17/9/2015 Russell's course, lec 2

Lower bounds: reasons why functions are hard to compute.

Upper bounds: alg design. So far, techniques for LBs = techniques for algs.

Next few lectures: LBs → algs → LBs, whatever technique is used to derand, boosting.

Derandomization. Primality testing: given binary string x, is it a prime?

PIT: given some compact desc of a poly, as a product of monomials, is poly $\equiv 0$.

Eg 1, $(x+y+z+w)^4 + (x+y-z-w)^4 - (x-z+y-w)^4 - (x-y-z+w)^4 \equiv 0$?

Algebraic circuits over integers.

DLSZ: plug in random values. If more values than deg, then can derandomize.

Issue: numbers get too big - (e.g.: $x^n + y^n = z^n$, large n, x, y, z now to check?)

Take mod random primes. Non-zero mod random prime is non-zero. coRP algs.

PIT $\in$ coRP., BPP: two-sided error, const.

Rabin, Solovay, Strassen: randomized polynomial testing. in TC^0.

2001: Agrawal, Kayal, Saxena: Primality $\in P$.

PIT: not known to be in P.

Generic probabilistic alg: $A(x, \text{pick } r \text{ at random}) \xrightarrow{\text{acc}} \{0,1\}$

Let test $T_x(r) \equiv A(x,r)$

$|r|$ = time for A on $|x|$. For each x, $T_x(r)$ is a circuit with $\oplus$ gates.

Want to find: prob $T_x(r)=1$ to within small additive error. (amount to solve

CAPP: given $C(r_1 \dots r_m)$ estimate $\Pr_r[C(r)=1]$ to $\pm 1/4$ (see BPP correct?, go

Randomized alg: pick random, estimate fraction which is accepted.

Generator $G: \{0,1\}^l \to \{0,1\}^m$, usual: $l \ll m$ and G is not too horrible to compute

Brute force G-derandomization: for each $s \in S_1 \dots S_l$, define $r_s = G(s)$, time

Compute $C(r_s)$. Estimate fraction on which C accepts. Time: $2^l(\text{Time}_G(l) + \text{circuit}$

Goal: crypto PRG: $\text{Time}_G(l) = \text{poly}(l)$. here $\{0,1\}^l$: $\text{Time}_G(l) \approx 2^{O(l)}$

Goal: minimize l.

A circ. C fools G-RDP: G vs. randomness distinguishing problem, if

$\Pr_s[C(G(s))=1] - \Pr_r[C(r)=1] \geq \varepsilon$

Lower bound for G-RDP: for every small C, C fails to solve G-RDP.

Upper bound (correctness for our algs): $\forall$ small C, $C(G(s))$ is good approx to $C($

(here, small = fixed poly size), i.e., C & G fail to solve G-RDP.

Constructing a PRG: Let $F(y_1 \dots y_n)$ be (S, ε)-hard if $\forall$ circ C of size S, $\Pr_y[F(y)=1]$

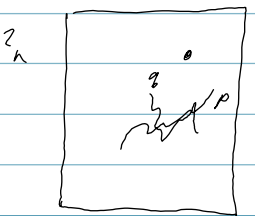
Special case: $\varepsilon = 0$ & next most easy case. ε = worst, almost everywhere: $\varepsilon = 1/2 - \delta$, where $\delta = 1/n$

NW: construction of a PRG $G: \{0,1\}^{n^2} \to \{0,1\}^m$ $S = \text{poly}(m)$, $\varepsilon = \frac{1}{2} - \frac{1}{S}$.

For AC^0, $S = 2^{n^{1/a}}$ parity is hard.

Let $G_{i+1}: \{0,1\}^n \to \{0,1\}^{n+1}$. $G_{i+1}(x) = x, F(x)$

C : $\mathbb{Z} \to x_1 \ldots x_m$, where each x_i, $|x_i| = r$. Then output $F(x_1) \ldots F(x_m)$.
Design (in new terms). Let $\ell = O(n^2)$. $Z_1 \ldots Z_\ell$: restriction of Z to some bit positions. $A_i \subseteq \{0,1,\ldots,\ell\}$, $x_i = Z|_{A_i}$
would be good if $A_i \cap A_j = \emptyset$, but next best $|A_i \cap A_j| \le k$ for small k.
If so, $\le 2^k$ advantage in computing x_i. Could provide $F(x_i)$ given $\{F(x_j) \mid j \ne i\}$.
Z : $Z|_{A_i}$ | $Z|_{\mathbb{Z}_\ell - A_i}$. Find w_i, prediction of $F(x_i)$ given $F(x_j)$ and
x_i w_i,
fixed w_i string just as ... So every bit of x_j not in $A_i \cap A_j$ fixed to a bit in w_i. So $x_j|_{w_i}$ is k-bit input. 2^k possible values.
for small k, can do a truth table. For each $F(x_j)$, of size $< 2^k$.
Get circ of size $S_2 = S_1 + O(2^k \cdot m)$



Suppose n a prime (round up), $\ell = n \times n$
n^2 : because of birthday paradox.
polynomials of low deg have few points in common,
look at univariate polynomials of deg k mod n.
$A_p = \{ i, p(i) \mid i = 0 \ldots n-1 \}$ ... $a_0 + a_1 x + \ldots + a_k x^k$.
$m = n^k$. $k = \log n / \log m < \log m$,
$|A_p \cap A_q| \le k$. $= |\{i \mid p(i) = q(i)\}|$ So $O(2^k \cdot m) = O(m^2)$, $S_2 = S_1 \times O(m^2)$
As S_1 is $O(m)$, need F to be $O(m^2)$-hard
Application: if C is size m AC^0 circ, depth d, it can estimate parity on
with prob $\le \exp(-n / \log^d m)$. Pick $n = \log^{d+1} m$. Let P be parity on
$\log^{d+1} m$ bit inputs, 6 looks, size m depth d circs.
$CAPP(AC^0_d)$ in time $2^{\log^{2d+2} m}$: guessing in circ size.