

CS21 Decidability and Tractability

Lecture 24
March 7, 2008

March 7, 2008

CS21 Lecture 24

1

Outline

- “Challenges to the (extended) Church-Turing Thesis”
 - randomized computation
 - quantum computation

March 7, 2008

CS21 Lecture 24

2

Extended Church-Turing Thesis

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

The “extended” Church-Turing Thesis

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

March 7, 2008

CS21 Lecture 24

3

Randomness in computation

- Example of the power of randomness
- Randomized complexity classes

March 7, 2008

CS21 Lecture 24

4

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
- count number of bits exchanged (computation free)
- at each step: one party sends bits that are a function of held input and received bits so far

March 7, 2008

CS21 Lecture 24

5

Communication complexity

- simple function (equality):

$$EQ(x, y) = 1 \text{ iff } x = y$$

- simple protocol:
 - Alice sends x to Bob (n bits)
 - Bob sends $EQ(x, y)$ to Alice (1 bit)
 - total: $n + 1$ bits
 - (works for any predicate f)

March 7, 2008

CS21 Lecture 24

6

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

March 7, 2008

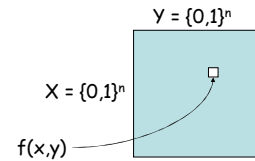
CS21 Lecture 24

7

Communication complexity

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

- Proof:
 - “input matrix”:



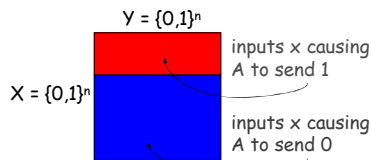
March 7, 2008

CS21 Lecture 24

8

Communication complexity

- assume 1 bit sent at a time (but proof works for general case)
- A sends 1 bit depending only on x :



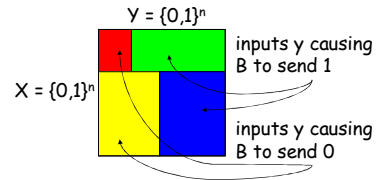
March 7, 2008

CS21 Lecture 24

9

Communication complexity

- B sends 1 bit depending only on y and received bit:



March 7, 2008

CS21 Lecture 24

10

Communication complexity

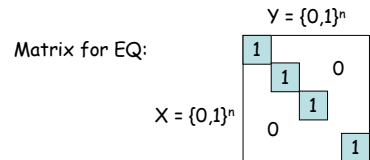
- at end of protocol involving k bits of communication, matrix is partitioned into at most 2^k combinatorial rectangles
- bits sent in protocol are the same for every input (x, y) in given rectangle
- conclude: $f(x, y)$ must be constant on each rectangle

March 7, 2008

CS21 Lecture 24

11

Communication complexity



- any partition into combinatorial rectangles with constant $f(x, y)$ must have at least $2^n + 1$ rectangles
- protocol that exchanges $\leq n$ bits can only create 2^n rectangles, so must exchange at least $n+1$ bits.

March 7, 2008

CS21 Lecture 24

12

Communication complexity

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

$$(x \bmod p) = (y \bmod p)$$

March 7, 2008

CS21 Lecture 24

13

Communication complexity

- $O(\log n)$ bits exchanged
 - if $x = y$, always correct
 - if $x \neq y$, incorrect if and only if:
 - p divides $|x - y|$
 - # primes in range is $\geq 2n$
 - # primes dividing $|x - y|$ is $\leq n$
 - probability incorrect $\leq 1/2$
- Randomness gives an exponential advantage!!

March 7, 2008

CS21 Lecture 24

14

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

March 7, 2008

CS21 Lecture 24

15

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)

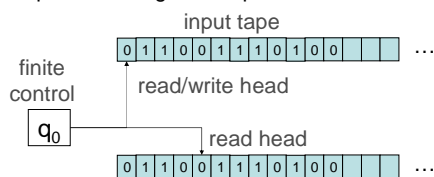
March 7, 2008

CS21 Lecture 24

16

Randomized complexity classes

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



March 7, 2008

CS21 Lecture 24

17

Randomized complexity classes

- **RP** (Random Polynomial-time)
 - $L \in \mathbf{RP}$ if there is a p.p.t. TM M :
 - $x \in L \Rightarrow \Pr_y[M(x,y) \text{ accepts}] \geq \frac{1}{2}$
 - $x \notin L \Rightarrow \Pr_y[M(x,y) \text{ rejects}] = 1$
 - **coRP** (complement of Random Polynomial-time)
 - $L \in \mathbf{coRP}$ if there is a p.p.t. TM M :
 - $x \in L \Rightarrow \Pr_y[M(x,y) \text{ accepts}] = 1$
 - $x \notin L \Rightarrow \Pr_y[M(x,y) \text{ rejects}] \geq \frac{1}{2}$
- “p.p.t.” = probabilistic polynomial time

March 7, 2008

CS21 Lecture 24

18

Randomized complexity classes

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

March 7, 2008

CS21 Lecture 24

19

Randomized complexity classes

These classes may capture “efficiently computable” better than **P**.

- “1/2” in **RP**, **coRP** definition unimportant
 - can replace by $1/\text{poly}(n)$, amplify to $1-2^{-\text{poly}(n)}$
- “2/3” in **BPP** definition unimportant
 - can replace by $1/2 + 1/\text{poly}(n)$, amplify to $1-2^{-\text{poly}(n)}$
- Why? error reduction
 - simple error reduction by repetition

March 7, 2008

CS21 Lecture 24

20

Error reduction for RP

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

March 7, 2008

CS21 Lecture 24

21

Error reduction

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

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

- if $x \in L$:
 - probability a given simulation “bad” $\leq (1 - \epsilon)$
 - probability all simulations “bad” $\leq (1 - \epsilon)^{(k/\epsilon)} \leq e^{-k}$
 - $\Pr_y[M'(x, y') \text{ accepts}] \geq 1 - e^{-k}$
- if $x \notin L$:
 - $\Pr_y[M'(x, y') \text{ rejects}] = 1$

March 7, 2008

CS21 Lecture 24

22

Error reduction for BPP

- given L , and p.p.t. TM M :
 - $x \in L \Rightarrow \Pr_y[M(x,y) \text{ accepts}] \geq 1/2 + \epsilon$
 - $x \notin L \Rightarrow \Pr_y[M(x,y) \text{ rejects}] \geq 1/2 + \epsilon$
- new p.p.t. TM M' :
 - simulate M $O(k/\epsilon^2)$ times, each time with independent coin flips
 - accept if **majority** of simulations accept
 - otherwise reject
 - error probability reduced to $\leq e^{-k}$

March 7, 2008

CS21 Lecture 24

23

Randomized complexity classes

One more important class:

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

March 7, 2008

CS21 Lecture 24

24

RP, coRP, BPP

• from definitions: $ZPP \subset RP$, $coRP \subset BPP$

March 7, 2008 CS21 Lecture 24 25

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 \subset NP** (and **coRP \subset coNP**)
 - multitude of accepting computations
 - **NP** requires only one

March 7, 2008 CS21 Lecture 24 26