CS21 Decidability and Tractability

Lecture 25

March 4, 2024

1

Outline

- Challenges to Extended Church-Turing
  - randomized computation
  - quantum computation

2

Extended Church-Turing Thesis

- the belief that TMs formalize our intuitive notion of an efficient algorithm is:

> <u>The “extended” Church-Turing Thesis</u>
>
> everything we can compute in time t(n) on a physical computer can be computed on a Turing Machine in time $t(n)^{O(1)}$ (polynomial slowdown)

- randomized computation challenges this belief

3

Randomness in computation

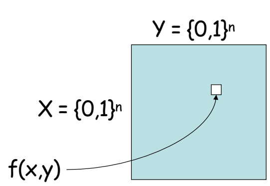
- Example of the power of randomness
- Randomized complexity classes

4

Communication complexity

Theorem: no deterministic protocol can compute EQ(x, y) while exchanging fewer than n+1 bits.

- Proof:
  - “input matrix”:



5

Communication complexity

- Can we do better?
  - deterministic protocol?
  - probabilistic protocol?
    - at each step: one party sends bits that are a function of held input and received bits so far and the result of some coin tosses
    - required to output f(x, y) with high probability over all coin tosses

6

Communication complexity

- protocol for EQ employing randomness?
  - Alice picks random prime p in $\{1...4n^2\}$, sends:
    - p
    - (x mod p)
  - Bob sends:
    - (y mod p)
  - players output 1 if and only if:

(x mod p) = (y mod p)

7

Communication complexity

- O(log n) bits exchanged
- if x = y, always correct
- if x ≠ y, incorrect if and only if:

p divides |x – y|

- # primes in range is ≥ 2n
- # primes dividing |x – y| is ≤ n
- probability incorrect ≤ 1/2

Randomness gives an exponential advantage!!

8

Communication complexity

two parties: Alice and Bob
function $f:\{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$
Alice holds $x \in \{0,1\}^n$; Bob holds $y \in \{0,1\}^n$

- Goal: compute f(x, y) while communicating as few bits as possible between Alice and Bob

Example: EQ(x, y) = 1 iff x = y

- Deterministic protocol: no fewer than n+1 bits
- Randomized protocol: O(log n) bits

9

Extended Church-Turing Thesis

- Common to insert "probabilistic":

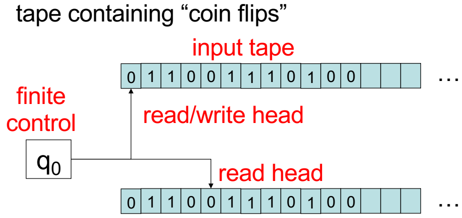
The "extended" Church-Turing Thesis

everything we can compute in time t(n) on a physical computer can be computed on a *probabilistic* Turing Machine in time $t(n)^{O(1)}$ (polynomial slowdown)

10

Randomized complexity classes

- model: probabilistic Turing Machine
  - deterministic TM with additional read-only tape containing "coin flips"



11

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:
    - $x \in L \rightarrow Pr_y[M(x,y) \text{ accepts}] = 1$
    - $x \notin L \rightarrow Pr_y[M(x,y) \text{ rejects}] \geq ½$

"p.p.t" = probabilistic polynomial time

12

Randomized complexity classes

- **BPP** (Bounded-error Probabilistic Poly-time)
  - L ∈ **BPP** if there is a p.p.t. TM M:
    - $x \in L \rightarrow \Pr_y[M(x,y) \text{ accepts}] \geq 2/3$
    - $x \notin L \rightarrow \Pr_y[M(x,y) \text{ rejects}] \geq 2/3$

13

Randomized complexity classes

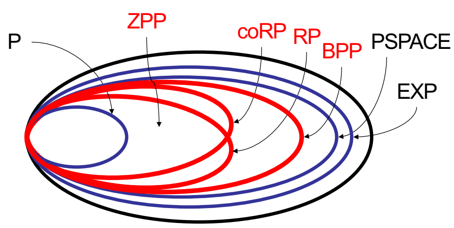
These classes may capture "efficiently computable" better than **P**.

One more important class:

- **ZPP** (Zero-error Probabilistic Poly-time)
  - **ZPP = RP ∩ coRP**
  - $\Pr_y[M(x,y) \text{ outputs "fail"}] \leq 1/2$
  - otherwise outputs correct answer

14

RP,coRP, BPP



- from definitions: ZPP ⊆ RP, coRP ⊆ BPP

15

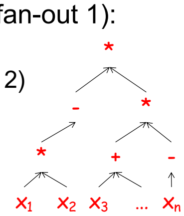
Relationship to other classes

- all these classes 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

16

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)



variables take values in finite field F

17

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 $(p - q) \equiv 0$

18

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
- Best known deterministic algorithm places in EXP

Polynomial identity testing

Lemma (Schwartz-Zippel): Let

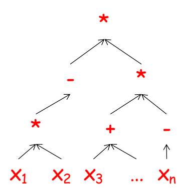
$$p(x_1, x_2, \ldots, x_n)$$

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,

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

Polynomial identity testing

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



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

Polynomial identity testing

- randomized algorithm: pick a subset $S \subseteq \mathbf{F}$ 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

Randomized complexity classes

- We have shown:
  - Polynomial Identity Testing is in coRP
  - note: no sub-exponential time deterministic algorithm know

Randomized complexity classes

- How powerful is randomized computation?
- We have seen an example of a problem in

BPP

that we only know how to solve deterministically in **EXP**.

Is randomness a panacea for intractability?