Lecture 4

Circuit Complexity and the Polytime Hierarchy

April 8, 2004
Lecturer: Paul Beame
Notes: Ashish Sabharwal

So far we have seen that circuits are quite powerful. In particular, P/poly contains undecidable problems, and $\mathsf{RP} \subseteq \mathsf{BPP} \subseteq \mathsf{P}/\mathsf{poly}$. In this lecture, we will explore this relationship further, proving results that show circuits are very unlikely to be super-powerful compared to uniform complexity classes.

Theorem 4.1. *A.* **(Shannon, 1949)** *"Most" Boolean functions $f : \{0,1\}^n \to \{0,1\}$, have circuit complexity $\mathsf{SIZE}(f) \geq \frac{2^n}{n} - \phi_n$, where ϕ_n is $o\left(\frac{2^n}{n}\right)$. (More precisely, for any $\epsilon > 0$ this holds for at least a $(1-\epsilon)$ fraction of all Boolean functions.)*

B. **(Lupanov, 1965)** *Every Boolean function $f : \{0,1\}^n \to \{0,1\}$ can be computed in $\mathsf{SIZE}(f) \leq \frac{2^n}{n} + \theta_n$, where θ_n is $o\left(\frac{2^n}{n}\right)$.*

Proof. A. The proof is a by a counting argument. Let $\mathbb{B}_n = \{f : \{0,1\}^n \to \{0,1\}\}$, that is, the set of all Boolean functions on n bits. $|\mathbb{B}_n| = 2^{2^n}$. We will show that the number of circuits of size much smaller than $2^n/n$ is only a negligible fraction of $|\mathbb{B}_n|$, proving the claim.

Let us compute the number of circuits of size at most $S \geq n+2$ over $\{\neg, \wedge, \vee\}$. Note that the argument we present works essentially unchanged for any complete basis of gates for Boolean circuits. What does it take to specify a given circuit? A gate labeled i in the circuit is defined by the labels of its two inputs, j and k ($j = k$ for unary gates), and the operation g the gate performs. The input labels j and k can be any of the S gates or the n inputs or the two constants, 0 and 1. The operation g can be any one of the three Boolean operations in the basis $\{\neg, \wedge, \vee\}$. Adding to this the name i of the gate, any circuit of size at most S can be specified by a description of length at most $(S+n+2)^{2S}3^S S$. Note, however, that such descriptions are the same up to the $S!$ ways of naming the gates. Hence, the total number of gates of size at most S, noting that $S! \geq (S/e)^S$, is at most

$$
\begin{aligned}
\frac{(S+n+2)^{2S}3^S S}{S!} &\leq \frac{(S+n+2)^{2S}(3e)^S S}{S^S} \\
&= \left(\frac{S+n+2}{S}\right)^S (3e\,(S+n+2))^S S \\
&= \left(1+\frac{n+2}{S}\right)^S (3e\,(S+n+2))^S S \\
&\leq \left(e^{\frac{n+2}{S}}\, 3e\,(S+n+2)\right)^S S \qquad \text{since } 1+x \leq e^x \\
&< (6e^2 S)^{S+1} \qquad \text{since we assumed } S \geq n+2.
\end{aligned}
$$

To be able to compute at least an ϵ fraction of all functions in $\mathbb{B}_n$, we need

$$\begin{array}{rrcl} & (6e^2S)^{S+1} & \geq & \epsilon\, 2^{2^n} \\ \Rightarrow & (S+1)\log_2(6e^2S) & \geq & 2^n - \log_2(1/\epsilon) \\ \Rightarrow & (S+1)(5.5+\log_2 S) & \geq & 2^n - \log_2(1/\epsilon) \end{array}$$

Hence, we must have $S \geq 2^n/n - \phi_n$ where ϕ_n is $o(2^n/n)$ to compute at least an ϵ fraction of all functions in $\mathbb{B}_n$ as long as ϵ is $2^{-o(2^n)}$. This proves part A of the Theorem.

B. Proof of this part is left as an exercise (see Problem 3, Assignment 1). Note that a Boolean function over n variables can be easily computed in $\mathsf{SIZE}(n2^n)$ by using its canonical DNF or CNF representation. Bringing it down close to $\mathsf{SIZE}(2^n/n)$ is a bit trickier.

$\square$

This gives a fairly tight bound on the size needed to compute most Boolean functions over n variables. As a corollary, we get a circuit size hierarchy theorem which is even stronger than the time and space hierarchies we saw earlier; circuits can compute many more functions even when their size is only roughly doubled.

Corollary 4.2 (Circuit-size Hierarchy). *For any $\epsilon > 0$ and $S_1, S_2 : \mathbb{N} \to \mathbb{N}$, if $n \leq (2+\epsilon)S_1(n) \leq S_2(n) \ll 2^n/n$, then $\mathsf{SIZE}(S_1(n)) \subsetneq \mathsf{SIZE}(S_2(n))$.*

Proof. Let $m = m(n)$ be the maximum integer such that $S_2(n) \geq (1+\epsilon/2)\, 2^m/m$. By the preconditions of the Corollary, $S_1(n) \leq (1-\epsilon/2)\, 2^m/m$ and $m \ll n$. Consider the set $\mathcal{F}$ of all Boolean functions on n variables that depend only on m bits of their inputs. By the previous Theorem, all functions in $\mathcal{F}$ can be computed by circuits of size $2^m/m + o(2^m/m)$ and are therefore in $\mathsf{SIZE}(S_2(n))$. On the other hand, most of the functions in $\mathcal{F}$ cannot be computed by circuits of size $2^m/m - o(2^m/m)$ and are therefore not in $\mathsf{SIZE}(S_1(n))$. $\square$

The following theorem, whose proof we will postpone until the next lecture, shows that circuits can quite efficiently simulate uniform computation. Its corollaries will be useful in several contexts.

Theorem 4.3 (Pippenger-Fischer, 1979). *If $T(n) \geq n$, then $\mathsf{TIME}(T(n)) \subseteq \bigcup_c \mathsf{SIZE}(cT(n)\log_2 T(n))$.*

We now show that although $\mathsf{P/poly}$ contains undecidable problems, it is unlikely to contain even all of NP. This implies that circuits, despite having the advantage of being non-uniform, may not be all that powerful. We start with a simple exercise:

Theorem 4.4 (Karp-Lipton). *If $\mathsf{NP} \subseteq \mathsf{P/poly}$, then $\mathsf{PH} = \Sigma_2\mathsf{P} \cap \Pi_2\mathsf{P}$.*

The original paper by Karp and Lipton credits Sipser with sharpening the result. The proof below which uses the same general ideas in a slightly different way is due to Wilson.

Proof. Suppose to the contrary that $\mathsf{NP} \subseteq \mathsf{P/poly}$. We'll show that this implies $\Sigma_2\mathsf{P} = \Pi_2\mathsf{P}$. From Lemma 2.6 this will prove the Theorem.

Let $L \in \Pi_2\mathsf{P}$. Therefore there exists a polynomial-time computable set R and a polynomial p such that $L = \{x \mid \forall^{p(|x|)}y\ \exists^{p(|x|)}z.\ (x,y,z) \in R\}$. The idea behind the proof is as follows. The inner relation in this definition, $\{(x,y) \mid \exists^{p(|x|)}z.\ (x,y,z) \in R\}$, is an NP language. $\mathsf{NP} \subseteq \mathsf{P/poly}$ implies that there exists a polynomial size circuit family $\{C_R\}$ computing this inner relation. We would like to simplify the definition of L using this circuit family. by

$$\left\{x \mid \exists \langle C_R \rangle\ \forall^{p(|x|)} y.\ C_R \text{ correctly computes } R \text{ on } (x, y) \text{ and } C_R(x, y) = 1\right\}.$$

This would put L in $\Sigma_2\mathsf{P}$, except that it is unclear how to efficiently verify that C_R actually computes the correct inner relation corresponding to R. (Moreover, the whole circuit family may not have a finite specification.)

To handle this issue, we modify the approach and use *self-reduction* for NP to verify correctness of the circuit involved. More precisely, we create a modified version of R suitable for self-reduction. Let

$$R' = \left\{(x, y, z') \mid |z'|, |y| \le p(|x|) \text{ and } \exists^{p(|x|)-|z'|} z''.\ (x, y, z', z'') \in R\right\}.$$

Here z' acts as a prefix of z in the earlier definition of R. Note that $R' \in \mathsf{NP}$ since R is polynomial-time computable. Therefore, by the assumption $\mathsf{NP} \subseteq \mathsf{P/poly}$, R' is computed by a polynomial size circuit family $\{C_n\}_{n=0}^{\infty}$ with a polynomial size bound $q : \mathbb{N} \to \mathbb{N}$. We, of course, can't encode the whole circuit family for showing $L \in \Sigma_2\mathsf{P}$. We use the fact that on input x, we only query R' on inputs (x, y, z) of length at most $2(|x| + 2p(|x|))$, say, assuming some reasonable encoding of the tuples.

Let $\mathcal{C}_{pref,|x|}$ be the smallest prefix of $\{C_n\}_n$ that contains circuits corresponding to all input sizes that are queried. The size of this is bounded some polynomial q' that involves the composition of p and q. We claim that there exists a polynomial-time algorithm M that given x, y and $\mathcal{C}_{pref,|x|}$ as input, either

a. outputs a z such that $(x, y, z) \in R$, in which case there exists a z satisfying this property, or

b. fails, in which case either $\mathcal{C}_{pref,|x|}$ is not a prefix of $\{C_n\}_{n=0}^{\infty}$ for computing the NP set R', or no such z exists.

We prove the claim by describing an algorithm M that behaves as desired. It will be clear that M runs in polynomial time.

```
Algorithm M: On input x, y, C_{pref,|x|},
    Let z' be the empty string
    If C_{pref,|x|}(x, y, z') = 0 then fail
    While (x, y, z') ∉ R and |z'| ≤ p(|x|)
        If C_{pref,|x|}(x, y, z'0) = 1
            then z' ← z'0
            else z' ← z'1
        EndIf
    EndWhile
    If (x, y, z') ∈ R
        then output z'
        else fail
    EndIf
End
```

Given M satisfying the conditions of our claim above, we can characterize the language L as follows: $x \in L$ iff $\exists^{q'(|x|)} \langle \mathcal{C}_{pref,|x|} \rangle\ \forall^{p(|x|)} y.\ M^{decision}(x, y, \langle \mathcal{C}_{pref,|x|} \rangle)$. Here $M^{decision}$ denotes the decision version of M that outputs true or false rather than z' or fail. Since M is polynomial-time computable, this shows that $L \in \Sigma_2\mathsf{P}$. Note that we were able to switch $\exists$ and $\forall$ quantifiers because $\mathcal{C}_{pref,|x|}$ doesn't depend on y.

This proves that $\Pi_2\mathsf{P} \subseteq \Sigma_2\mathsf{P}$. By the symmetry between $\Sigma_2\mathsf{P}$ and $\Pi_2\mathsf{P}$, this implies $\Sigma_2\mathsf{P} \subseteq \Pi_2\mathsf{P}$, making the two classes identical and finishing the proof. $\square$

The following exercise uses the same kind of self reduction that we employed in the above argument:

Exercise 4.1. Prove that $\mathsf{NP} \subseteq \mathsf{BPP}$ implies $\mathsf{NP} = \mathsf{RP}$.

We now prove that even very low levels of the polynomial time hierarchy cannot be computed by circuits of size n^k for any fixed k. This result, unlike our previous Theorem, is *unconditional*; it does not depend upon our belief that the polynomial hierarchy is unlikely to collapse.

Theorem 4.5 (Kannan). *For all k, $\Sigma_2\mathsf{P} \cap \Pi_2\mathsf{P} \not\subseteq \mathsf{SIZE}(n^k)$.*

Proof. We know that $\mathsf{SIZE}(n^k) \subsetneq \mathsf{SIZE}(n^{k+1})$ by the circuit hierarchy theorem. To prove this Theorem, we will give a problem in $\mathsf{SIZE}(n^{k+1})$ and $\Sigma_2\mathsf{P} \cap \Pi_2\mathsf{P}$ that is not in $\mathsf{SIZE}(n^k)$.

For each n, let C_n be the lexically first circuit on n inputs such that $size(C_n) \geq n^{k+1}$ and C_n is minimal; *i.e.*, C_n is not equivalent to a smaller circuit. (For lexical ordering on circuit encodings, we'll use $\prec$.) Let $\{C_n\}_{n=0}^{\infty}$ be the corresponding circuit family and let A be the language decided by this family. By our choice of C_n, $A \notin \mathsf{SIZE}(n^k)$. Also, by the circuit hierarchy theorem, $size(A)$ is a polynomial $\leq (2+\epsilon)n^{k+1}$ and the size of its encoding $|\langle A\rangle| \leq n^{k+3}$, say. Note that the factor of $(2+\epsilon)$ is present because there may not be a circuit of size exactly n^{k+1} that computes A, but there must be one of size at most roughly twice this much.

<u>Claim</u>: $A \in \Sigma_4\mathsf{P}$.
The proof of this claim involves characterizing the set S using a small number of quantifiers. By definition, $x \in A$ if and only if

$$\begin{aligned}
\exists^{p(|x|)}\langle C_{|x|}\rangle. \quad & \big(size(C_{|x|}) \geq |x|^{k+1} \\
& \bigwedge \forall^{p(|x|)}\langle D_{|x|}\rangle. \, [size(D_{|x|}) < size(C_{|x|}) \rightarrow \exists^{|x|}y. \, D_{|x|}(y) \neq C_{|x|}(y)] \\
& \bigwedge \forall^{p(|x|)}\langle D_{|x|}\rangle. \, [\, [(\langle D_{|x|}\rangle \prec \langle C_{|x|}\rangle) \wedge (size(D_{|x|}) \geq |x|^{k+1})] \rightarrow \\
& \qquad \exists^{p(|x|)}\langle E_{|x|}\rangle. \, [size(E_{|x|}) < size(D_{|x|}) \wedge \forall^{|x|}z. \, D_{|x|}(z) = E_{|x|}(z)] \,]\big)
\end{aligned}$$

The second condition states that the circuit is minimal, *i.e.*, no smaller circuit $D_{|x|}$ computes the same function as $C_{|x|}$. The third condition enforces the lexically-first requirement; *i.e.*, if there is a lexically-earlier circuit $D_{|x|}$ of size at least $|x|^{k+1}$, then $D_{|x|}$ itself is not minimal as evidenced by a smaller circuit $E_{|x|}$. When we convert this formula into prenex form, all quantifiers, being in positive form, do not flip. This gives us that $x \in A$ iff $\underbrace{\exists\langle C_{|x|}\rangle}\ \underbrace{\forall\langle D_{|x|}\rangle}\ \underbrace{\exists^{|x|}y\ \exists\langle E_{|x|}\rangle}\ \underbrace{\forall^{|x|}z}.\ \phi$ for a certain quantifier free polynomially decidable formula ϕ. Hence $A \in \Sigma_4\mathsf{P}$.

This proves the claim and imples that $\Sigma_4\mathsf{P} \not\subseteq \mathsf{SIZE}(n^k)$. We finish the proof of the Theorem by analyzing two possible scenarios:

a. $\mathsf{NP} \subseteq \mathsf{P/poly}$. In this case, by the Karp-Lipton Theorem, $A \in \Sigma_4\mathsf{P} \subseteq \mathsf{PH} = \Sigma_2\mathsf{P} \cap \Pi_2\mathsf{P}$ because the polynomial time hierarchy collapses, and we are done.

b. $NP \not\subseteq \mathsf{P/poly}$. In this simpler case, for some $B \in \mathsf{NP}$, $B \notin \mathsf{P/poly}$. This implies $B \notin \mathsf{SIZE}(n^k)$ and proves, in particular, that $\Sigma_2\mathsf{P} \cap \Pi_2\mathsf{P} \not\subseteq \mathsf{SIZE}(n^k)$.

This finishes the proof of the Thoerem. We note that unlike the existential argument (the witness is either the language A or the language B), one can also define a single language A' witnessing it where A' is a hybrid language between A and a diagonal language in NP. $\square$