Natural Proofs

Miguel Romero

June 19, 2012

Why are circuit lower bounds so difficult?

Natural proofs:

Why are circuit lower bounds so difficult?

Natural proofs:

Demonstrate that *known* methods are inherent too weak to prove strong circuit lower bounds.

Why are circuit lower bounds so difficult?

Natural proofs:

Why are circuit lower bounds so difficult?

Natural proofs:

- All known circuit lower bound proofs are natural.

Why are circuit lower bounds so difficult?

Natural proofs:

- All known circuit lower bound proofs are natural.
- No natural proofs can prove polynomial lower bounds for general circuits

Why are circuit lower bounds so difficult?

Natural proofs:

- All known circuit lower bound proofs are natural.
- No natural proofs can prove polynomial lower bounds for general circuits (modulo widely believed cryptographic assumptions).

Why are circuit lower bounds so difficult?

Natural proofs:

- All known circuit lower bound proofs are natural.
- No natural proofs can prove polynomial lower bounds for general circuits (modulo widely believed cryptographic assumptions).

We use computational complexity to shed light on a metamathematical question about computational complexity.

Related work

Limits of diagonalization and relativizing methods.

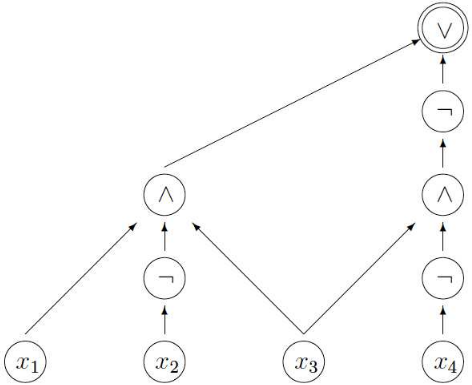
Theorem (Baker, Gill, Solovay 75')

There exists oracles A, B such that $\mathrm{P}^A = \mathrm{NP}^A$ and $\mathrm{P}^B \neq \mathrm{NP}^B$.

Outline

Outline

Basic definitions



$$f(x_1, x_2, x_3, x_4) = (x_1 \wedge \neg x_2 \wedge x_3) \vee \neg(x_3 \wedge \neg x_4)$$

Basic definitions

Circuit parameters:

- Number of inputs
- Size (number of vertices)
- Depth (longest directed path from an input node to the output node).

Basic definitions

Circuit parameters:

- Number of inputs
- Size (number of vertices)
- Depth (longest directed path from an input node to the output node).

Different models of circuits:

- Bounded-unbounded fan-in.
- Monotone circuits.
- Additional gates (MOD_m gates, majority gates,...).

Circuit-based complexity classes

Let $\{C_n\}$ be a circuit family where C_n has n inputs.

Definition (language recognition)

Let $L \subseteq \{0,1\}^*$ be a language. We say that $\{C_n\}$ *decides* L if for every $x \in \{0,1\}^n$

$$x \in L \iff C_n(x) = 1$$

Restriction on circuit families

Let $\{C_n\}$ be a circuit family and $T : \mathbb{N} \to \mathbb{N}$ be a function:

Definition (family size and depth)

We say that $\{C_n\}$ has size (depth) $T(n)$ if for every n, the size (depth) of C_n is at most $T(n)$.

Restriction on circuit families

Let $\{C_n\}$ be a circuit family and $T : \mathbb{N} \to \mathbb{N}$ be a function:

Definition (family size and depth)

We say that $\{C_n\}$ has size (depth) $T(n)$ if for every n, the size (depth) of C_n is at most $T(n)$.

Another important restriction:

Definition (uniformity)

We say that $\{C_n\}$ is *uniform* if there exists a polynomial time TM that on input 1^n outputs (a representation of) C_n.

Some complexity classes

AC^0: Languages decided by circuit families of $O(poly(n))$ size, $O(1)$ depth and *unbounded* fan-in.

Some complexity classes

AC^0: Languages decided by circuit families of $O(poly(n))$ size, $O(1)$ depth and *unbounded* fan-in.

$\mathsf{ACC}^0[m]$: Languages decided by circuit families of $O(poly(n))$ size, $O(1)$ depth, *unbounded* fan-in and MOD_m gates.

Some complexity classes

AC^0: Languages decided by circuit families of $O(poly(n))$ size, $O(1)$ depth and *unbounded* fan-in.

$\mathsf{ACC}^0[m]$: Languages decided by circuit families of $O(poly(n))$ size, $O(1)$ depth, *unbounded* fan-in and MOD_m gates.

$\mathsf{ACC}^0 = \bigcup_{m>1} \mathsf{ACC}^0[m]$

Some complexity classes

AC^0: Languages decided by circuit families of $O(poly(n))$ size, $O(1)$ depth and *unbounded* fan-in.

$ACC^0[m]$: Languages decided by circuit families of $O(poly(n))$ size, $O(1)$ depth, *unbounded* fan-in and MOD_m gates.

$ACC^0 = \bigcup_{m>1} ACC^0[m]$

$$AC^0 \subsetneq ACC^0$$

Some complexity classes

TC^0: Languages decided by circuit families of $O(poly(n))$ size, $O(1)$ depth, *unbounded* fan-in and MAJORITY gates.

Some complexity classes

TC^0: Languages decided by circuit families of $O(poly(n))$ size, $O(1)$ depth, *unbounded* fan-in and MAJORITY gates.

NC^1: Languages decided by circuit families of $O(poly(n))$ size, $O(logn)$ depth and *bounded* fan-in.

Some complexity classes

TC^0: Languages decided by circuit families of $O(poly(n))$ size, $O(1)$ depth, *unbounded* fan-in and MAJORITY gates.

NC^1: Languages decided by circuit families of $O(poly(n))$ size, $O(logn)$ depth and *bounded* fan-in.

P/poly: Languages decided by circuit families of $O(poly(n))$ size.

Some complexity classes

TC^0: Languages decided by circuit families of $O(poly(n))$ size, $O(1)$ depth, *unbounded* fan-in and MAJORITY gates.

NC^1: Languages decided by circuit families of $O(poly(n))$ size, $O(logn)$ depth and *bounded* fan-in.

P/poly: Languages decided by circuit families of $O(poly(n))$ size.

$$\mathrm{AC}^0 \subsetneq \mathrm{ACC}^0 \subseteq \mathrm{TC}^0 \subseteq \mathrm{NC}^1 \subseteq \mathrm{P/poly}$$

$\mathsf{ACC}^0 \subseteq \mathsf{TC}^0$

We want to simulate MOD_m gates using majority gates. W.l.o.g. assume n is even (otherwise, add an extra input with the constant 0).

$$\mathsf{MOD}_m(x_1, ..., x_n) = \bigvee\{\#k(x_1, ..., x_n) : 0 \le k \le n, m \text{ divides } k\}.$$

where $\#k$ outputs 1 iff there is exactly k 1's in the inputs.

$$\#k(x_1, ..., x_n) = Th_k(x_1, ...x_n) \wedge \neg Th_{k+1}(x_1, ..., x_n).$$

where $Th_k(x_1, ..., x_n)$ outputs 1 iff there is at least k 1's in the inputs.

$\mathsf{ACC}^0 \subseteq \mathsf{TC}^0$

Finally, we can simulate Th_k gates using majority gates.

If $k \leq n/2$, then $Th_k(x_1, ...x_n) = MAJORITY(x_1, ..., x_n, 1, ..., 1)$
where the additional number of 1's in the input is $n - 2k$.
If $k > n/2$, then $Th_k(x_1, ...x_n) = MAJORITY(x_1, ..., x_n, 0, ..., 0)$
where the additional number of 0's in the input is $2k - n$.

Still polynomial size and constant depth.

Outline

Proving P $\neq$ NP

Researchers moved from relativizing methods to circuit lower bounds.

Theorem

P $\subseteq$ P/poly

Proving P $\neq$ NP

Researchers moved from relativizing methods to circuit lower bounds.

Theorem

P $\subseteq$ P/poly

So, it suffices to prove:

$$\mathsf{NP} \not\subseteq \mathsf{P/poly}$$

Proving $P \neq NP$

There was some reasons to think that $NP \not\subseteq P/poly$.

Proving P $\neq$ NP

There was some reasons to think that NP $\not\subseteq$ P/poly.

Theorem (Karp-Lipton 80')

If NP $\subseteq$ P/poly, then PH$=\Sigma_2^p$.

Proving P $\neq$ NP

There was some reasons to think that NP $\not\subseteq$ P/poly.

Theorem (Karp-Lipton 80')

If NP $\subseteq$ P/poly, then PH$=\Sigma_2^p$.

Theorem (Shannon 49')

For every $n > 1$, there exists a function $f : \{0,1\}^n \to \{0,1\}$ that cannot be computed by a circuit C of size $2^n/(10n)$.

Proving P $\neq$ NP

Main Idea:

- Prove lower bounds for restricted classes of circuit.
- Extend techniques to general circuits.

Some circuit lower bounds

Theorem (Furst,Saxe,Sipser 81' - Ajtai 83')

PARITY $\notin$ AC^0

Some circuit lower bounds

Theorem (Furst,Saxe,Sipser 81' - Ajtai 83')

PARITY $\notin$ AC^0

Theorem (Razborov 85')

There is no polynomial sized family of *monotone* circuits for CLIQUE.

Some circuit lower bounds

Theorem (Furst,Saxe,Sipser 81' - Ajtai 83')

PARITY $\notin$ AC^0

Theorem (Razborov 85')

There is no polynomial sized family of *monotone* circuits for CLIQUE.

Theorem (Razborov 87'- Smolensky 87')

For distinct primes p and q, the language MOD_p is not in $\mathsf{ACC}^0[q]$.

Some circuit lower bounds

Theorem (Furst,Saxe,Sipser 81' - Ajtai 83')

PARITY $\notin$ AC^0

Theorem (Razborov 85')

There is no polynomial sized family of *monotone* circuits for CLIQUE.

Theorem (Razborov 87'- Smolensky 87')

For distinct primes p and q, the language MOD_p is not in $\mathsf{ACC}^0[q]$.

The project ground to a halt at the class ACC^0...

Outline

Natural property

$\mathcal{F}_n$ denotes the set of all boolean functions in n variables. A property P of boolean functions is a family $\{P_n \subseteq \mathcal{F}_n\}_{n \geq 1}$.

Natural property

$\mathcal{F}_n$ denotes the set of all boolean functions in n variables. A property P of boolean functions is a family $\{P_n \subseteq \mathcal{F}_n\}_{n\geq 1}$.

Definition (Natural property)

A property $P = \{P_n\}$ is *natural* if satisfies the following two conditions:

- Constructiveness: There is an $2^{O(n)}$ time algorithm that on input (the truth table of) a function $g : \{0,1\}^n \to \{0,1\}$ outputs $P(g)$.
- Largeness: $Pr_{g\in_R \mathcal{F}_n}[P(g) = 1] \geq 1/n$, for sufficiently large n.

Natural property

Let $\mathcal{C}$ be a complexity class.

Definition (usefulness)

A property $P = \{P_n\}$ is *useful against $\mathcal{C}$* if for any family of function $\{f_n\}$ (with $f_n \in \mathcal{F}_n$) where $P_n(f_n) = 1$ happens infinitely often, then $\{f_n\} \notin \mathcal{C}$.

Natural property

Let $\mathcal{C}$ be a complexity class.

Definition (usefulness)

A property $P = \{P_n\}$ is *useful against $\mathcal{C}$* if for any family of function $\{f_n\}$ (with $f_n \in \mathcal{F}_n$) where $P_n(f_n) = 1$ happens infinitely often, then $\{f_n\} \notin \mathcal{C}$.

Any language in $\mathcal{C}$ "is not" in the property P.

Natural proof

What is a natural proof?

"Definition"

A natural proof is any proof that use (explicitly or implicitly) a natural property.

"There exists (no) natural proofs ..." = "There exists (no) natural property P ..."

Examples

1. $P_n(g) = 1$ iff g has circuit complexity more than $n^{\log n}$.
2. $P_n(g) = 1$ iff g correctly solves 3SAT on inputs of size n.

PARITY $\notin$ AC^0 is natural

The proof defines the following natural property useful against AC^0 (where $k(n)$ is an appropriate function):

$P_n(g) = 1$ iff there is no restriction of the variables with $k(n)$ unassigned variables which forces g to be a constant function.

PARITY $\notin$ AC^0 is natural

The proof defines the following natural property useful against AC^0 (where $k(n)$ is an appropriate function):

$P_n(g) = 1$ iff there is no restriction of the variables with $k(n)$ unassigned variables which forces g to be a constant function.

Almost all circuit lower bounds are natural.

Outline

Negative results

Theorem (Razborov, Rudich 96')

Suppose that subexponentially strong one-way functions exists. Then there is no natural property useful against P/poly.

Proof of the Theorem

The existence of subexponentially strong one-way functions implies the existence of subexponentially strong *pseudorandom function families*

That is, a family $\{f_s\}_{s \in \{0,1\}^*}$, where for $s \in \{0,1\}^m$, f_s is a function from $\{0,1\}^m$ to $\{0,1\}$, and satisfies the following conditions:

1. There is a polynomial-time algorithm that given s, x outputs $f_s(x)$.
2. $f_s(\cdot)$ for $s \in_R \{0,1\}^m$ cannot be distinguished from a random function in $\mathcal{F}_m$ by 2^{m^ε}- algorithms, for some fixed constant ε.

Proof of the Theorem

Suppose there exists natural property P useful against P/poly. We construct the following algorithm $\mathcal{A}$ with oracle access to a function h:

1. On input 1^m define $n = \lceil m^{\varepsilon/2} \rceil$.
2. Construct the truth table of $g(x) = h(x0^{m-n})$.
3. Output $P(g)$.

Proof of the Theorem

Suppose there exists natural property P useful against P/poly. We construct the following algorithm $\mathcal{A}$ with oracle access to a function h:

1. On input 1^m define $n = \lceil m^{\varepsilon/2} \rceil$.
2. Construct the truth table of $g(x) = h(x0^{m-n})$.
3. Output $P(g)$.

$\mathcal{A}$ runs in $2^{m^{\varepsilon}}$-time and breaks $\{f_s\}_{s \in \{0,1\}^*}$

Proof of the Theorem

Running time:

1. On input 1^m define $n = \lceil m^{\varepsilon/2} \rceil$.
2. Construct the truth table of $g(x) = h(x0^{m-n})$ $2^{O(n)}$.
3. Output $P(g)$ $2^{O(n)}$ (constructiveness condition) .

Proof of the Theorem

Running time:

1. On input 1^m define $n = \lceil m^{\varepsilon/2} \rceil$.
2. Construct the truth table of $g(x) = h(x0^{m-n})$ $2^{O(n)}$.
3. Output $P(g)$ $2^{O(n)}$ (constructiveness condition) .

$\mathcal{A}$ runs in time $2^{O(n)} = 2^{O(m^{\varepsilon/2})} = O(2^{m^{\varepsilon}})$.

Proof of the Theorem

$\mathcal{A}$ breaks $\{f_s\}_{s\in\{0,1\}^*}$ means:

$\mathsf{Adv}_{\mathcal{A}}(m) = |Pr_{r\in_R\mathcal{F}_m}[\mathcal{A}^r(1^m)=1] - Pr_{s\in_R\{0,1\}^m}[\mathcal{A}^{f_s(\cdot)}(1^m)=1]|$
is non-negligible.

A function $\epsilon : \mathbb{N} \to [0,1]$ is *negligible* if for all $c > 0$, $\epsilon(m) < 1/m^c$ for sufficiently large m.

Proof of the Theorem

Suppose $\mathcal{A}$ has access to a random function $r \in \mathcal{F}_m$, then g is random in $\mathcal{F}_n$.
Using the largeness condition:

$Pr_{r \in_R \mathcal{F}_m}[\mathcal{A}^r(1^m) = 1] = Pr_{g \in_R \mathcal{F}_n}[P_n(g) = 1] \geq 1/m^{\varepsilon/2}$ for sufficiently large m.

Proof of the Theorem

Suppose $\mathcal{A}$ has access to $f_s(\cdot)$, where s is random in $\{0,1\}^m$.

Since $f_s(x)$ is computable in polynomial time in s and x, it follows that can be computed by a polynomial sized circuit family. Using that P is useful against P/poly we have that for sufficiently large m it holds that

$\forall s \in \{0,1\}^m \quad f_s(\cdot, 0^{m-n})$ is not in P_n

Therefore, for those m's

$$Pr_{s \in_R \{0,1\}^m}[\mathcal{A}^{f_s(\cdot)}(1^m) = 1] = 0$$

We conclude that

$\mathrm{Adv}_{\mathcal{A}}(m) \geq 1/m^{\varepsilon/2}$ for sufficiently large m. Then $\mathrm{Adv}_{\mathcal{A}}(m)$ is non-negligible.

Outline

Limits of natural proofs

$$\mathsf{AC}^0 \subsetneq \mathsf{ACC}^0 \subseteq \mathsf{TC}^0 \subseteq \mathsf{NC}^1 \subseteq \mathsf{P/poly}$$

Even classes like TC^0 and NC^1 contain plausible pseudorandom functions. Thus natural proofs useful against TC^0 or NC^1 are unlikely to exist.

Limits of natural proofs

$$\mathsf{AC}^0 \subsetneq \mathsf{ACC}^0 \subseteq \mathsf{TC}^0 \subseteq \mathsf{NC}^1 \subseteq \mathsf{P/poly}$$

Even classes like TC^0 and NC^1 contain plausible pseudorandom functions. Thus natural proofs useful against TC^0 or NC^1 are unlikely to exist.

There is little evidence that ACC^0 contains pseudorandom functions.

Thus natural proofs useful against ACC^0 can exists.

Limits of natural proofs

$$\mathsf{AC}^0 \subsetneq \mathsf{ACC}^0 \subseteq \mathsf{TC}^0 \subseteq \mathsf{NC}^1 \subseteq \mathsf{P/poly}$$

Even classes like TC^0 and NC^1 contain plausible pseudorandom functions. Thus natural proofs useful against TC^0 or NC^1 are unlikely to exist.

There is little evidence that ACC^0 contains pseudorandom functions.

Thus natural proofs useful against ACC^0 can exists.

but, there wasn't any strong ACC^0 lower bound.

Limits of natural proofs

Theorem (R. Williams 11')

NEXP $\not\subseteq$ ACC^0

Use non natural arguments (diagonalization). Although it's not clear that natural proof should considered a barrier for ACC^0.

Another related works

- Algebrizing techniques (A. Wigderson 08').
- Alternating-Trading proofs for Time-Space lower bounds (S. Buss, R. Williams 11').

Natural Proofs

Miguel Romero

June 19, 2012