Michael Forbes
M.forbes@illinois.edu

2019-04-04.4 → 2019-04-09.1
2019-04-05.2 ← CS579

CS 579 Computational Complexity : Lecture 23 11 | 7

admin : ps6 delayed - due tonight
today : query complexity - extended deadline
so far : comparison of different TM resources - deterministic
- nondeterministic
- space
- non uniformity
- alternation
- randomness
- counting
- interaction

results : - trivial inclusions e.g. $NP \subseteq PSPACE$
- robustness of models, e.g. $BPP_{1/3} = BPP_{1/10}$
- completeness results, e.g. Cook-Levin
- non trivial inclusions, e.g. $BPP \subseteq PH$, $TQBF \in IP$

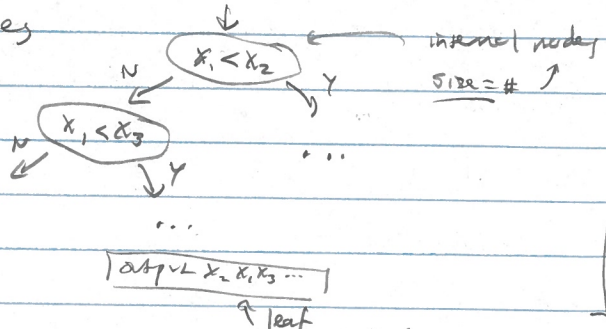
Q : can we prove complexity classes are different?
A : hierarchy thms $TIME(n^3) \subsetneq TIME(n^5)$
Q : anything else?
A : "nothing" else known ← relativization barrier

next two weeks : complexity theory of concrete models of computation
└ not full power of TM ⇒ can prove results!
- capture "natural" computational phenomena
today : query complexity

ex : sorting
given n items $x_1, \dots, x_n$, sort them
↑ abstract ↑ have comparison oracle "$x_i \stackrel{?}{<} x_j$"
∃ many sorting algs. fit this model, not all ∀

recall : └ can be done w/ $O(n \lg n)$ comparisons, e.g. merge sort, heap sort, quicksort, etc

formalism : via decision trees, e.g.
⇒ query complexity



Prop : any sorting alg. in comparison model requires $\Omega(n \lg n)$ depth
Pf : decision tree depth d ⇒ $\leq 2^d$ leaves ∃ binary answers ∀
↑ all $n!$ orderings must appear
⇒ $d \geq \lg(n!) \geq \Omega(n \lg n)$

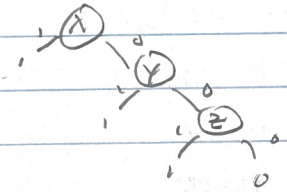
def : a decision tree T on variables $x_1, \dots, x_n$ is a labelled binary tree w/ single root
- leaves $\in \{0,1\}$ ∃ output ∀
- internal nodes labelled w/ variables x_i, children "$x_i = 0$" "$x_i = 1$"
- depth = max distance from root to leaf
- computes $T : \{0,1\}^n \to \{0,1\}$ by - starting at root node
- if at "x_i" labelled node, take "$x_i = b_i$" child, for b_i the value of x_i
- if at leaf, output its value

Michael Forbes
MForbes@illinois.edu
2019-04-09.2 ← → 2019-04-09.3
CS577

length of root → output pair

- cost(T, x) = # of variables queried when computing T(x)

ex: $OR_3(x,y,z)$ =



cost(T, 100) = 1
OR(1) = 1

→ cost ≤ depth
∥ per input ∥ worst case

def: $f: \{0,1\}^n \to \{0,1\}$ $D(f) = \min_{\text{T computing } f} \max_x \text{cost}(T,x)$ = depth(T)

Cor: $D(f) \le n$ ∥ only n vars to query ∥

pf: query $x_1, x_2, \dots, x_n$, output f(x)

Prop: $D(OR_n) \ge n$

pf: use "adversary argument" on depth < n tree T

adversary: feed "$x_i = 0$" to each variable in tree until leaf is reached, output b

as depth < n tree ⇒ some variable not queried, say x_{i_0}

if b = 0: set $x_{i_0} = 1$ ⇒ $OR_n(x) = 1$ but T(x) = 0
∥ consistent ∥

b = 1: $x_i = 0$ all i ⇒ $OR_n(x) = 0$ ∴ ☐

def: $P^{dec} = \{ f: \{0,1\}^n \to \{0,1\} \text{ s.t. } D(f) \le \text{poly}(\log(n)) \}$ ∥ families of functions ∥
"efficient"

Cor: $OR \notin P^{dec}$ ∥ NP ≠ P ∥

def: a b-certificate for $f: \{0,1\}^n \to \{0,1\}$ at x is a set $S \subseteq [n]$, $|S| = $ size of, and values $x|_S \in \{0,1\}^{|S|}$ such that for any $y \in \{0,1\}^n$: $y|_S = x|_S \Rightarrow f(y) = b$ ∥ proof/witness ∥

def: $N_b(f) = \max_x \min_k \{ \exists \text{ } k \text{ size } S \text{ s.t. } f(x) = b \text{ have b-certificate size} \le k \}$
∥ guess variables then check

ex: $N_1(OR_n) = 1$

$N_0(OR_n) = n$ ∥ follows from adversary argument ∥

def: $NP^{dec} = \{ f: \{0,1\}^n \to \{0,1\} \text{ s.t. } N_1(f) \le \text{poly}(\log(n)) \}$
"

$coNP^{dec} = \{ f: \{0,1\}^n \to \{0,1\} \text{ s.t. } N_0(f) \le \text{poly}(\log(n)) \}$
"

lem: $D(f) \ge N_0(f), N_1(f)$

pf: T decision tree for f, any x. $S_x = \{\text{queries of T on x}\}$

⇒ $|S_x| = \text{cost}(T,x) \le D(f)$. variables $\notin S_x$ do not affect T on x

⇒ S_x is $(f(x))$-cert to x ☐

Cor: $P^{dec} \subseteq NP^{dec}, coNP^{dec}$

Cor: $P^{dec} \subsetneq NP^{dec} \ne coNP^{dec}$ ∥ expected ∥
∥ ∥ ∥
OR $N_0(OR_n) \ge n$

Michael Forbes
mforbes@illinois.edu

2019-04-09-2 → 2019-04-09-3
← 2019-04-09-4
CS 579

thm: any f, $D(f) \le N_0(f) \cdot N_1(f)$

Cor: $P^{dec} = NP^{dec} \cap coNP^{dec}$ ∏ unexpected! ∐

Pf: $f \in NP^{dec} \Rightarrow N_1(f) \le poly(\lg n)$
" $coNP$ " N_0 "
$\Rightarrow D(f) \le poly(\lg n) \cdot poly(\lg n) \le poly(\lg n)$
$\Rightarrow f \in P^{dec}$ ☒

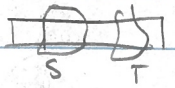
Pf: $k = N_0(f)$ $\ell = N_1(f)$

Idea: Let $C_0 \subseteq \binom{[n]}{k}$ be minimal 0-cert
$C_1 \subseteq \binom{[n]}{\ell}$ " 1 cert

Use queries to reduce C_0, C_1 until function is constant ← solvable w/ no query

Key Lem: $(S, x|_S)$ 0-cert, $(T, x|_T)$ 1-cert
$\Rightarrow |S \cap T| > 0$

Pf: by contradiction, assume $S \cap T = \emptyset$



pick $y \in \{0,1\}^n$ s.t. $y|_S = x|_S \Rightarrow f(y) = 0$
$y|_T = x|_T \Rightarrow f(y) = 1$ ⇒⇐ ☒
(possible as S, T disjoint)

algo: on input x:

1) define $Q \subseteq [n]$ w/ $Q = \emptyset$
2) while $f(x)$ undetermined subject to $x|_Q$ known
   a) pick minimum size 0-cert S for f, consistent w/ $x|_Q$ ← $|S| \le k$
   b) query $x|_{S \setminus Q}$
   c) $Q \leftarrow Q \cup S$
3) output consistent value of f subject to $x|_Q$

correctness: $f(x)$ undetermined s.t. $x|_Q$ iff

$\exists$ 0-cert $(S, y|_S)$ s.t. $x|_{Q \cap S} = y|_{Q \cap S}$ ← consistency
1-cert (T, y_T) $x|_{Q \cap T} = y|_{Q \cap T}$
$\Rightarrow |S \setminus Q| > 0$, $|T \setminus Q| > 0$, $|(S \cap T) \setminus Q| > 0$ ← Key Lemma

hence, $\Rightarrow$ some minimal 0-cert S exists in step 2a, a)
$\Rightarrow$ additional coordinate of x queried in step 2b
$\Rightarrow$ will eventually query enough of x to determine $f(x)$
$\Rightarrow$ converges

complexity: uses $\le k \cdot (\#\text{ rounds})$ queries $\le k \cdot \ell$

Clm: # rounds $\le \ell = N_1(f)$ ⇒

Michael Forbes
miforbes@illinois.edu
2019-04-09.4 ← 2019-04-09.3
2019-04-11.1 →
CS579

consider any 1-cert $(T, y|_T)$ consistent w/ $x|_Q$

$\Rightarrow$ $|T \setminus Q| > 1$ $\quad |(S \cap T) \setminus Q| > 1$

$\Rightarrow$ $|T \setminus (Q \cup S)| \leq |T \setminus Q| - 1$

↑ S queries ≥ 1 value in T

$\Rightarrow$ each round all 1-certs consistent w/ Q either
- become inconsistent w/ $x|_{Q \cup S}$
- have 1 fewer unknown variable

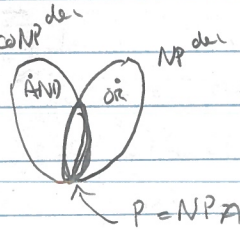
as $\ell = N_0(f)$ each minimal 1-cert starts w/ ℓ unknown variables

$\Rightarrow$ in ℓ rounds
- $\emptyset$ is 1-cert $\Rightarrow f|_{x|_Q} = 1$
- no 1-certs consistent w/ $x|_Q$ $\Rightarrow f|_{x|_Q} = 0$

f on $x|_Q$ is constant

$\Rightarrow$ algo outputs value ∎

hence:



next time - communication complexity.