DD2445 COMPLEXITY THEORY: LECTURE 11

LAST TIME

- Wrapped up circuit complexity (for now)
- Started talking about RANDOMIZED COMPUTATION

Probabilistic Turing machine (PTM)

- Two transition functions in every state
- Flips coin which one to take
- Output $M(x)$ random variable

For language L, let $L(x) = \begin{cases} 0 & \text{if } x \notin L \\ 1 & \text{if } x \in L \end{cases}$

$\boxed{BPP}$ Bounded-error probabilistic polynomial time

Class of languages for which $\exists$ PTM M that <u>always</u> runs in polynomial time (regardless of coin flips) and $\forall x$

$$\Pr\left[M(x) = L(x) \right] \geq \frac{2}{3}$$

Probability <u>not</u> over input x

<u>only over</u> internal randomness of algorithm

$$P \subseteq BPP \subseteq EXP$$

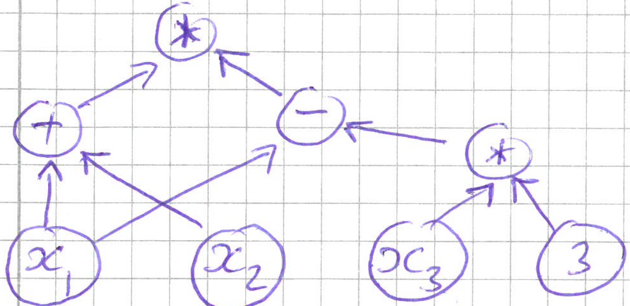
Believe $P = BPP$

But there are things we currently don't know how to do efficiently without randomness

POLYNOMIAL IDENTITY TESTING

<u>Input</u>: Algebraic circuit representing multivariate polynomial with integer coefficients

<u>Task</u>: Decide whether polynomial identically zero



$$(x_1 + x_2)(x_1 - 3x_3)$$
$$= x_1^2 - 3x_1x_3 + x_1x_2 - 3x_2x_3$$

$$\text{ZEROP} = \left\{ \begin{array}{l} \text{algebraic circuits corresponding to} \\ \text{identically zero polynomials} \end{array} \right\}$$

<u>Idea for randomized algorithm</u>

- Pick random input $(\alpha_1, \alpha_2, \dots, \alpha_n)$
- Set $x_i = \alpha_i$ for $i = 1, \dots, n$ and evaluate circuit
- If result $\neq 0$, declare circuit $\notin$ ZEROP
- If result $= 0$, say probably $\in$ ZEROP

<u>Always correct</u> on no-instances

But what about yes-instances?

SCHWARTZ-ZIPPEL LEMMA

Let $p(x_1, \ldots, x_m)$ non-zero polynomial of total degree $\leq d$. Let S finite set of integers. Then for $a_1, \ldots, a_m$ chosen from S uniformly randomly with replacements it holds that

$$\Pr\left[p(a_1, \ldots, a_m) \neq 0\right] \geq 1 - \frac{d}{|S|}$$

Proof By induction over m

Base case $m = 1$: Univariate polynomial of degree $\leq d \Rightarrow$ at most d distinct roots

Induction step: Write

$$p(x_1, x_2, \ldots, x_m) = \sum_{i=0}^{d} x_1^i \, p_i(x_2, \ldots, x_m)$$

p is non-zero $\Rightarrow$ some p_i non-zero. Pick largest i s.t. p_i non-zero. By induction hypothesis

$$\Pr_{a_2, \ldots, a_m}\left[p_i(a_2, \ldots, a_m) \neq 0\right] \geq 1 - \frac{d-i}{|S|}$$

When $p_i(a_2, \ldots, a_m) \neq 0$ we get that

$$p(x_1, a_2, \ldots, a_m) = \sum_{j=0}^{i} x_1^j \, p_j(a_2, \ldots, a_m)$$

is a univariate polynomial of degree $\leq i$, and so $= 0$ for at most i values, so

$$\Pr\left[p(a_1, a_2, \ldots, a_m) \neq 0\right] \geq \left(1 - \frac{i}{|S|}\right)\left(1 - \frac{d-i}{|S|}\right) \geq 1 - \frac{d}{|S|}$$

$\boxed{\equiv}$

R V

Refined testing idea

Circuit of size m $\Rightarrow$ $\leq m$ multiplications
$\Rightarrow$ degree $\leq 2^m$

Let $S = \{1, 2, \ldots, 10 \cdot 2^m\}$

Pick $a_i \in S$ randomly & evaluate circuit.

By Schwartz-Zippel, if circuit encodes non-zero polynomial, then 90% chance of seeing non-zero output

If circuit encodes zero polynomial, then output always zero

Done? Not quite...

Problem: If degree $\approx 2^m$, then numbers grow as large as $(10 \cdot 2^m)^{2^m}$ $\Rightarrow$ exponentially many bits

Hard to achieve in polynomial time...

Solution: "FINGERPRINTING"

Compute modulo some $k \in [2^{2m}]$

Computing modulo k

After each operation, divide by k and take remainder

Suppose $y = C(a_1, \ldots, a_m)$

If $y = 0$, then $y = 0 \pmod{k}$

CLAIM 5

If $y \neq 0$, then randomly chosen $k \in [2^{2m}]$ will not divide y with prob $\geq \delta = \frac{1}{4m}$

Given this claim, run test $O(m)$ times and accept only if always get 0 output
$\Rightarrow$ arbitrarily high constant probability of success

Proof of Claim 5

Assume $y \neq 0$. $y \leq (10 \cdot 2^m)^{2^m}$

Let B = prime factors of y.

Sufficient to show that with prob $\geq \delta$ k is a prime not in B

y has at most $\log y \leq 5m\,2^m$ prime factors

By Prime Number Theorem — constant is actually 1 (*)

$$\#\text{primes} \leq N \sim N / \ln N$$

$$\#\text{primes} \leq 2^{2m} \sim \frac{2^{2m}}{2m} > \frac{2^{2m}}{4m} \quad \text{for large enough } m$$

$$5m\,2^m = o\left(\frac{2^{2m}}{2m}\right) < \frac{2^{2m}}{8m} \quad \text{for large enough } m$$

$$\Pr[k \text{ prime not in } B] \geq \frac{2^{2m}/8m}{2^{2m}} = \frac{1}{8m}$$

*) See Thm A.23 in Arora-Barak for sufficient, simpler version

Many natural randomized algorithms have <u>one-sided error</u>

Might make mistake when $x \in L$ but never when $x \notin L$ or the other way round

(we just saw one such example)

<u>DEF 6</u> $RTIME(T(n))$ contains every language L for which $\exists$ PTM M running in time $O(T(n))$ such that

$$x \in L \Rightarrow \Pr[M(x) = 1] \geq 2/3$$
$$x \notin L \Rightarrow \Pr[M(x) = 0] = 1$$

$$RP = \bigcup_{c \in \mathbb{N}^+} RTIME(n^c)$$

<u>OBS 7</u> $RP \subseteq NP$

<u>Pf</u> Every accepting branch is a certificate.

Don't know if $BPP \subseteq NP$

RP: "Never false positives" (<u>positive</u> answers always <u>right</u>)

$coRP \doteq \{L \mid \bar{L} \in RP\}$ "Never false negatives"

Given general PTM M, can define random variable

$T_{M,x}$ = running time of M on x.

Take <u>expectation</u> of this random variable

$$E[T_{M,x}] = \sum_{t=1}^{\infty} t \cdot \Pr[T_{M,x} = t]$$

Say M has expected running time $T(n)$ if

$$\forall x \in \{0,1\}^* \quad E[T_{M,x}] \leq T(|x|)$$

DEF 8 ZTIME$(T(n))$ contains all languages L for which $\exists$ PTM M that runs in expected time $O(T(n))$ such that

$$\Pr[\, M(x) = L(x) \mid M \text{ halts} \,] = 1$$

$$ZPP = \bigcup_{c \in \mathbb{N}^+} ZTIME(n^c)$$

"Zero-sided error"

ZPP zero-error probabilistic polynomial time

THM 9 $ZPP = RP \cap coRP$

Proof Exercise.

Also immediately clear from def

$$RP \subseteq BPP$$
$$coRP \subseteq BPP$$

ROBUSTNESS OF DEFINITIONS

(a) Error probability: constant 2/3 arbitrary
(b) Can use expected running time instead of worst case
(c) can use biased coins
(d) Can even use imperfect random sources ("weak random sources")

Will show (a) — see Sec 7.4 for the rest

LEMMA 10 For $c > 0$ constant, let $BPP_{1/2+n^{-c}}$ denote class of languages L for which $\exists$ poly-time PTM M s.t. $\forall x \in \{0,1\}^*$

$$\Pr[M(x) = L(x)] \geq \frac{1}{2} + |x|^{-c}$$

Then $BPP_{1/2+n^{-c}} = BPP$.

Need to show: can go from success prob $\frac{1}{2} + n^{-c}$ to $2/3$.

Show sth stronger: can go to $1 - 2^{-n^d}$

exponentially small failure prob.

THM 11 (ERROR REDUCTION FOR BPP)

Suppose $\exists$ poly-time PTM M for L s.t.

$$\forall x \quad \Pr[M(x) = L(x)] \geq 1/2 + |x|^{-c}.$$

Then $\forall d > 0$ $\exists$ poly-time PTM M' s.t.

$$\forall x \quad \Pr[M(x) = L(x)] \geq 1 - 2^{-|x|^d}$$

Proof

M' runs M for $k = 8|x|^{2c+d}$ times, collects answers, and takes majority vote.

How confident can we be that this is correct? Use material from App A.2.1 & A.2.4

Let $X_i = \begin{cases} 1 & \text{if } i\text{th run of } M \text{ gets } x \text{ right} \\ 0 & \text{otherwise} \end{cases}$

$\Pr[X_i = 1] = p$ for $p \geq \frac{1}{2} + |x|^{-c}$ [suppose $p = 1/2 + |x|^{-c}$ for simplicity]

$$E\left[\sum_{i=1}^{k} X_i\right] = kp = \underbrace{\frac{8|x|^{2c+d}}{2}}_{\text{half}} + \underbrace{8|x|^{c+d}}_{\text{margin}}$$

"If you repeat independent trials sufficiently many times, you will get very close to expected value with very high probability"

R IX

LEMMA 12 (CHERNOFF BOUND)

deviation from expected value

$$\Pr\left[\left|\sum_{i=1}^{k} X_i - pk\right| > \delta pk\right] < \exp\left(-\frac{\delta^2}{4} pk\right)$$

Plug in $p = \frac{1}{2} + |x|^{-c}$

$\delta = |x|^{-c}/2$

We will be correct unless $\sum_{i=1}^{k} X_i < pk - \delta pk$.

That probability is bounded by

$$\exp\left(-\frac{1}{4|x|^{2c}} \cdot \frac{8|x|^{2c+d}}{2}\right) = \exp\left(-|x|^d\right) < 2^{-|x|^d} \quad \boxed{}$$

Relationship between BPP and other classes?

<u>THM 12</u> BPP $\subseteq$ P/poly

<u>THM 13</u> BPP $\subseteq \Sigma_2^p \cap \Pi_2^p$ ($\subseteq$ PH)

Both proofs use error reduction in Thm 11 plus some other ideas.

Proof of Thm 13 is <u>extremely</u> neat... But will have to skip it due to time constraints.

Try to sketch proof of Thm 12

L R X

Proof of Thm 12

If $L \in BPP$, then by Thm 11 (and Prop 3)
$\exists$ PTM M that on input size n

- uses m random bits
- gets answer right except with prob $2^{-(n+1)}$

Let r be the random bits
Say r bad for x if $M(x,r) \neq L(x)$

For every x, M succeeds with prob $\geq 1 - 2^{-(n+1)}$
$\Rightarrow$ out of 2^m random strings, $\leq 2^m / 2^{n+1}$ bad for x.

$$\left| \{ r \mid r \text{ bad for some } x \} \right| \leq \sum_{x \in \{0,1\}^n} \left| \{ r \mid r \text{ bad for this } x \} \right|$$

$$\leq \cancel{\tfrac{1}{2}}\, 2^n \cdot \frac{2^m}{2^{n+1}} = 2^m / 2$$

But this means that there is at least one random string $r^* \in \{0,1\}^m$ (in fact, at least half) that are good for all $x \in \{0,1\}^n$

Run M with this r^* as advice!

Checking Thm 11 again, r^* will have poly size.

What about complete problems for BPP?

Typical complete problems
$\exists$ TM of correct type running with resource bound such-and-such

"Correct type": DTM – easy to check } syntactic
NDTM – easy to check

BPP-style: accept x with prob $\geq 2/3$
or prob $\leq 1/3$
but not in between

Undecidable to check

Hierarchy theorems?

Fail for similar reasons.

A final useful notion: Randomized reductions

DEF 14 Language B reduces to language C under randomized reductions, denoted $B \leq_r C$, if $\exists$ PTM M s.t.
$$\forall x \in \{0,1\}^* \quad \Pr[C(M(x)) = B(x)] \geq 2/3$$

Not transitive
But if $C \in BPP$ and $B \leq_r C$ then $B \in BPP$

Could have defined NP in terms of randomized reductions instead (if BPP better formalization of "efficient computation")

R XII

$NP = \{ L \mid L \leq_p 3\text{-}SAT \}$

DEF 15

$BP \cdot NP = \{ L \mid L \leq_r 3\text{-}SAT \}$