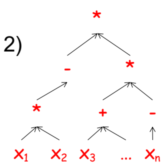
CS151
Complexity
Theory

Lecture 7
April 25, 2023

1

2. Polynomial identity testing

- Given: polynomial $p(x_1, x_2, \ldots, x_n)$ as arithmetic formula (fan-out 1):
  - multiplication (fan-in 2)
  - addition (fan-in 2)
  - negation (fan-in 1)



2

Polynomial identity testing

- Question: Is p identically zero?
  - i.e., is $p(\mathbf{x}) = 0$ for all $\mathbf{x} \in \mathbf{F}^n$
  - (assume $|\mathbf{F}|$ larger than degree…)
- "polynomial identity testing" because given two polynomials p, q, we can check the identity $p \equiv q$ by checking if $(p - q) \equiv 0$

3

Polynomial identity testing

- try all $|\mathbf{F}|^n$ inputs?
  - may be exponentially many
- multiply out symbolically, check that all coefficients are zero?
  - may be exponentially many coefficients
- can randomness help?
  - i.e., flip coins, allow small probability of wrong answer

4

Polynomial identity testing

Lemma (Schwartz-Zippel): Let

$$p(x_1, x_2, \ldots, x_n)$$

be a total degree d polynomial over a field **F** and let S be any subset of **F**. Then if p is not identically 0,

$$\Pr_{r_1, r_2, \ldots, r_n \in S}[\, p(r_1, r_2, \ldots, r_n) = 0] \leq d/|S|.$$

5

Polynomial identity testing

- Proof:
  - induction on number of variables n
  - base case: n = 1, p is univariate polynomial of degree at most d
  - at most d roots, so

$$\Pr[\, p(r_1) = 0] \leq d/|S|$$

6

Polynomial identity testing

– write $p(x_1, x_2, \ldots, x_n)$ as

$p(x_1, x_2, \ldots, x_n) = \Sigma_i (x_1)^i p_i(x_2, \ldots, x_n)$

– k = max. i for which $p_i(x_2, \ldots, x_n)$ not id. zero

– by induction hypothesis:

$\Pr[\ p_k(r_2, \ldots, r_n) = 0] \leq (d-k)/|S|$

– whenever $p_k(r_2, \ldots, r_n) \neq 0$, $p(x_1, r_2, \ldots, r_n)$ is a univariate polynomial of degree k

$\Pr[p(r_1,r_2,\ldots,r_n)=0 \mid p_k(r_2,\ldots,r_n) \neq 0] \leq k/|S|$

7

Polynomial identity testing

$\Pr[\ p_k(r_2, \ldots, r_n) = 0] \leq (d-k)/|S|$

$\Pr[p(r_1,r_2,\ldots,r_n)=0 \mid p_k(r_2,\ldots,r_n) \neq 0] \leq k/|S|$

– conclude:

$\Pr[\ p(r_1, \ldots, r_n) = 0] \leq (d-k)/|S| + k/|S| = d/|S|$

– Note: can add these probabilities because

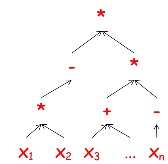
$\Pr[E_1] = \Pr[E_1|E_2]\Pr[E_2] + \Pr[E_1|\neg E_2]\Pr[\neg E_2]$

$\leq \Pr[E_2] + \Pr[E_1|\neg E_2]$

8

Polynomial identity testing

- Given: polynomial $p(x_1, x_2, \ldots, x_n)$
- Is p identically zero?



- Note: degree d is at most the size of input

9

Polynomial identity testing

- randomized algorithm: field **F**, pick a subset $S \in \mathbf{F}$ of size $2d$
  - pick $r_1, r_2, \ldots, r_n$ from S uniformly at random
  - if $p(r_1, r_2, \ldots, r_n) = 0$, answer "yes"
  - if $p(r_1, r_2, \ldots, r_n) \neq 0$, answer "no"
- if p identically zero, never wrong
- if not, Schwartz-Zippel ensures probability of error at most ½

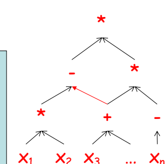
10

Polynomial identity testing

- Given: polynomial $p(x_1, x_2, \ldots, x_n)$
- Is p identically zero?

What if polynomial is given as arithmetic circuit?

- max degree?
- does the same strategy work?



11

3. Unique solutions

- a positive instance of SAT may have many satisfying assignments
- maybe the difficulty comes from not knowing which to "work on"
- if we knew # satisfying assignments was 1 or 0, could we zoom in on the 1 efficiently?

12

Unique solutions

Question: given polynomial-time algorithm that works on SAT instances with at most 1 satisfying assignment, can we solve general SAT instances efficiently?

- Answer: yes
  - but (currently) only if "efficiently" allows randomness

13

Unique solutions

Theorem (Valiant-Vazirani): there is a randomized poly-time procedure that given a 3-CNF formula

$$\varphi(x_1, x_2, \ldots, x_n)$$

outputs a 3-CNF formula φ' such that
- if φ is not satisfiable then φ' is not satisfiable
- if φ is satisfiable then with probability at least $1/(8n)$ φ' has exactly one satisfying assignment

14

Unique solutions

- Proof:
  - given subset $S \in \{1, 2, \ldots, n\}$, there exists a 3-CNF formula θ_S on $x_1, x_2, \ldots, x_n$ and additional variables such that:
    - θ_S is satisfiable iff an even number of variables in $\{x_i\}_{i\in S}$ are true
    - for each such setting of the x_i variables, this satisfying assignment is unique
    - $|\theta_S| = O(n)$
    - not difficult; details omitted

15

Unique solutions

- set $\varphi_0 = \varphi$
- for $i = 1, 2, \ldots, n$
  - pick random subset S_i
  - set $\varphi_i = \varphi_{i-1} \wedge \theta_{S_i}$
- output random one of the φ_i

- T = set of satisfying assignments for φ
- Claim: if $|T| > 0$, then

$$\Pr_{k\in\{0,1,2,\ldots,n-1\}}[2^k \le |T| \le 2^{k+1}] \ge 1/n$$

16

Unique solutions

Claim: if $2^k \le |T| \le 2^{k+1}$, then the probability φ_{k+2} has exactly one satisfying assignment is $\ge 1/8$

- fix $t, t' \in T$

t = 0101 $\boxed{00101}$ 0111

t' = 1010 $\boxed{111000}$ 101

S_i

S_i contains even # of positions i where $t_i \ne t_i'$

- $\Pr[t$ "agrees with" t' on $S_i] = ½$
- $\Pr[t$ agrees with t' on $S_1, S_2, \ldots, S_{k+2}] = (½)^{k+2}$

17

Unique solutions

- $\Pr[t$ agrees with *some* t' on $S_1,\ldots, S_{k+2}]$

$$\le (|T|-1)(½)^{k+2} < ½$$

- $\Pr[t$ satisfies $S_1, S_2, \ldots, S_{k+2}] = (½)^{k+2}$
- $\Pr[t$ *unique* satisfying assignment of $\varphi_{k+2}]$

$$> (½)^{k+3}$$

- sum over at least 2^k different $t \in T$ (disjoint events); claim follows.

18

Randomized complexity classes

- model: probabilistic Turing Machine
  - deterministic TM with additional read-only tape containing "coin flips"
- **BPP** (Bounded-error Probabilistic Poly-time)
  - L ∈ **BPP** if there is a p.p.t. TM M:

    $x \in L \Rightarrow \Pr_y[M(x,y) \text{ accepts}] \geq 2/3$

    $x \notin L \Rightarrow \Pr_y[M(x,y) \text{ rejects}] \geq 2/3$
  - "p.p.t" = probabilistic polynomial time

April 27, 2023 CS151 Lecture 8

19

Randomized complexity classes

- **RP** (Random Polynomial-time)
  - L ∈ **RP** if there is a p.p.t. TM M:

    $x \in L \Rightarrow \Pr_y[M(x,y) \text{ accepts}] \geq ½$

    $x \notin L \Rightarrow \Pr_y[M(x,y) \text{ rejects}] = 1$
- **coRP** (complement of Random Polynomial-time)
  - L ∈ **coRP** if there is a p.p.t. TM M:

    $x \in L \Rightarrow \Pr_y[M(x,y) \text{ accepts}] = 1$

    $x \notin L \Rightarrow \Pr_y[M(x,y) \text{ rejects}] \geq ½$

April 27, 2023 CS151 Lecture 8

20

Randomized complexity classes

One more important class:

- **ZPP** (Zero-error Probabilistic Poly-time)
  - **ZPP = RP ∩ coRP**
  - $\Pr_y[M(x,y)$ outputs "fail"$] \leq ½$
  - otherwise outputs correct answer

April 27, 2023 CS151 Lecture 8

21

Randomized complexity classes

These classes may capture "efficiently computable" better than **P**.

- "1/2" in **ZPP**, **RP**, **coRP** definition unimportant
  - can replace by 1/poly(n)
- "2/3" in **BPP** definition unimportant
  - can replace by ½ + 1/poly(n)
- Why? error reduction
  - we will see simple error reduction by repetition
  - more sophisticated error reduction later

April 27, 2023 CS151 Lecture 8

22

Error reduction for RP

- given L and p.p.t TM M:

  $x \in L \Rightarrow \Pr_y[M(x,y) \text{ accepts}] \geq \varepsilon$

  $x \notin L \Rightarrow \Pr_y[M(x,y) \text{ rejects}] = 1$
- new p.p.t TM M':
  - simulate M k/ε times, each time with independent coin flips
  - accept if **any** simulation accepts
  - otherwise reject

April 27, 2023 CS151 Lecture 8

23

Error reduction

$x \in L \Rightarrow \Pr_y[M(x,y) \text{ accepts}] \geq \varepsilon$

$x \notin L \Rightarrow \Pr_y[M(x,y) \text{ rejects}] = 1$

- if x ∈ L:
  - probability a given simulation "bad" $\leq (1 - \varepsilon)$
  - probability all simulations "bad" $\leq (1-\varepsilon)^{(k/\varepsilon)} \leq e^{-k}$

    $\Pr_{y'}[M'(x, y') \text{ accepts}] \geq 1 - e^{-k}$
- if x ∉ L:

  $\Pr_{y'}[M'(x,y') \text{ rejects}] = 1$

April 27, 2023 CS151 Lecture 8

24

Error reduction for BPP

- given L, and p.p.t. TM M:
  - $x \in L \Rightarrow \Pr_y[M(x,y) \text{ accepts}] \geq ½ + \varepsilon$
  - $x \notin L \Rightarrow \Pr_y[M(x,y) \text{ rejects}] \geq ½ + \varepsilon$
- new p.p.t. TM M':
  - simulate M k/ε^2 times, each time with independent coin flips
  - accept if **majority** of simulations accept
  - otherwise reject

April 27, 2023 CS151 Lecture 8

25

Error reduction for BPP

- X_i random variable indicating "correct" outcome in i-th simulation (out of $m = k/\varepsilon^2$)
  - $\Pr[X_i = 1] \geq ½ + \varepsilon$
  - $\Pr[X_i = 0] \leq ½ - \varepsilon$
- $E[X_i] \geq ½+\varepsilon$
- $X = \Sigma_i X_i$
- $\mu = E[X] \geq (½ + \varepsilon)m$
- Chernoff: $\Pr[X \leq m/2] \leq 2^{-\Omega(\varepsilon^2 \mu)}$

April 27, 2023 CS151 Lecture 8

26

Error reduction for BPP

$x \in L \Rightarrow \Pr_y[M(x,y) \text{ accepts}] \geq ½ + \varepsilon$

$x \notin L \Rightarrow \Pr_y[M(x,y) \text{ rejects}] \geq ½ + \varepsilon$

- if $x \in L$

  $\Pr_{y'}[M'(x, y') \text{ accepts}] \geq 1 - (½)^{\Omega(k)}$
- if $x \notin L$

  $\Pr_{y'}[M'(x,y') \text{ rejects}] \geq 1 - (½)^{\Omega(k)}$

April 27, 2023 CS151 Lecture 8

27

Randomized complexity classes

- We have shown:
  - polynomial identity testing is in **coRP**
  - a poly-time algorithm for detecting unique solutions to SAT implies

    NP = RP

April 27, 2023 CS151 Lecture 8

28

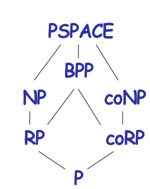
Relationship to other classes

- ZPP, RP, coRP, BPP, contain **P**
  - they can simply ignore the tape with coin flips
- all are in **PSPACE**
  - can exhaustively try all strings y
  - count accepts/rejects; compute probability
- **RP ⊆ NP** (and **coRP** ⊆ **coNP**)
  - multitude of accepting computations
  - **NP** requires only one

April 27, 2023 CS151 Lecture 8

29

Relationship to other classes



April 27, 2023 CS151 Lecture 8

30

BPP

- How powerful is **BPP**?
- We have seen an example of a problem in **BPP** that we only know how to solve in **EXP**.

Is randomness a panacea for intractability?

April 27, 2023 CS151 Lecture 8

31

BPP

- It is not known if **BPP = EXP** (or even **NEXP!**)
  - but there are strong hints that it does not
- Is there a deterministic simulation of **BPP** that does better than brute-force search?
  - yes, if allow non-uniformity

Theorem (Adleman): **BPP** $\subseteq$ **P/poly**

April 27, 2023 CS151 Lecture 8

32

BPP and Boolean circuits

- Proof:
  - language L $\in$ **BPP**
  - error reduction gives TM M such that
    - if x $\in$ L of length n
      $\Pr_y[M(x, y) \text{ accepts}] \geq 1 - (½)^{n^2}$
    - if x $\notin$ L of length n
      $\Pr_y[M(x, y) \text{ rejects}] \geq 1 - (½)^{n^2}$

April 27, 2023 CS151 Lecture 8

33

BPP and Boolean circuits

- say "y is bad for x" if M(x,y) gives incorrect answer
- for fixed x: $\Pr_y[y \text{ is bad for } x] \leq (½)^{n^2}$
- $\Pr_y[y \text{ is bad for some } x] \leq 2^n(½)^{n^2} < 1$
- Conclude: there exists some y on which M(x, y) is always correct
- build circuit for M, hardwire this y

April 27, 2023 CS151 Lecture 8

34

BPP and Boolean circuits

- Does **BPP** = **EXP** ?
- Adleman's Theorem shows:

  BPP = **EXP** implies **EXP** $\subseteq$ **P/poly**

If you believe that randomness is all-powerful, you must also believe that non-uniformity gives an exponential advantage.

April 27, 2023 CS151 Lecture 8

35

BPP

- Next:

  further explore the relationship between

  randomness

  and

  nonuniformity

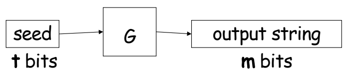
- Main tool: pseudo-random generators

April 27, 2023 CS151 Lecture 8

36

Derandomization

- Goal: try to simulate BPP in subexponential time (or better)
- use Pseudo-Random Generator (PRG):



- often: PRG "good" if it passes (ad-hoc) statistical tests

Derandomization

- ad-hoc tests not good enough to prove BPP has non-trivial simulations
- Our requirements:
  - G is efficiently computable
  - "stretches" **t** bits into **m** bits
  - "fools" small circuits: for all circuits C of size at most **s**:

$$|\Pr_y[C(y) = 1] - \Pr_z[C(G(z)) = 1]| \leq \boldsymbol{\varepsilon}$$

Simulating **BPP** using PRGs

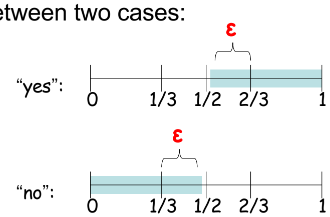
- Recall: $L \in$ **BPP** implies exists p.p.t.TM M
  - $x \in L \Rightarrow \Pr_y[M(x,y) \text{ accepts}] \geq 2/3$
  - $x \notin L \Rightarrow \Pr_y[M(x,y) \text{ rejects}] \geq 2/3$
- given an input x:
  - convert M into circuit C(x, y)
  - simplification: pad y so that |C| = |y| = m
- hardwire input x to get circuit C_x
  - $\Pr_y[C_x(y) = 1] \geq 2/3$ ("yes")
  - $\Pr_y[C_x(y) = 1] \leq 1/3$ ("no")

Simulating **BPP** using PRGs

- Use a PRG G with
  - output length **m**
  - seed length **t** « m
  - error **ε** < 1/6
  - fooling size **s** = m
- Compute $\Pr_z[C_x(G(z)) = 1]$ exactly
  - evaluate $C_x(G(z))$ on every seed $z \in \{0,1\}^t$
- running time $(O(m)+(\text{time for G}))\mathbf{2^t}$

Simulating **BPP** using PRGs

- knowing $\Pr_z[C_x(G(z)) = 1]$, can distinguish between two cases:



Blum-Micali-Yao PRG

- Initial goal: for all 1 > δ > 0, we will build a family of PRGs $\{G_m\}$ with:
  - output length **m**
  - fooling size **s** = m
  - seed length $\mathbf{t} = m^{\delta}$
  - running time m^c
  - error **ε** < 1/6
- implies: $\mathbf{BPP} \subseteq \cap_{\delta>0} \mathbf{TIME}(2^{n^\delta}) \subsetneq \mathbf{EXP}$
- Why? simulation runs in time $O(m+m^c)(2^{m^\delta}) = O(2^{m^{2\delta}}) = O(2^{n^{2k\delta}})$