CS 2429 - Approaches to the P versus NP Question

Lecture #3: 29 January 2014

Lecturer: Toniann Pitassi

Scribe Notes by: Brent Mombourquette

1 Applications of AC^0 lower bounds

In this lecture we will be examining some applications of AC^0 lower bounds proofs.

1.1 Pseudo-random Generators

A big question that pseudo-random generators (PRGs) may be able to answer is whether $BPP =^? P$ or $RP =^? P$, that is can probabilistic algorithms be derandomized and made to run deterministically in polynomial time.

First recall from a previous lecture the following theorem.

Theorem 1 (Hastad) *For sufficiently large n, any family $\{C_n\}$ of depth d circuits of size $s \leq 2^{n^{1/(d+1)}}$ has:*

$$|Pr[C_n(x) = Parity(x)] - 1/2| \leq 2^{n^{1/(d+1)}}$$

Additionally, we have the following theorem for PRGs for AC^0 circuits, using the above results.

Theorem 2 (NW94) *$\forall d$ there exists a family of functions $\{g_n : \{0,1\}^\ell \to \{0,1\}^n\}$ where $\ell = O(log(n)^{2d+6})$ such that:*

(1) $\{g_n\}$ is computed by log-space uniform circuits of polynomial size depth $d+4$
(2) $\forall \{C_n\}$ of polynomial size depth d and $\forall$ poly $p(n)$ for sufficiently large n:

$$|Pr[C_n(y) = 1] - Pr[C_n(g_n(y)) = 1]| \leq \frac{1}{p(n)}$$

assuming y is uniform from $\{0,1\}^n$.

The generating function g_n "fools" the circuits. Here it is defined as:

$$g_n(x) = Parity(x|_{s_1})Parity(x|_{s_2}) \ldots Parity(x|_{s_n})$$

Where the seed $s_1 \ldots s_n \subset \{0,1\}^\ell$ is such that $|s_i| = (logn)^{d+3}$ and $|s_i \cap s_j| \leq log(n)$ $\forall i \neq j$. Essentially, the seed is divided into a number of almost disjoint subsets and and applied as a restriction to the input of the hard functions.

Therefore, any probabilistic AC^0 circuit C of depth d can be simulated with a deterministic circuit of depth roughly $2d$.

1.2 Algorithms for $AC^0 - SAT$ and $AC^0 - \#SAT$

The $AC^0 - SAT$ problem is the satisfiability problem defined as follows.

Definition The $AC^0 - SAT$ problem is given some circuit $C_n \in AC_d^0$ of size s, accept C_n if $\exists a \in \{0,1\}^n$ such that $C_n(a) = 1$.

The $AC^0 - \#SAT$ is defined similarly except it outputs the number of such satisfying assignments a. The trivial brute force approach for both problems is to try all possible assignments, taking time $poly(|C_n|) \cdot 2^n$.

The general approach is to express the worst case runtime of algorithm solving these problems in the for $|C_n| \cdot 2^{n(1-\mu}$ where μ is the savings over the brute force method.

The following theorem was proved concerning the existence of an algorithm for solving these $AC^0 - SAT$ problems in better than brute force worst case runtime.

Theorem 3 (IMP12) *There exists a Las Vegas algorithm (zero-error randomized algorithm) that takes as input a depth d circuit C_n with cn gates and produces a set of restrictions $\{\rho_i\}_i$ partitioning $\{0,1\}^n$ such that $\forall i$ $C_n|_{\rho_i}$ is 0 or 1. The expected runtime and number of restrictions is*

$$poly(n) \cdot |C_n| \cdot 2^{n(1-\mu_{c,d})}$$

where $\mu_{c,d} = \frac{1}{O(log(c)+dlog(d))^{d-1}}$.

The high level proof idea for this theorem begins with the a slightly modified version of Hastad's Switching Lemma, which tells us that, with high probability, a restriction ρ on a circuit C_n produces a small height decision tree. A restriction that extends ρ partitions the circuit space. The restrictions that do not extend ρ are then partitioned such that they partition the Boolean cube $\{0,1\}^n$ into not too many disjoint regions such that the original circuit is constant over each region.

The following corollary comes directly from the previous theorem.

Corollary 4 ($AC^0 - SAT$ and $AC^0 - \#SAT$ Algorithm) *There exists a Las Vegas algorithm for $AC^0 - SAT$ and $AC^0 - \#SAT$ for depth d circuits with cn gates with expected savings* $\mu_{c,d} = \frac{1}{O(log(c)+dlog(d))^{d-1}}$.

The theorem also produces the following bounds on correlation between AC^0 circuits and the Parity function, improving Hastad's lower bound.

Corollary 5 (AC^0 correlation with Parity) *Any depth d size cn AC^0 circuit has correlation with Parity at most $2^{n(1-\mu_{c,d})}$.*

1.3 Nontrivial Compression Algorithm for the Circuit Class C

The nontrivial compression algorithm problem is defined as follows.

Definition The *compression algorithm problem for C* is given the truth table of a boolean function $f_n \in C$, so the length of the input is 2^n, output a circuit computing f_n of size $\leq 2^n/n$ (the trivial achievable for any n-variate Boolean function) such that the runtime of the algorithm is polynomial in the input size, $2^{O(n)}$.

Such a compression algorithm exists for small sized AC^0 circuits as a result of the following theorem.

Theorem 6 (CKK+13) *Size s depth d AC^0 circuits are compressible in time $2^{O(n)}$ to circuits of size $\leq 2^{n(1-\frac{1}{O(logs)^{d-1}})}$.*

Proof Using the results of [IMP12], every depth d circuit with s gates and n inputs has an equivalent DNF representation with at most $poly(n) \cdot s \cdot 2^{n(1-\mu)}$ where $\mu \geq \frac{1}{O(log(c)+dlog(d))^{d-1}}$. No suppose some minimal DNF representation of a function $f : \{0,1\}^n \to \{0,1\}$, given by its truth table, has ℓ terms. We can compute a DNF representation of f that is at most $O(n)$ factor larger than that of the minimal DNF for f through a greedy Set Cover approach.
First, compute all of the minimum terms of f, the truth table, by brute force. That is, try all possible terms and check any assignment to it evaluates to 1 on f and removing any one variable makes some input not evaluate to 1. Let this set of possible minimum terms be $\{t_1, t_2, \ldots\}$. Note that there are at most 2^{2n} such terms (one can use an n bits to describe the characteristic functions of a subset of n variables, and another n bits to describe the signs of the chosen variables) so this can be done in time $2^{O(n)}$.
Let S_i be the set of assignments that extend t_i and let U be the set of all strings $\alpha \in \{0,1\}$ such that $f(\alpha) = 1$. Note that each $S_i \subset U$. The following greedy Set Cover algorithm is run.
Find a subset S_i that covers at least $\frac{1}{\ell}$ fraction of the points in U that have not been covered before. By an averaging argument, some such S_i must exist. Repeat until all of U is covered.
Since ℓ subsets cover U, they also cover every subset of U. Therefore, in each iteration, there exists a subset that covers at least $\frac{1}{\ell}$ fraction of points that were uncovered in the previous iteration. After each iteration, the size of the set of points that are not covered reduces by the factor $(1 - \frac{1}{\ell})$. After t iterations, the number of points uncovered is at most $|U| \cdot (1 - \frac{1}{\ell})^t \leq |U| \cdot e^{-\frac{t}{\ell}}$. Setting $t = O(\ell log|U|)$ makes this value less than 1 and since $|U| = 2^n$ t is size $O(\ell n)$.
The whole algorithm is $poly(2^n)$ and returns a DNF representation of f with $poly(n) \cdot s \cdot 2^{n(1-\mu)}$ terms.

Note that the above algorithm gives nontrivial compression for depth d AC^0 circuits of size at most $2^{n^{\frac{1}{d-1}}}$, the size of which we know lower bounds for AC^0 circuits for explicit functions.

These types of nontrival compression algorithms can be used to determine circuit lower bounds through their relation to *natural properties*. [IKW02] shows natural properties against **P/poly** imply **NEXP** $\subsetneq$ **P/poly**, which extends to compression algorithms as they are natural properties. This is summarized in the following theorem.

Theorem 7 *Let $\mathcal{C} \subseteq$ **P/poly**. Suppose for all natural numbers c there exists a deterministic polynomial time algorithm that compresses $f \in \mathcal{C}[n^c]$ to a circuit of size less than $2^n/n$. Then **NEXP** $\subsetneq \mathcal{C}$.*

1.4 Compression Games - Computing Bounded Communication Complexity

Given a circuit class $\mathcal{C}$ and a language $L \subset \{0,1\}^*$ the $\mathcal{C}$-compression game for L between two players, Alice and Bob, is as follows. Alice has some input bit string x and a sequence of circuits

$\{\mathcal{C}_n\} \in \mathcal{C}$ while Bob has a strategy, call it f. Alice first applies $\mathcal{C}_{|x|}$ to x getting the result y_1 which is sent then sent to Bob. Depending on how many rounds of communication are defined in the message passing protocol Q, Bob may send message back to Alice. After receiving y_1 Bob calculates $f(y_1) = z_1$ and sends z_1 to Alice. In turn, Alice applies a fixed circuit $\mathcal{C}_{|x|}$ to $\langle x, y_1, z_1 \rangle$ computing y_2, continuing the processes until the last round in which the final bit sent is the answer to whether $x \in L$. The cost of the compression game is sum of the lengths of all messages sent by Alice - the cost does not include the aggregate length of messages sent by Bob.

For compression games, we have the following result.

Lemma 8 (CS12) *Let $c(n) \leq n$ and $\mathcal{C}$ be a class of circuits closed under logical OR and negation (i.e. $\mathcal{C} = AC^0$) of size $s(n)$. If there is a $\mathcal{C}(s(n))$ compression game for language L of cost $\leq c(n)$ then L has correlation at least $\frac{1}{O(2^{c(n)})}$ with $\mathcal{C}(s(n))$.*

Proof The idea of the proof of this lemma involves first reformulating the existence of a $\mathcal{C}(s(n))$ compression game for language L into the existence of a transcript Π that is accepting, Alice-consistent, and Bob-consistent.
A transcript $T = \langle y_1, z_1, y_2 \ldots y_r \rangle$ is a sequence of messages in the protocol - it may not be a valid sequence of messages though. A transcript is Bob-consistent if $\forall i, 1 \leq i \leq r-1, z_i = f(y_1 \ldots y_r)$. Therefore, it is Bob-consistent if the sequence of messages agree with Bob's strategy f. It is important to note that a transcript being Bob-consistent depends only on the transcript itself and not on x. Similarly, a transcript is Alice-consistent on x if $\forall i, 1 \leq i \leq r, y_i = \mathcal{C}_{|x|}(x, y_1, z_1 \ldots z_{r-1})$. A transcript is accepting if the final message y_r is 1, meaning $x \in L$.
Now assuming $x \in L$ then clearly the accepting transcript following the given protocol for the circuits $\{\mathcal{C}_n\} \in \mathcal{C}$ used by Alice and the strategy f used by Bob is both Alice-consistent and Bob-consistent by definition. In the other direction, assuming the protocol being used is correct for the $\mathcal{C}(s(n))$ compression game for L and that the given transcript T is consistent on x and accepting. We can easily see by induction on the elements of T that it must be both Alice-consistent and Bob-consistent and in the end the final message reflects the acceptance of x, implying $x \in L$.
Returning to the lemma at hand, notice that there are at most $2^{c(n)}$ Bob-consistent accepting transcripts bounded by size $c(n)$. The idea is then to check each Bob-consistent accepting transcript for whether it is also Alice-consistent. This can be done using a large OR over small circuits that compute the Alice-consistency over all Bob-consistent accepting transcripts. The Alice checking is done efficiently and in parallel by a circuit $\mathcal{C}'_\Pi$ that consists of a top level AND gate fan-in r where r is half of the size of the transcript Π (checking the consistency of all y_i messages with $x, y_1 \ldots z_{i-1}$ using $O(|y_i|)$ OR and negation gates). The size of $\mathcal{C}'_\Pi$ is bounded by $O(s(n))$ and since $\mathcal{C}$ is closed under OR and negation, $\mathcal{C}'_\Pi \in \mathcal{C}$.
By the Discriminator Lemma, if L is computed by the OR of at most $f(n)$ circuits from $\mathcal{C}$ then L has correlation at least $\frac{1}{O(f(n))}$ with $\mathcal{C}$. Replacing $f(n)$ with $2^{c(n)}$ produces the lemma.

This connection between compression games and correlation produces the following lower bound for AC^0-compression for the Parity language.

Theorem 9 (IMP12) *Parity has correlation at most $2^{-n/O((log(s))^{d-1})}$ with for size s depth d AC^0-circuits.*

2 References

[CKK+13] Ruiwen Chen, Valentine Kabanets, Antonina Kolokolova, Ronen Shaltiel, and David Zuckerman. Mining circuit lower bound proofs for meta-algorithms. *Electronic Colloquium on Computational Complexity (ECCC)*, 20:57, 2013.

[CS12] Arkadev Chattopadhyay and Rahul Santhanam. Lower bounds on interactive compressibility by constant-depth circuits. In *FOCS*, pages 619628. IEEE Computer Society, 2012.

[IMP12] R. Impagliazzo, W. Matthews, and R. Paturi. A satis
ability algorithm for AC0. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 961-972, 2012.

[IKW02] R. Impagliazzo, V. Kabanets, and A. Wigderson. In search of an easy witness: Exponential time vs. porbabilistic polynomial time. *Journal of Computer and System Sciences*, 65(4):672-694, 2002.

[NW94] N. Nisan and A. Wigderson. Hardness vs. randomness. *Journal of Computer and System Sciences*, 49:149-167, 1994.