Lecture 16

Circuit Lower Bounds for NP problems

May 27, 2004
Lecturer: Paul Beame
Notes: Nilesh Dalvi

In the few years after the definitions of NP-completeness, there was some hope that techniques such as diagonalization from recursive function theory would be able to resaolve the question. However, those hopes were dashed in the late 1970's by the following construction.

Theorem 16.1 (Baker-Gill-Solovay). *There exists oracles A, B such that*

$$\mathsf{P}^A = \mathsf{NP}^A, \mathsf{P}^B \neq \mathsf{NP}^B$$

Since diagonal arguments generally work even when the machines involved are given access to oracles, this theorem suggests that diagonalization cannot help in deciding if $\mathsf{P} = \mathsf{NP}$ or $\mathsf{P} \neq \mathsf{NP}$.

Throughout the 1970's, there was also more focus by Cook and others on approaching the P versus NP question via the following containments of complexity classes

$$\mathsf{L} \subseteq \mathsf{NL} \subseteq \mathsf{P} \subseteq \mathsf{NP}.$$

This led to the use of more restrictive log-space reductions instead of polynomial-time reductions and to look at which problems could be solved in both polynomial time and polylogarithmic space with a view to separating classes such as L from NP. This led to the naming of the following classes of languages which eventually came to be named after Steve Cook.

Definition 16.1.

$$\begin{aligned} \mathsf{SC}^k &= \mathsf{TIMESPACE}(n^{O(1)}, \log^k n) \\ \mathsf{SC} &= \cup_k \mathsf{SC}^k \qquad \text{["Steve's Class" after Steve Cook].} \end{aligned}$$

Open Problem 16.1. Is $\mathsf{SC} = \mathsf{P}$? Is $\mathsf{NL} \subseteq \mathsf{SC}$?

In the late 1970's in part because of the proved weakness of diagonalization above, the study of non-uniform complexity in general and circuits in particular rose to prominence. In particular, both for complexity-theoretic reasons and for understanding the power of parallel computation, the following complexity class analogues of SC were suggested.

Definition 16.2.

$$\begin{aligned} \mathsf{NC}^k &= \mathsf{SIZEDEPTH}(n^{O(1)}, O(\log^k n)) \\ \mathsf{NC} &= \cup_k \mathsf{NC}^k \qquad \text{["Nick's Class" after Nick Pippenger].} \end{aligned}$$

If each gate has a constant time delay, problems solvable in NC can be solved in polylog time using a polynomial amount of hardware. Both to understanding how one would actually build such parallel machines it is natural to define uniform versions of the NC circuit classes, which express how easy it is to build the n-th circuit. There are many variants of such uniform complexity classes:

polytime uniform	: there is a TM that on input 1^n outputs the n^{th} circuit in time $n^{o(1)}$
log-space uniform	: there is a TM that on input 1^n outputs the n^{th} circuit using space $O(logn)$ or, equivalently, there is a TM that given a triple (u, v, op) of gate names u and v and an operation op determines whether or not u is an input to v and gate v is labeled by op and operates in linear space in the size of its input.
FO uniform	: the language (u, v, op) as above can be recognized by a first-order logic formula.

Theorem 16.2. *The following containment holds*

$$\textit{log-space uniform } \mathsf{NC}^1 \subseteq \mathsf{L} \subseteq \mathsf{NL} \subseteq \mathsf{NC}^2$$

Proof sketch. log-space uniform $\mathsf{NC}^1 \subseteq \mathsf{L}$: An NC^1 circuit has $O(\log n)$ depth. A log-space machine can evaluate the circuit by doing a depth-first traversal using stack height at most $O(\log n)$ and accessing the gates as needed using the log-space constructibility of the circuit as needed. in log-space and, the circuit can be evaluated.

($\mathsf{NL} \subseteq \mathsf{NC}^2$) We show that directed graph reachability can be computed in NC^2. Graph reachability can be computed by using $\wedge - \vee$ matrix powering to compute transitive closure. This can be computed efficiently using repeated squaring.

$$A \to A^2 \to A^4 \to \cdots \to A^{2^{\log n}} = A^n$$

where A is the adjacency matrix. Each matrix squaring can be performed in $O(\log n)$ depth and polynomial size since there is a simple $O(\log n)$ depth fan-in circuit computing $\bigvee_{k=1}^{n}(a_{ik} \wedge a_{kj})$. Thus, graph reachability can be performed in $O(\log^2 n)$ depth and polynomial size. □

Open Problem 16.2. Is $\mathsf{NP} \not\subseteq \mathsf{NC}^1$? Even more specifically it is consistent with our current knowledge that $\mathsf{NP} \subseteq \mathsf{SIZEDEPTH}(O(n), O(\log n))$!

Additional Circuit Complexity Classes in NC

Definition 16.3. Define $\mathsf{AC}{-}\mathsf{SIZEDEPTH}(S(n), d(n))$ to be the circuit complexity class with appropriate size and depth bounds that allows unbounded fan-in $\vee$ and $\wedge$ gates in addition to binary fan-in $\vee$ and $\wedge$ gates. [The AC stands for "alternating class" or "alternating circuits".]
Define $\mathsf{AC}^k = \mathsf{AC}{-}\mathsf{SIZEDEPTH}(n^{O(1)}, O(\log^k n))$.

Analogously, we define $\mathsf{AC}[p]{-}\mathsf{SIZEDEPTH}(S(n), d(n))$ and $\mathsf{AC}^k[p]$ where one also allows unbounded fan-in $\oplus_p$ gates, where

$$\oplus_p(x_1, \cdots x_n) \begin{cases} 0 & \text{if } \sum x_i \equiv 0 \pmod p \\ 1 & \text{if } \sum x_i \not\equiv 0 \pmod p. \end{cases}$$

and $\mathsf{ACC}{-}\mathsf{SIZEDEPTH}(S(n), d(n))$ and ACC^k where where unbounded fan-in $\oplus_p$ gates for any values of p are allowed. [ACC stands for "alternating circuits with counters".

Finally, define threshold circuits $\mathsf{TC}{-}\mathsf{SIZEDEPTH}(S(n), d(n))$ and TC^k to allow threshold gates T_m^n, where

$$T_m^n(x_1, \cdots x_n) = \begin{cases} 1 & \text{if } \sum x_i \geq m \\ 0 & \text{otherwise.} \end{cases}$$

These circuits are important since TC^0 corresponds to bounded-depth neural networks.

Lemma 16.3. *Following containments hold*

$$\mathsf{NC}^0 \subseteq \mathsf{AC}^0 \subseteq \mathsf{AC}^0[p] \subseteq \mathsf{ACC}^0 \subseteq \mathsf{TC}^0 \subseteq \mathsf{NC}^1$$

Proof. All of the containments follow easily from definitions. For example $\mathsf{ACC}^0 \subseteq \mathsf{TC}^0$ because count can be implemented by threshold gates. □

Additionally, there is the following non-trivial containment:

Lemma 16.4. $\mathsf{NC}^1 \subseteq \mathsf{AC}{-}\mathsf{SIZEDEPTH}(n^{O(1)}, O(\log n/\log\log n))$.

Proof. Break the NC^1 circuit into $O(\log n/\log\log n)$ layers of depth $m = \log\log n$ each. Eac gate at the boundaries between the layers can be expressed as a binary fan-in circuit with at most 2^m inputs from the boundary gates of the previous layer. Any function on $M = 2^m$ inputs can be expressed as a (depth-2) DNF formula of size $M2^M = 2^m 2^{2^m} = O(n\log n)$ so we can replace the circuitry between each layer by the appropriate unbounded fan-in circuitry from these DNFs, retaining polynomial size but reducing depth by a factor of $\frac{1}{2}\log\log n$. □

The following are the two main theorems we will prove over the next lecture and a half. As stated, the latter theorem is stronger than the former but the proof techniques for the former yield sharper bounds and are interesting and useful in their own right.

Theorem 16.5 (Furst-Saxe-Sipser, Ajtai). *Parity, $\oplus_2$, is not in* AC^0.

Theorem 16.6 (Razborov, Smolensky). *Let $p \neq q$ be primes. Then $\oplus_p \notin \mathsf{AC}^0[q]$.*

Corollary 16.7. $\oplus_p \notin \mathsf{AC}^0[q]$ *where q is a prime power and p contains a prime factor not in q.*

For the rest of this lecture we give the proof of Theorem 16.5.

Intuition: For an unbounded fan-in $\vee$ gate, setting any bit to 1 fixes the output. In an unbounded fan-in $\wedge$ gate, setting any bit to 0 fixes the output. However, for a parity gate, all the inputs need to be fixed to determine the output. Therefore, set bits to simplify the AC^0 circuit (and eventually fix its value) while leaving some bits unset which ensure that the circuit cannot compute parity.

Definition 16.4. Define a *restriction* to be a function $\rho : \{1, \cdots n\} \to \{0, 1, *\}$, where

$$\rho(i) = \begin{cases} 0 & \text{means that variable } x_i \text{ is set to 0,} \\ 1 & \text{means that variable } x_i \text{ is set to 1, and} \\ * & \text{means that variable } x_i \text{ is not set.} \end{cases}$$

Let $*(\rho) = \rho^{-1}(*)$ denote the set of variables unset by ρ.
Define $f|_\rho$ or $C|_\rho$ as the simplification of the function or circuit that results from applying the restriction ρ.

Definition 16.5. Define $\mathcal{R}_p$ to be a probability distribution on the set of restrictions such that for each i, the probabilities of $\rho(i)$ being *, 0 and 1 are p, $\frac{1-p}{2}$ and $\frac{1-p}{2}$ respectively and are independent for each i.

Lemma 16.8 (Hastad's Switching Lemma). *Let $0 \leq p \leq 1$ and let F be an s-DNF formula, i.e., having terms of length at most s. For $\rho \in_R \mathcal{R}_p$,*

$$\Pr[F|_\rho \textit{ cannot be expressed as a } t\textit{-CNF}] < (5ps)^t.$$

The proof of this lemma is too long to present here but some useful intuition is in order. Suppose we examine the terms of F, one by one. Any clause that is not set to by ρ leaves an s-DNF remaining without that clause, which is a problem of essentially the same type as before. Given that the term is not set to 0 then every variable in the term is either unset or set according to its sign in the term. Given this, it has only a roughly $2p$ chance that it is unset versus set according to the term. Therefore the expected number of unset variables in any term is at most $2ps$ and it is very unlikely that more than one will be found in any term. Of course the above argument ignores all sorts of probability conditioning which yields the extra .

Corollary 16.9. *Let $f : \{0,1\}^n \to \{0,1\}$ be in* AC$-$SIZEDEPTH(S,d). *Then, there exists ρ such that $|*(\rho)| \geq n/(10^d(\log S+1)^{d-1})$ and $f|_\rho$ can be expressed as a $(\log S+1)$-DNF or CNF.*

Proof. Without loss of generality assume that all negations are at leaves and $\vee$ and $\wedge$ alternate.

The general idea is to apply the Hastad switching lemma to the subcircuits of the circuit computing f that are nearest the inputs (usually called the bottom level of the circuit). At the bottom level, the functions at the $\vee$ (resp. $\wedge$) gates are switched to $\wedge$ (reps. $\vee$) gates and merged with the level above.

In general, in applying the Hastad switching lemma, the argument will maintain $s = t = \log S + 1$ and set $p = \frac{1}{10(\log S+1)} = \frac{1}{10s}$. In this case

$$(5ps)^t = 2^{-t} = 2^{-\log S-1} = \frac{1}{2S}.$$

At the start of the argument however, the bottom level $\wedge$ or $\vee$ gates correspond to 1-DNF or 1-CNFs so one begins with $s = 1$ and $t = \log S+1$. In this first step $p = \frac{1}{10}$ is good enough to yield a $\frac{1}{2S}$ failure probability at most.

Let $i = 1$. For each gate g at the bottom level, the probability that g doesn't simplify under $\rho_i \in_R \mathcal{R}_p$ is less than $\frac{1}{2S}$. There are at most S such gates; so, the probability that there is some gate that doesn't simplify is less than $1/2$.

Note that $|*(\rho_i)|$ is a binomially distributed random variable with mean $E[|*(\rho_i)|] = pn$. Because when $p(1-p)n \to \infty$ the binomial distribution behaves in the limit like a normal distribution a constant fraction of its weight is above the mean, so we have $\Pr[|*(\rho_i)| \geq pn] \geq 1/3$.
Therefore $\Pr[\rho_i$ has $|*(\rho_i)| \geq pn$ and circuit depth shrinks by $1] \geq 1/6$. Hence, by probabilistic method there exists a ρ_i that has these properties. Fix it and repeat for $i+1$, reducing depth every time. This gives us a combined restriction ρ which is the composition of all the ρ_i and has the desired properties. □

Theorem 16.10. *Any* AC *circuit computing parity in size S and depth d has $S \geq 2^{\frac{1}{10}n^{1/(d-1)}} - 1$.*

Proof. To compute parity, we need

$$\begin{aligned} *(\rho) &\leq \log S + 1 \\ \Rightarrow \frac{n}{10^d(\log S+1)^{d-1}} &\leq \log S + 1 \\ \Rightarrow n &\leq 10^d(\log S+1)^d \\ \Rightarrow S+1 &\geq 2^{\frac{1}{10}n^{1/d}} \end{aligned}$$

To obtain a stronger result, observe that the subcircuits of depth $d-1$ that are combined to produce the parity function also require terms/clauses of size equal to the number of unset variables. Therefore we can apply the above argument to the depth $d-1$ subcircuits of the parity circuit and replace d by $d-1$. □

Note that for the size to be polynomial this requires depth $d = \Omega(\log n / \log \log n)$.

The above argument is essentially tight since parity can be computed by AC circuits of depth d and size $2^{O(n^{1/(d-1)})}$.