

Undergraduate Complexity Theory 15-455 @ CMU

TOC

Marcyhm

October 30, 2022

1 Course Overview

resources; efficient; famous open problems; notations ($\langle X \rangle$).

Three types of problems: *decision / function / search.*

2 Turing Machines

language; Turing machine; decider; computation trace; Church-Turing Thesis (extended ver.)

3 Simulation and Turing Machine Variants

Standard model: 1-tape TM.

Variations: 1-way / alphabet / multitape / RAM; Boolean Circuits; TM operation tricks; **simulation**.

4 Time Complexity and Universal Turing Machines

(time) complexity class $\text{TIME}(f(n))$; Speedup Theorem; P; *Universal TM*; ***diagonalization method***.

5 The Time Hierarchy Theorem

Time Hierarchy Theorem (more time, more power).

diagonalization method (important technique: simulation then do the opposite).

time constructible.

6 Problems in P

by THT \exists more than P: EXP.

PATH, 2-COL, LCS, c -CLIQUE.

7 SAT

different forms: CKT-SAT, FORMULA-SAT a.k.a. SAT, CNF-SAT, k -SAT (descending on generality).

All NP-complete; CKT-EVAL not *parallelizable*; *formula*; time complexity about c -SAT.

8 NP

conjectures: $P \neq NP$;

ETH: $\exists \delta > 0 : 3\text{-SAT} \notin \text{TIME}((1 + \delta)^n)$;

SETH: $\forall \delta > 0 : \exists k : k\text{-SAT} \notin \text{TIME}((2 - \delta)^n)$.

verifier-based def for NP. (In fact a trivial interactive proof system w/o randomness.)

9 Nondeterminism

nondeterminism (a non-realistic feature for computation models); NTIME; NP.

the equivalence between different views/defs about NP (nondeterminism / verifier).

10 Reductions

mapping reduction; $3\text{COL} \leq_m^P 4\text{COL} \leq_m^P \text{SAT} \leq_m^P \text{CIRCUIT-SAT} \leq_m^P 3\text{SAT}$. ($3\text{SAT} \leq_m^P 3\text{COL}$ in lec12)

Turing reduction; $4\text{CHROMA} \leq_T^P \text{SAT}$.

These reductions are *transitive*; P is closed under both reductions, but NP only under mapping reduction.

11 NP-Completeness and the Cook-Levin Theorem

NP-hard; NP-complete.

Cook-Levin Theorem: $\forall L \in \text{NP} : L \leq_m^P \text{SAT}$. (proof idea: computation is local.)

Circuit-based proof of Cook-Levin.

Reading: *boolean circuit, circuit family, size/depth complexity*.

12 NP-Completeness Reductions

$3\text{SAT} \leq_m^P \text{E3SAT} \leq_m^P \text{NAE-3SAT} \leq_m^P 3\text{COL} \leq_m^P \text{INDSET}$; *CSP*; *gadget*.

Reading: NP-complete problems, VERTEX-COVER, HAMPATH, UHAMPATH, SUBSET-SUM.

13 Search-to-Decision, Padding, Dichotomy Theorems

SAT is *downward-self-reducible*; **search-to-decision reduction**.

padding (common technique: add meaningless contents to blow input size up);

Dichotomy Theorem: Every boolean CSP is either in P (2SAT, XOR-SAT, HornSAT) or NP-complete.

Dichotomy Conjecture: every CSP is either in P or NP-complete.

14 Ladner's Theorem and Mahaney's Theorem

Ladner's Theorem: $P \neq NP \implies \exists L \in \text{NP} \setminus P$ s.t. L is not NP-complete.

On class gives a weaker proof assuming ETH, with the idea basically the same: “water down” SAT to make it not so hard, thus get a valid algo for SAT which contradicts the assumption.

Mahaney's Theorem: $P \neq NP \implies$ *sparse* NP-complete language not exists.

Proof idea: reduce SAT to L , the width of DSR's search tree is poly (restricted by L), thus $\text{SAT} \in P$.

15 coNP

coNP; P is closed under complement; UNSAT is coNP-complete.

$L \in \text{NP} \cap \text{coNP}$ has “good characterization”.

16 Space Complexity

model for space complexity: individual work tape; space complexity class $\text{SPACE}(f(n))$, L , PSPACE .
Savitch's Theorem: $\text{ST-PATH} \in \text{SPACE}(\log^2 n)$. (for generalized version, see lec17)
Space Hierarchy Theorem. (proof is similar to THT: diagonalization method)

17 Savitch's Theorem

Savitch's Theorem: $\forall f(n) \geq n : \text{NSPACE}(f(n)) \subseteq \text{SPACE}(f^2(n))$. (proof idea: "middle-first search")
Nondeterminism-based def of NL : $\text{NL} = \text{NSPACE}(\log n)$; $\text{ST-PATH} \in \text{NL}$.
Reading: $\text{NL} \subseteq \text{SPACE}(\log^2 n)$, $\text{NL} \subseteq P$; Can extend Savitch's Theorem to $f(n) \geq \log n$.

18 NL-completeness and Logspace Reductions

$\forall f(n) \geq \log n : \text{NSPACE}(f(n)) \subseteq \text{TIME}(2^{O(f(n))})$, $\text{NSPACE}(f(n)) \subseteq \text{SPACE}(f(n)^2)$.
logspace reduction (to reason about logspace classes, so must be as weak as logspace): closure, transitivity.
(logspace algos are usually insane since they can use crazy time for saving space.)
 ST-PATH is NL -complete.

19 From P-completeness to PSPACE-completeness

P -complete languages: HornSAT , LP , CKT-EVAL ;
Empirically, polytime reduction implies logspace reduction. (also Cook-Levin's)
 TQBF is PSPACE -complete. (proof idea: use quantifiers to reduce size from exp to poly)
(PSPACE 's essence seems to be "games", like TQBF .)

20 The Immerman-Szelepcsényi Theorem

$\text{NPSPACE} = \text{coNPSPACE} = \text{PSPACE}$ by Savitch's. What about scaling down to logspace?
Immerman-Szelepcsényi Theorem: $\overline{\text{ST-PATH}} \in \text{NL}(\text{NL} = \text{coNL})$.
Proof idea: one step each time from s to exploit the locality of "certificates".

21 Randomized Complexity: RP , coRP , and ZPP

new computation resource / power: randomness; can greatly reduce running time with error allowed.
randomized time complexity class (one-sided error): $\text{RTIME}(f(n))$, RP , coRP .
zero-sided error: $\text{ZPTIME}(f(n))$, ZPP .
 $\text{COMPOSITES}, \text{PRIMES} \in \text{RP}$; $\text{PRIMES} \in \text{coRP}$;
 $P \subseteq \text{RP} \subseteq \text{NP}$, $P \subseteq \text{coRP} \subseteq \text{coNP}$; $\text{ZPP} = \text{RP} \cap \text{coRP}$.
Reading: amplification lemma.

22 BPP

two-sided error: $\text{BPTIME}(f(n))$, BPP . (named by "bounded error probabilistic time")
 $\text{BPP} \subseteq \text{PSPACE} \subseteq \text{EXP}$;
 $P = \text{NP} \implies P = \text{BPP}$;
 $\text{BPP} \subseteq P/\text{poly}$. (by amplification, then union bound)
derandomization result: If 3SAT requires $\text{SIZE}(2^{\delta n})$ for some $\delta > 0$, then $P = \text{BPP}$. (worst-case hardness
 \implies strong average-case hardness $\implies \text{PRNG}$)
Reading: Read-Once Branching Programs, $\text{EQ}_{\text{ROBP}} \in \text{BPP}$ (proof idea: arithmetization on \mathbb{F}_p .)

23 The Polynomial Hierarchy

$P = NP \implies NP = \text{coNP}$; quantifier-based def of PH; $\Sigma_i\text{-SAT}$ is $\Sigma_i P$ -complete.

PH “*collapses*” to the i -th level if $\Sigma_i P = \Pi_i P$. (e.g., if $P = NP$ or $NP = \text{coNP}$)

Reading: *Alternating TM*, ATIME , ASPACE , AP , APSPACE , AL .

$\forall f(n) \geq n : \text{ATIME}(f(n)) \subseteq \text{SPACE}(f(n)) \subseteq \text{ATIME}(f^2(n))$; $\forall f(n) \geq \log n : \text{ASPACE}(f(n)) = \text{TIME}(2^{O(f(n))})$.

corollary: $\text{AL} = P$, $\text{AP} = \text{PSPACE}$, $\text{APSPACE} = \text{EXP}$.

24 Oracle TMs & P^{NP}

Oracle TM; $P^{\text{NP}} \subseteq \Sigma_2 P$ (and thus $P^{\text{NP}} = \text{co-}P^{\text{NP}} \subseteq \text{co-}\Sigma_2 P = \Pi_2 P$). In fact P^{NP} is just $\Delta_2 P = \Sigma_2 P \cap \Pi_2 P$.

25 Interactive Proofs: $\text{IP} = \text{PSPACE}$

A glimpse past the 80s from this lec; from mid 60s to the end of 80s (space/time/randomness) in prev.

Interactive Proof System (interaction + randomness): $\text{IP}[k]$, $\text{IP} = \text{IP}[\text{poly}(n)]$.

$\text{NP} \subseteq \text{IP}[1]$; $\text{BPP} \subseteq \text{IP}[0]$; $\text{IP} \subseteq \text{PSPACE}$.

The two-sided error def of IP can be automatically upgraded to one-sided error.

[Shamir '89] $\text{TQBF} \in \text{IP}$ is motivated by [LFKN '89] $\#3\text{SAT} \in' \text{IP}$. (proof idea: arithmetization on \mathbb{F}_p .)

Reading: IP is kind of a probabilistic analog of NP, like the probabilistic analog RP of P.

26 Beyond Worst-Case Analysis

Assume $P \neq \text{NP}$ in next lecs. (Thus $3\text{SAT} \notin P$.)

To solve 3SAT : 1. allow more time; 2. allow errors (correct for $\geq 99\%$ inputs); 3. approx algos (satisfies 90% clauses for every input). Dream is to show hardness for these 3 possibilities, just with $P \neq \text{NP}$.

27 Hardness within P

What is really efficient maybe not P, but $O(n \text{ polylog}(n))$; For these the model matters! (RAM here)

$\text{SETH} \implies \forall \epsilon > 0 : \text{LCS} \notin \text{TIME}(n^{2-\epsilon})$. Similar results for 3SUM , APSP , $k\text{-CLIQUE}$, etc.

fine-grained complexity; care about the exact complexity of reductions, not just P.

some reductions between problems, with assumptions SETH / CNF-SETH .

28 Why is P vs. NP difficult?

History of P vs. NP, & some negative results: HALTS not computable; THT ; both use diagonalization.

Baker-Gill-Solovay's Theorem: $\exists A, B : P^A = \text{NP}^A, P^B \neq \text{NP}^B$. (negative result about *proof tech!*)

Proof idea: $A = \text{TQBF}$, construct B using diagonalization.

Approaches after BGS: In '80s theorists try to prove harder statements; In '88 Håstad shows ckt lower-bound for PARITY ; In '94 Razborov shows limitations of “natural” proof strategy.

Assuming good PRNG exists, \nexists “natural” proof that NP has no poly-size ckts.

29 Additional Remark

$\text{NP} \neq \text{SPACE}(n)$ (hw7-4), core tech: NP is closed under poly-time reduction, but $\text{SPACE}(n)$ is not.

[Sipser-Lautemann] $\text{BPP} \subseteq \Sigma_2 \cap \Pi_2$ (hw11-4), core tech: reduce “randomized amount” to quantifier.