COMS3261: Computer Science Theory

Fall 2013

Mihalis Yannakakis

Lecture 24, 12/2/13

Time Complexity

- Time complexity of a Deterministic TM M:
  T(n) = max # steps of M on inputs of length $\leq$n
  worst-case complexity: max

- Asymptotics: Big-Oh, small-o notation
- For f,g positive functions on positive integers,
  f=O(g) if $\exists$ constant c, integer n_0 s.t. $f(n) \leq cg(n)$ $\forall n \geq n_0$
- Example: $3n^2+7n=O(n^2) =O(n^3)$

- $f=\Theta(g)$ if $\exists$ constants c_1,c_2, integer n_0 s.t. $c_1 g(n) \leq f(n) \leq c_2 g(n)$ $\forall n \geq n_0$

Time Complexity Class

- Let t: N→N
- TIME(t(n)) = { L | language L decided by a (multitape) DTM in time O(t(n)) }
- i.e. $L \in TIME(t(n)) \Leftrightarrow \exists$ DTM M $\exists$constant c, integer n_0 s.t L(M)=L and $T_M(n) \leq ct(n)\ \forall n \geq n_0$
- Similarly for other computational problems that are not decision (language) problems
- Are allowed to use multitape TM

- O(t(n)) in definition because of linear speed-up theorem: can use a TM with more states and tape symbols to speed up computation by a constant factor (above O(n) time)

Time Complexity Class

- Are allowed to use multitape TM
- If L in TIME(t(n)) then recognized in $O(t^2(n))$ time by 1-tape TMs

- Simulation of real computers: if t(n) on RC then $O(t^3(n))$ by multitape Turing machine (generous – more carefully it is more like $t^2(n)$ – effects of random access memory versus sequential access as in tape of TMs)

- Different models differ in time complexity by small polynomial changes (e.g. t vs t^2)
- Important for low complexities (n, nlogn, n^2), but not important if we want to distinguish polynomial vs. exponential

Polynomial Time

- polynomial examples: n, n^2, $3n^2+7n+2$, $6n^4+3n^2$, n^{100}
- $p(n)=\Theta(n^d)$ where d the degree of the polynomial (leading term)
- Defn: $P = \bigcup_{k=1}^{\infty} TIME(n^k)$
- All reasonable models of computation can simulate each other with polynomial overhead
- Robust class
- Can use 1-tape TM for lower bounds
- Can use real computer (RAM) for upper bounds (algorithms)
- Robust also with respect to reasonable input representations

Examples of Problems in P (1)

- Sorting a set of numbers, strings etc: nlogn
- Membership in a regular language: linear in length n of string
  (example apps: pattern matching, lexical analysis, etc)
  - polynomial time also with respect to the size of the representation of the language, for all types of representations (DFA, NFA, ε-NFA, Regular expression)
- Membership in a context-free language: n^3 (n for DCFL) for string of length n
  (ex. parsing)

Examples of Problems in P (2)

- Graph problems

- Graph Reachability, Cycle detection, Shortest paths, Minimum Spanning Tree…
- Representation of graphs: by adjacency matrix, or adjacency lists, or set of nodes and edges.

  - Important for exact complexity, but all reasonable representations polynomially related, so exact representation does not matter for membership in P

Examples of Problems in P (3)

- Problems with numbers: numbers represented in binary: size = #bits (magnitude of number exponential in size!)

- Basic operations on numbers: +, - , *, /
- Divisibility : given a,b in Z, does a|b?
- Greatest common divisor (Euclidean algorithm)
- Naive algorithm (try every d $\leq$a,b, pick max) not polynomial

- Primality, Compositeness. In P
- Factorization: Given number a, compute its prime factors. Big open problem.
- Basis of cryptographic schemes: assumption: not in P

Examples of Problems in P (4)

- Linear algebra: Solution of linear equations, matrix operations (inversion, multiplication etc)

- Solution of linear inequalities and linear optimization (Linear Programming)

Nondeterministic Time Complexity

- For a Nondeterministic Turing machine M (that always halts), time complexity $T_M(n)$ = max # steps used on *any branch* of the computation on *any input* of length $\leq n$.

  i.e. time f(n) means that M halts in time $\leq f(n)$ for *all* computations on *all inputs* of length $\leq n$; not only accepting computations.
- (If some branch does not halt then time = infinite)

- NTIME(t(n)) = { L | language L accepted by a (multitape) NTM with time complexity O(t(n)) }

Nondeterministic vs. Deterministic Time

- Let $t(n) \geq n$.
- Every $t(n)$ time NTM has an equivalent (1-tape) DTM with time complexity $c^{t(n)}$ for some constant c (equivalently time $2^{O(t(n))}$)
- Showed this in the simulation of NTM with DTM.
- Big Open Question: Is the exponential blowup unavoidable?
- Especially important in the special case of polynomial time complexity: P=NP ? question

Nondeterministic Polynomial Time: NP

- Definition of NP: Class of languages L (decision problems: Yes/No problems) that can be accepted by some Nondeterministic Turing Machine that runs in polynomial time, i.e. time $O(n^k)$ for some constant k.

$$NP = \bigcup_{k=1}^{\infty} NTIME(n^k)$$

- Alternative equivalent view of NP: Using certificates (witnesses, proofs,..): All guessing done up front at the beginning.

Alternative view of NP: Certificates

- Theorem: A language $L \subseteq A^*$ is in NP iff there is a 2-ary relation $R \subseteq A^* \times B^*$ (B could be A or {0,1} or any alphabet)

  that is

  1. polynomially balanced: $R(x,y) \Rightarrow |y| \leq |x|^c$ for some c

     (could also require = instead of ≤ , equivalent)

  2. polynomially decidable: there is a polynomial-time (deterministic) TM that accepts (x,y) iff R(x,y)

  such that $L = \{ x \mid \exists y \text{ s.t. } R(x,y) \}$

Alternative view of NP: Certificates

$L = \{ x \mid \exists y \text{ s.t. } R(x,y) \}$

Means:

- An input x is in L iff there is a certificate (witness, solution, proof) y such that R(x,y) holds.

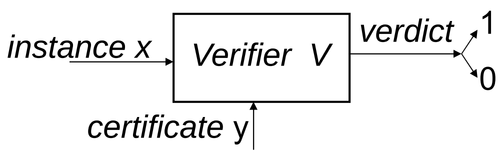
- Two important properties:

1. certificates are short (polynomially bounded)
2. certificates are easy to check (polynomial-time checkable)

NP = short, verifiable certificates

- Verifier V: Polynomial time algorithm for the relation R(x,y); V checks that y is a certificate for x

$$L = \{x \in A^* \mid \exists y \in B^*,\ |y| \le |x|^c,\ V(x,y) = 1\}$$



- $x \in L \;\Rightarrow\; \exists y \in B^*,\ |y| \le |x|^c$ s.t. $V(x,y) = 1$
- $x \notin L \;\Rightarrow\; \forall y \in B^*,\ |y| \le |x|^c.\ \ V(x,y) = 0$

Proof

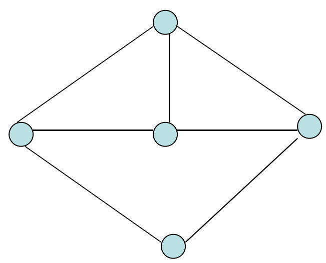
1. Suppose L is accepted by a polynomial time NTM M.
certificate y = sequence of choices of M that makes M accept
(Could also use certificate =complete accepting computation)
R(x,y): M accepts input x with sequence of choices y

2. Suppose L has a relation R that is polynomially balanced and polynomially decidable. Then L accepted by NTM M that guesses first a certificate y of length ≤ $|x|^c$ and then verifies that R(x,y) holds.

Examples

- Graph 3 Colorability: Instance: Graph G

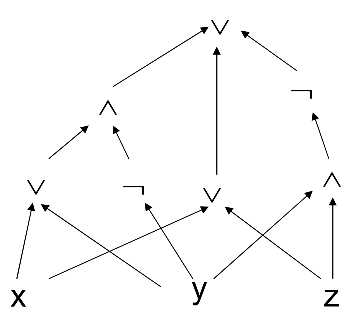
  certificate y = assignment of a “color”$\in$\{1,2,3\} to each node s.t. adjacent nodes are assigned different colors
- Constraint Satisfaction problems, e.g. schedule a set of events (eg. exams) in a given number of slots so that no conflicts



Example: Boolean Circuit Satisfiability

- Input: Boolean (combinational) circuit using AND,OR, NOT gates with 1 output, many inputs
- Output: Yes, if there is an assignment of True/False (1/0) to the inputs that makes the circuit output true (1)

Example:



x=y=z=0 satisfies the circuit

Fundamental Question

P = NP?

Is it always as easy to generate a proof/certificate as it is to check a proof/certificate that is given to us?