6.841: Advanced Complexity Theory Fall 2012

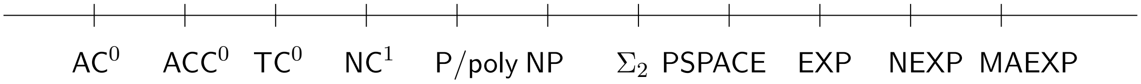
Lecture 7 — September 27, 2012

Prof. Dana Moshkovitz *Scribe: Arturs Backurs*

1 Overview

We want to prove that $\mathsf{NP} \not\subseteq \mathsf{P/poly}$. This problem seems to hard to solve with our present proof techniques. At least we can replace class NP with some stronger class C_1 and class $\mathsf{P/poly}$ with some weaker class C_2 and to prove that $C_1 \not\subseteq C_2$. In this lecture we investigate for which classes we can prove this statement and what are the most promising current approches.

2 The Zoo



Some of the definitions below are from the textbook or [4].

Definition 1. AC^0 *is the class of languages computable by circuit families of constant depth, polynomial size, and whose gates are from the set* $\{\mathsf{NOT}, \mathsf{OR}, \mathsf{AND}\}$ *with unbounded fan-in.*

Definition 2. *(*ACC^0*) For any integer* m*, the* MOD_m *gate outputs* 0 *is the sum of its inputs is* 0 *modulo* m*, and* 1 *otherwise.*

For integers $m_1, m_2, ..., m_k > 1$ *we say a language* L *is in* $\mathsf{ACC}^0(m_1, m_2, ..., m_k)$ *if there exists a circuit family* $\{C_n\}$ *with constant depth and polynomial size (and unbounded fan-in) consisting of* AND*,* OR*,* NOT*, and* MOD_{m_1}*,* MOD_{m_2}*, ...,* MOD_{m_k} *gates accepting* L*.*

The class ACC^0 *contains every language that is in* $\mathsf{ACC}^0(m_1, m_2, ..., m_k)$ *for some* $k \geq 0$ *and* $m_1, m_2, ..., m_k > 1$*.*

Definition 3. *(*MA*) Language* L *is in* MA *is there exists probabilistic polynomial-time Turing machine* V *and polynomial* p *such that for every input string* x *of length* $n = |x|$*,*

- *if* $x \in L$*, then* $\exists y \in \{0,1\}^{p(n)} \mathsf{Pr}(V(x,y) = 1) \geq 2/3$*,*
- *if* $x \notin L$*, then* $\forall y \in \{0,1\}^{p(n)} \mathsf{Pr}(V(x,y) = 1) \leq 1/3$*.*

M stands for Merlin and A stands for Arthur. Arthur is a verifier with a random number generating device and Merlin is a prover with infinite computational pover. Class MA is a probabilistic version of NP.

We obtain class $\mathsf{MA}_{\mathsf{EXP}}$ if we replace polynomial time with exponential time in the definition of MA.

We have containment $\mathsf{NEXP} \subseteq \mathsf{MA}_{\mathsf{EXP}}$ because Arthur can ignore random number generating device. We also have containment $\mathsf{AC}^0 \subseteq \mathsf{ACC}^0$ because we can ignore MOD gates.

Definition 4. TC^0 *is a class of decision problems solvable by polynomial-size, constant-depth circuits with unbounded fanin, which can use* AND*,* OR*, and* NOT *gates as well as threshold gates. A threshold gate returns* 1 *if at least half of its inputs are* 1*, and* 0 *otherwise.*

Definition 5. NC^1 *is a class of decision problems solvable by a family of Boolean circuits, with polynomial size,* $O(\log(n))$ *depth, and bounded fan-in over* AND*,* OR*, and* NOT *gates.*

NC stands for Nick's Class. (Named in honor of Nick Pippenger.)

Known facts:

- We showed $\mathsf{ACC}^0 \not\subseteq \mathsf{AC}^0$. We proved that $\mathsf{PARITY} \notin \mathsf{AC}^0$. Clearly, $\mathsf{PARITY} \in \mathsf{ACC}^0$ (use MOD_2 gate). Thus AC^0 is very weak.
- The best seperation we know for ACC^0 is $\mathsf{NEXP} \not\subseteq \mathsf{ACC}^0$ [1].
- $\mathsf{MA}_{\mathsf{EXP}} \not\subseteq \mathsf{P/poly}$.

3 Super-polynomial lower bound for $\mathsf{MA}_{\mathsf{EXP}}$

Theorem 6. $\mathsf{MA}_{\mathsf{EXP}} \not\subseteq \mathsf{P/poly}$.

Proof. Suppose that $\mathsf{MA}_{\mathsf{EXP}} \subseteq \mathsf{P/poly}$. It follows that $\mathsf{PSPACE} \subseteq \mathsf{P/poly}$. We know that $\mathsf{IP} = \mathsf{PSPACE}$ [2], where IP stands for Interactive Polynomial time. IP is generalization of MA for polynomially many rounds. In this case $\mathsf{PSPACE} = \mathsf{MA}$ (the prover can send in one round the circuit for computing the prover strategy (strategy can be computed in PSPACE) in the interactive proof). By simple padding this implies that $\mathsf{MA}_{\mathsf{EXP}} = \mathsf{EXPSPACE}$. But $\mathsf{EXPSPACE} \not\subseteq \mathsf{P/poly}$ because $\Sigma_3\mathsf{EXP} \not\subseteq \mathsf{P/poly}$ (by similar argument as for $\Sigma_3^{\mathsf{P}} \not\subseteq \mathsf{SIZE}(n^k)$). Thus $\mathsf{MA}_{\mathsf{EXP}} \not\subseteq \mathsf{P/poly}$. □

We believe that $\mathsf{NEXP} = \mathsf{MAEXP}$.

4 Lower bound for NC^1? (Valiant's approach)

4.1 Depth-reduction

Lemma 7 ([3])**.** *Any circuit width* m *wires and depth* d *has a set* S *of* km/l *wires (*$l = \lceil \log d \rceil$*) whose removal leaves depth* $\leq d/2^k$*.*

Proof. First we topologically sort the gates in d layers such that the input wires at the bottom, and each wire (u, v) "goes up", where u and v are gates in the layers. Let $depth(x)$ denote the index of layer in which gate x is located. We label each wire (u, v) by the index of the most significant bit in which binary representations of $depth(u)$ and $depth(v)$ differs. Because there are m wires and l labels there must be a label that repeats in $\leq m/l$ wires. We remove those wires. Now the depth is $\leq d/2$ (there is no more need for corresponding index in the binary representation of layer). Repeat the same procedure k times. □

4.2 Application of Lemma 7

Consider a linear-sized, logarithmic depth, fan-in-2 circuit over $\{\mathsf{NOT}, \mathsf{OR}$ and $\mathsf{AND}\}$ with n outputs. Note that these circuits doesn't exactly capture NC^1 because they are linear in size instead of polynomial and they have n outputs, but it is a good model of NC^1.

We apply lemma in the following way. Suppose that $d = c\log n$. Choose $k = \log(c/\epsilon)$ and remove $\frac{mk}{\log\lceil c\log n\rceil} = O(n/\log\log n)$ wires, where the last equality holds because m is the size of the circuit and the circuit is linear in size. We obtain circuit of size $\leq \epsilon\log n$. Since we have fan-in-2 gates, we must have that each output is connected to at most $2^{\epsilon\log n} = n^\epsilon$ inputs. Thus each output is completely determined by n^ϵ inputs and the values of the removed wires (the number of the removed wires is $O(n/\log\log n)$).

Now the idea to prove some lower bound is as follows: choose your favorite strong class and show that some function in this class doesn't have this property. Suprisingly, no one has been able to prove some interesting lower bound using this approach. It is possible to show that random function does not have this property with high probability.

4.3 A linear-algebraic analog

Consider the previous circuit except with gates of the following kind: $\mathsf{ADD}_{\alpha,\beta}(g_1, g_2) = \alpha g_1 + \beta g_2$, where α, β, g_1 and g_2 are elements from some field. The circuit computes some linear function f over some field, i.e., there exists a matrix A such that $f(\overrightarrow{x}) = A\overrightarrow{x}$. The previous property allows us to rewrite any matrix A computed by such a circuit as a sum of two matrices B and C such that B has low rank ($\leq n/\log\log n$) and C is n^ϵ-sparse. We can say that A is "close to low rank". If matrix can not be written as the sum of a matrix of a low rank and a sparse matrix we call it a rigid matrix. It is possible to prove that almost all matrices are rigid.

To prove a lower bound in some complexity class using this method we have to find a linear function f in this class with a corresponding rigid matrix. This does not seem to face any known barriers – it just seems to require some cleverness.

References

[1] R. Williams, *Non-Uniform ACC Circuit Lower Bounds* In 26th IEEE Conference on Computational Complexity (CCC 2011)

[2] A. Shamir. IP = PSPACE. J. ACM, 39(4):869-877, 1992. Prelim version FOCS'90.

[3] L. G. Valiant. Graph-theoretic properties in computational complexity. J. Comput. Syst. Sci., 13(3):278-285, 1976. Prelim version STOC'75.

[4] http://qwiki.stanford.edu/index.php/Complexity_Zoo