CS21
Decidability and Tractability

Lecture 25

March 7, 2014

Outline

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

Challenges to the extended Church-Turing thesis

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

Randomness in computation

- Example of the power of randomness
- Randomized complexity classes

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

Communication complexity

- simple function (equality):

  EQ(x, y) = 1 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)

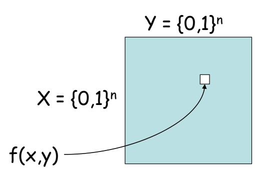
Communication complexity

- 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

Communication complexity

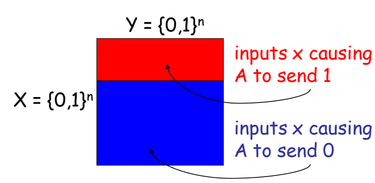
Theorem: no deterministic protocol can compute EQ(x, y) while exchanging fewer than n+1 bits.

- Proof:
  – "input matrix":



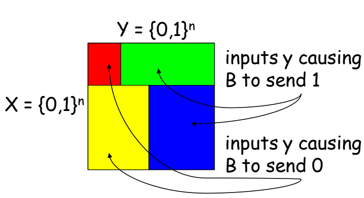
Communication complexity

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



Communication complexity

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



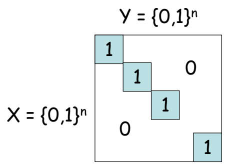
Communication complexity

– 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

Communication complexity

Matrix for EQ:



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

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

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)

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!!

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

Extended Church-Turing Thesis

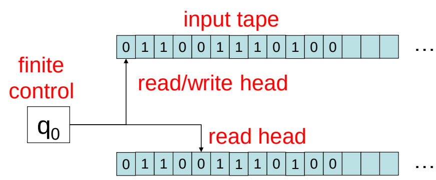
- Common to insert "probabilistic":

<u>The "extended" Church-Turing Thesis</u>

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)

Randomized complexity classes

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



Randomized complexity classes

- **RP** (Random Polynomial-time)
  - $L \in$ **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)
  - $L \in$ **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

Randomized complexity classes

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

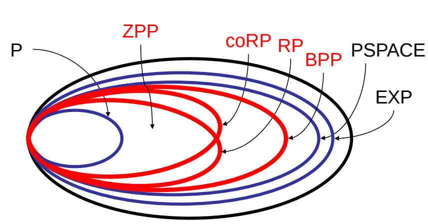
Randomized complexity classes

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

One more important class:

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

RP,coRP, BPP



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