CSE 531: Computational Complexity I **Winter 2016**

Lecture 15: Unique-SAT, Toda's Theorem, Circuit Lower Bounds

Feb 29, 2016

Lecturer: Paul Beame *Scribe: Paul Beame*

1 NP is as easy as Unique Solutions

Definition 1.1. *Let $USAT \subseteq SAT$ be the set of formulas φ such that φ has a unique satisfying assignment. Let* UP *be the set of languages A such that there is a polytime TM M and a polynomial p such that*

$$x \in A \Leftrightarrow \exists! y \in \{0,1\}^{p(|x|)}\ (M(x,y) = 1).$$

Lemma 1.2 (Valiant-Vazirani)**.** *There is a randomized polynomial-computable reduction f from SAT to $USAT$ with the following properties:*

$$[\varphi] \in SAT \rightarrow [f(\varphi)] \in USAT] \geq 1/(8n)$$
$$[\varphi] \notin SAT \rightarrow [f(\varphi)] \in SAT] = 0.$$

This follows immediately from

Lemma 1.3. *If $\emptyset \neq S \subseteq \mathbb{F}_2^n$, then for $v_1, \cdots v_{n+1} \in_R \mathbb{F}_2^n$,*

$$\mathbb{P}[\exists k \in \{2, \cdots n+1\}.\ |S \cap \{y \mid v_1 \cdot y = v_2 \cdot y = \ldots = v_k \cdot y = 0\}| = 1] > \frac{1}{8}.$$

This follows immediately from the following lemma.

Lemma 1.4. *Fix a set $S \subseteq \mathbb{F}_2^n$, then for $v_1, \ldots, v_{n+1} \in_R \mathbb{F}_2^n$,*

(a) if $0^n \in S$, then $\mathbb{P}[\ |S \cap \{y \mid v_1 \cdot y = v_2 \cdot y = \ldots = v_{n+1} \cdot y = 0\}| = 1\ ] > \frac{1}{2}$

(b) if $0^n \notin S$, and $2^{k-2} \leq |S| \leq 2^{k-1}$ then $\mathbb{P}[\ |S \cap \{y \mid v_1 \cdot y = v_2 \cdot y = \ldots = v_k \cdot y = 0\}| = 1\ ] > \frac{1}{8}$

Proof. We first show part (a). We always have $v_i \cdot 0^n = 0$ for all i. For any $x \in \mathbb{F}_2^n$, if $x \neq 0^n$ we have for any j that $\mathbb{P}[\ v_j \cdot x = 0\ ] = 1/2$. Therefore, since the v_j are chosen independently, for $x \in S \setminus \{0^n\}$,

$$\mathbb{P}[\ v_1 \cdot x = v_2 \cdot x = \ldots = v_{n+1} \cdot x = 0\ ] = \frac{1}{2^{n+1}}.$$

Thus

$$\mathbb{P}[\exists x \in S - \{0^n\},\ v_1 \cdot x = v_2 \cdot x = \ldots = v_{n+1} \cdot x = 0\ ] \leq \frac{|S| - 1}{2^{n+1}} < 1/2.$$

It follows that with probability greater than $1/2$, 0^n is the only element remaining in $S \cap \{y \mid v_1 \cdot y = v_2 \cdot y = \ldots = v_{n+1} \cdot y = 0\}$ as required.

We now prove part (b). Now assume that $0^n \notin S$ and $2^{k-2} \leq |S| \leq 2^{k-1}$. Define $h_k(x) = (v_1 \cdot x, \cdots, v_k \cdot x) \in \mathbb{F}_2^k$. As in the argument for part (a), for $x \neq 0^n$, $\mathbb{P}[h_k(x) = 0^k] = 1/2^k$. Suppose now that $x \neq y$ and $x, y \neq 0^n$, Then the condition that $h_k(x) = h_k(y) = 0^k$ is given by $2k$ equations whose coefficients are given by the coordinates of x and y. In particular, the constraints on the vector v_i given by the condition that both $v_j \cdot x = 0$ and $v_j \cdot y = 0$ are linearly independent and so both happen with probability $1/4$. There since the choices of the v_j are independent, $\mathbb{P}[h_k(x) = h_k(y) = 0^k] = 1/2^{2k}$ for $x, y \in S$, $x \neq y$. Let N be the number of elements of $x \in S$ such that $h_k(x) = 0^k$. By inclusion-exclusion,

$$\begin{aligned}
\mathbb{P}[N = 1] &\geq \sum_{x \in S} \mathbb{P}[h_k(x) = 0^k] - \sum_{x < y \in S} \mathbb{P}[h_k(x) = h_k(y) = 0^k] \\
&= \frac{|S|}{2^k} - \binom{|S|}{2} \frac{1}{2^{2k}} \\
&= \frac{|S|}{2^k} \left(1 - \frac{|S|}{2^{k+1}}\right) \\
&\geq \frac{1}{4}(1 - \frac{1}{4}) = 3/16 > 1/8
\end{aligned}$$

as required. □

Lemma 1.2 follows from Lemma 1.3 by replacing formula $\varphi(y)$ by $\varphi(y) \wedge (h_k(y) = 0^k)$ for a random choice of k between 2 and $n + 1$.

This is just a single random call but only succeeds with 1-sided error $1/(8n)$. By repeated this a linear number of times as with RP error reduction, we can obtain a reduction with success probability at least $2/3$.

Having a single satisfying assignment is a special case of having an odd number of assignments.

Definition 1.5. *For complexity class* C*, define* $\oplus$C *to be the set of languages A if and only if there is a relation $R \in$ C and a polynomial p such that $x \in A \Leftrightarrow$ there is an odd number of $y \in \{0,1\}^{p(|x|)}$ such that $R(x, y)$ is true. $\oplus SAT$ is the natural complete problem for* $\oplus$P.

Similarly define BP $\cdot$ C *to be the set of languages A such that there is a relation $R \in$ C and a polynomial p and constant $\varepsilon < 1/2$ such that the fraction of $y \in \{0,1\}^{p(|x|)}$ $R(x, y) = A(x)$ is at least $1 - \varepsilon$.*

Finally, define $\mathsf{R}\cdot\mathsf{C}$ *to be the set of languages* A *such that there is a relation* $R \in \mathsf{C}$ *and a polynomial* p *and* $\varepsilon < 1$ *such that if* $x \in A$ *then fraction of* $y \in \{0,1\}^{p(|x|)}$ *that satisfies* $R(x,y)$ *is at least* $1-\varepsilon$*, and if* $x \notin A$ *then no such* y *satisfies* $R(x,y)$*.*

As outlined in the text, with some extra work beyond the above construction, one can produce a randomized construction with a single call to $\oplus SAT$ that achieves probability at least $2/3$. This yields the following lemma:

Lemma 1.6. $\mathsf{NP} \subseteq \mathsf{R}\cdot\oplus\mathsf{P}$.

Toda shows how to extend this to multiple levels of the polynomial-time hierarchy. Because of the alternation, this can generate errors on both sides.

Lemma 1.7 (Toda)**.** $\mathsf{PH} \subseteq \mathsf{BP}\cdot\oplus\mathsf{P}$.

As a consequence of this, he showed the following theorem.

Theorem 1.8 (Toda)**.** $\mathsf{PH} \subseteq \mathsf{P}^{\#\mathsf{P}} = \mathsf{P}^{\mathsf{PERM}}$.

Therefore, though approximate counting can be done in a low level of the polynomial-time hierarchy, exact counting is enough to simulate the entire hierarchy.

We do not have time to give the details of the proof which are found in the text. It is convenient to think of the proof in terms viewing PH as uniform circuits of unbounded fan-in 2^{n^c} and constant-depth.

2 Low level circuit classes

Definition 2.1. *Define the circuit complexity class* $\mathsf{NC}^{\mathsf{k}} = \mathsf{SIZE\text{-}DEPTH}(\mathsf{n}^{\mathsf{O}(1)}, \mathsf{O}(\log^{\mathsf{k}}\mathsf{n}))$ *to be the set of fan-in 2 Boolean functions computable in polynomial size and* $O(\log^k n)$ *depth. Let* $\mathsf{NC} = \bigcup_{\mathsf{k}} \mathsf{NC}^{\mathsf{k}}$.

NC^{k} is the circuit analog of the sequential complexity class $\mathsf{SC}^{\mathsf{k}} = \mathsf{DTIME\text{-}SPACE}(\mathsf{n}^{\mathsf{O}(1)}, \log^{\mathsf{k}}\mathsf{n})$ and $\mathsf{SC} = \bigcup_{\mathsf{k}} \mathsf{SC}^{\mathsf{k}}$.

NC stands for "Nick's class" after Nick Pippenger, in contrast to SC which stands for "Steve's class" after Steve Cook. Cook had originally named the class SC, PLOPS. Both were at the University of Toronto and the awkwardness of pronouncing "plops" caused Alan Borodin to introduce the names informally and the names stuck.

On the midterm you showed that $\mathsf{NL} \subseteq \mathsf{NC}^2$. However, though we know that $\mathsf{NL} \in \mathsf{P} \cap \mathsf{DSPACE}(\log^2 \mathsf{n})$, it is open whether or not $\mathsf{NL} \in \mathsf{SC}$.

The following shows that NC^1 is special.

Theorem 2.2. $\mathsf{NC}^1 = \mathsf{DEPTH}(\mathsf{O}(\log \mathsf{n})) = \mathsf{FORMULA\text{-}SIZE}(\mathsf{n}^{\mathsf{O}(1)})$.

Proof. For the first equality observe that in any fan-in 2 circuit of depth $O(\log n)$ has at most $2^{O(\log n)} = n^{O(1)}$ paths to its inputs and hence has formula size at most $n^{O(1)}$. Therefore the circuit-size limitation in the definition of NC^1 is unnecessary.

To complete the proof it is necessary to show that any formula of size at most $n^{O(1)}$ can be converted so that it has depth at most $O(\log n)$. In general, a formula of polynomial size can have linear depth. The idea is to recursively re-balance the formula.

First we find a node v in the formula that has between $1/3$ and $2/3$ of all the leaves of the formula as descendants. (Start at the output node, follow the child having more leaf descendants until it has at most $2/3$ of the total leaves are descendants of v. Since its parent has more than $2/3$ and it was the larger child, it has more than $1/3$ of total.

Let T be the top of the formula and B be the part leading to v. We recursively re-balance T and B to get T' and B'. To build the final formula, we let T'_b for $b \in \{0, 1\}$ be T' with input v replaced by value b. The new balanced formula is $(T'_0 \wedge \neg B') \vee (T'_1 \wedge B')$. □

Not only do we not know how to prove that some explicit Boolean function is not in NC^1, it is an open question to prove that some explicit Boolean function is not $\mathsf{SIZE\text{-}DEPTH}(\mathsf{n}, \mathsf{O}(\log \mathsf{n}))$.

Definition 2.3. AC^k *is the set of Boolean functions computable by circuits of size* $n^{O(1)}$ *and depth* $O(\log^k n)$ *circuits with* unbounded fan-in $\vee$ *and* $\wedge$ *gates (as well as* $\neg$ *gates).*

The "A" in AC stands for "alternating". Since it is easy to simulate an unbounded fan-in gate with a polynomial number of inputs by a circuit of depth $O(\log n)$ we have:

Proposition 2.4. $\mathsf{NC}^\mathsf{k} \subseteq \mathsf{AC}^\mathsf{k} \subseteq \mathsf{NC}^{\mathsf{k}+1}$.

Note that AC^0 generalizes CNF and DNF formulas. There are interesting (multi-output) functions in AC^0. For example, using carry-lookahead adders we can see that $INTEGER\text{-}ADDITION \in \mathsf{AC}^0$.

We can save a little bit in depth by using unbounded fan-in circuits.

Lemma 2.5. NC^1 *has polynomial-size unbounded fan-in circuits of depth* $O(\log n / \log \log n)$.

Proof. Break up the NC^1 circuit of $O(\log n)$ depth into $O(\log n/\log\log n)$ layers of height $\log\log n$. Each fan-in 2 subcircuit of height $\log\log n$ can depend on at most $2^{\log\log n} = \log n$ gates (or input variables) in the next layer. By construction the worst CNF or DNF on $m = \log n$-bit strings is at most size $m2^m = n\log n$. Therefore we can remove any gates in the intermediate levels, with only a polynomial blow-up in size. □

As we have discussed earlier, the best circuit lower bounds known for any explicit Boolean function over $\wedge, \vee, \neg$ circuits is $5n - o(n)$. For general 2-input gates, the best circuit-size lower bound known was recently improved from $3n - o(n)$ to $3.18n$.

Even for Boolean formulas, the largest size lower bound using $\wedge, \vee, \neg$ gates is $n^{3-o(1)}$, and for arbitrary fan-in 2 gates is $\Omega(n^2/\log n)$.

It is a bit embarassing that these bounds are so small.

We now outline some of the lower bounds known for these low level circuit classes, which are much larger. The following theorem implies that $\mathsf{NC}^1 \not\subseteq \mathsf{AC}^0$:

Theorem 2.6 (Furst-Saxe-Sipser, Ajtai)**.** $PARITY \notin \mathsf{AC}^0$

Corollary 2.7. *$INTEGER\text{-}MULTIPLICATION$ is not in AC^0.*

In fact, Hästad showed that the above simulation of NC^1 by AC^0 is tight by proving

Theorem 2.8. *Any unbounded fan-in circuit computing $PARITY$ on n bits in depth d requires size at least $2^{\Omega(n^{1/(d-1)})}$. In particular, polynomial size requires depth $d = \Omega(\log n/\log\log n)$.*

Since $\exists$ quantifiers correspond to exponentially large unbounded fan-in $\vee$ gates, and $\forall$ quantifiers correspond to exponentially large unbounded fan-in $\wedge$ gates, there is a natural translation from problems in PH to problems on circuits. Indeed one of the motivations of Furst-Saxe-Sipser was to define an oracle separating PSPACE from PH.

One can also consider adding unbounded fan-in $\oplus$ gates to obtain the complexity class $\mathsf{AC}^0[2]$. More generally, one can add MOD_m gates that have Boolean values and take Boolean inputs and give value 0 iff the sum if the inputs is $0 \bmod m$ for any m and define a corresponding circuit complexity class $\mathsf{AC}^0[m]$.

Theorem 2.9 (Razborov,Smolensky)**.** *Let p and q be distinct primes. Then $MAJORITY \notin \mathsf{AC}^0[p]$ and $MOD_q \notin \mathsf{AC}^0[p]$.*

Let $\mathsf{ACC}^0 = \bigcup_m \mathsf{AC}^0[m]$. Until recently, finding any non-trivial lower bounds even for $\mathsf{AC}[6]$ was an open question. In 2011, Ryan Williams showed that

Theorem 2.10. $\mathsf{NEXP} \not\subseteq \mathsf{ACC}^0$.

(The weaker precise statement for the case of $\mathsf{AC}[6]$ was listed as an open problem in the text.)

The methods for these three lower bounds are very different.

The AC^0 lower bounds for $PARITY$ are proven by showing that randomly assigning values to some of the inputs, known as random restrictions simplify the circuit a lot but do not simplify the $PARITY$ function.

The Razborov-Smolensky lower bounds for $\mathsf{AC}^0[\mathsf{p}]$ follow by approximating each gate of the circuit by a low degree multivariate polynomial modulo p.

The lower bound for NEXP for ACC^0 follows by a complex diagonalixation and finding a satisfiability algorithm for n-input ACC^0 circuits that runs in $2^n/n^{\Omega(1)}$ time.

In addition to finding ACC^0 lower bounds for Boolean functions in NP some problems at the edge of circuit complexity

Definition 2.11. TC^k *is the set of Boolean functions computable by circuits of size* $n^{O(1)}$ *and depth* $O(\log^k n)$ *circuits with* unbounded fan-in $MAJORITY$ *gates (or* threshold *gates). (A threshold gate is given by a set of integer coefficients* $a_1, \ldots, a_n$ *and a threshold* b *and outputs 1 iff* $\sum_i a_i x_i \geq b$*.*

It is open to prove that some explicit function is in not computable in

- depth 2 circuits of threshold gates of polynomial size.
- depth 3 circuits of $MAJORITY$ gates of polynomial size (which can be simulated by depth 2 threshold circuits efficiently).

The kinds of techniques that have been shown to work for AC^0 and $\mathsf{AC}^0[\mathsf{p}]$ likely cannot be used to prove lower bounds for TC^0 because TC^0 is thought to contain cryptographic objects called pseudorandom function generators which are indistinguishable from truly random functions. The previous methods for proving the lower bounds, which are called "natural proofs" are capable of distinguishing all functions from the respective classes from truly random functions and so pseudorandom generation inside the class would be impossible. The recent approach beginning with that for ACC^0 may not be so limited.