, before the restriction the top layer had AND gates and second layer had OR gates, then, after the restriction, the first two layers can be equivalently realized with OR gates at the first layer and AND gates and the second layer, with no significant size increase.[1] Using associativity, we can then collapse the second and the third layer, obtaining a depth $(d-1)$ circuit.

3 Proof of the Lower Bound

We begin with the simple case of depth 2.

Lemma 4 *If a DNF or a CNF computes parity of n variables, then:*

1. *Each term includes all n variables, and*
2. *There are at least 2^{n-1} terms.*

Proof: We will prove the lemma for CNFs, which have OR gates at their top level and a single AND of all the ORs at the second level. The proof for DNFs is quite similar.

For any CNF circuit C:

1. *Each term includes all n variables:* Suppose by way of contradiction that C has some term t which does not depend on some variable x_i. Then when all inputs to t are 0, t outputs 0 and the single AND gate on the next level outputs 0, which is the output of the whole circuit. Now flip the value of x_i. The output of t is still 0, and thus the output of C has not changed. But since we've only changed one variable, the parity has flipped. Alas, we have a contradiction! So every term must depend on all variables.

2. *There are at least 2^{n-1} terms:* To compute parity, C must output 0 on 2^{n-1} different settings of the input variables. C outputs 0 only when one of the terms (OR gates) outputs 0. But each OR gate outputs 0 on exactly one setting of the input variables. Thus, C must have at least 2^{n-1} terms.

□

The following result is the technical core of the lower bound. It has a difficult proof that we omit

Lemma 5 *(Switching Lemma) Suppose f is a k-CNF or k-DNF over the variables $x_1, \ldots, x_n$. Pick at random a restriction that leaves a fraction p of the variables unfixed. For each of the $n(1-p)$ variables that are fixed, independently hardwire 0 or 1 as that variable's value. Then for every t,*

$$\mathbf{Pr}[\textit{after the restriction } f \textit{ can be expressed as a decision tree of depth } t\ ] > 1 - (7pk)^t.$$

[1] In the actual proof, it will be convenient to just keep track of the width of the top gates and of the size of the circuit not counting the first layer, instead of the total size. Through the construction, neither of these parameters will increase at all.

Notice that if a function can be specified by a depth-t decision tree then, for a stronger reason, it can be specified by a t-CNF and also by a t-DNF. We proceed with the proof of Theorem 1.

Proof:[Of Theorem 1] Let C be a depth d circuit for parity of size S, satisfying the conventions specified earlier.

Consider the gates at the first level, and suppose they are OR gates (a symmetric argument applies if they are AND gates). We think of each such gate as a 1-DNF formula. We apply the Switching Lemma with $t = \log S$ and $p = 1/14$, and we deduce that there relative to a random restriction each of the top-level gates becomes a $\log S$-CNF with probability bigger than $1 - 1/S$. In particular, there is a random restriction that makes all top-level gates expressible as a $\log S$-CNF formula. We apply such a restriction, we substitute each top gate by a $\log S$-CNF, and finally we use associativity to collapse the AND gate of each CNF into the AND gates of the second level of the original circuit.

Now we have a circuit of depth d such that each top gate has fan-in at most $\log S$, there are at most S gates from level 2 to level d, and the circuit computes parity of $n/14$ variables.

Now we apply the Switching Lemma with $k = \log S$, $p = 1/(14 \log S)$ and $t = \log S$. We get that, for each of the AND gates at level 2, after the restriction the gate can be replaced by a $(\log S)$-DNF with probability more than $1 - 1/S$. Then, there is a restriction for which this is true for all the at most S gates at level 2, we apply this restriction, we replace each level-2 gate with a $(\log S)$-DNF, and we collapse level 2 with level 3.

Now we have a circuit of depth $d - 1$ that computes parity of $n/(14 \cdot (14 \log S))$ bits, and such that every top gate has fan-in at most $\log S$ and there are at most S gates from level 2 to level $d - 1$.

If we repeat the same argument another $d - 3$ times, we end up with a circuit of depth 2 such that the fan-in of the top gates is at most $\log S$ and the circuit computes parity of $n/(14 \cdot (14 \log S)^{d-2}$ inputs. From Lemma 4 we have

$$\log S \geq n \cdot \frac{1}{14 \cdot (14 \log S)^{d-2}}$$

which is equivalent to

$$S \geq 2^{\frac{1}{14} n^{1/d-1}} .$$

□

4 References

The idea of using random restrictions, as well as the first proof that Parity is in $\mathbf{AC}^0$, is due to Furst, Saxe and Sipser [FSS84]. The lower bound was improved to exponential by Yao [Yao85], and the optimal lower bound is due to Håstad [Hås86].

References

[FSS84] Merrick L. Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984. 4

[Hås86] Johan Håstad. Almost optimal lower bounds for small depth circuits. In *Proceedings of the 18th ACM Symposium on Theory of Computing*, pages 6–20, 1986. 4

[Yao85] Andrew C Yao. Separating the polynomial-time hierarchy by oracles. In *Proceedings of the 26th IEEE Symposium on Foundations of Computer Science*, pages 1–10, 1985. 4