Michael A. Forbes
miforbes@illinois.edu
2023-09-06 1

cs579 Computational Complexity: Lecture 6 (2023-09-07)

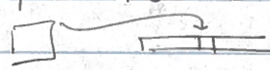
logistics: pset 1 due [late policy]
pset 2 out [2 weeks]

last lecture: CNF-SAT NP-hard [Cook-Levin] [extends to 3SAT w/ more work]
- NTM configs [encode TM snapshot into string] [state, head pos, tape contents] [valid sequence of configs]
- NTM tableau [non-det choices]
- boolean formula for tableau validity [- cell contents - init - accept - transition]
  ↳ via 2x3 windows [head moves ≤ 1 position per step]

today: space complexity

Q: what is a computational resource?
A: [so far] - time - deterministic - non-deterministic
- [Q] ability to solve computational problem [mapping reduction]
- space [det vs nondet]

eg:



≤ t steps ⟹ $\underbrace{\leq t \text{ cells used}}_{\text{space}}$ [used in Cook-Levin] [amount of RAM sometimes limiting in real life]

def: TM M runs in space $s(n): \mathbb{N} \to \mathbb{N}$ if M halts on all $x \in \Sigma^n$ using ≤ s(n) tape cells
NTM N — " — N — " — on all branches

def: $SPACE(s(n)) = \{A : A = L(M), \text{TM } M \text{ runs in space } O(s(n))\}$ [suppress constants] ["right" resolution of study]
$NSPACE(s(n)) = \{$ — " — $L(N)$, NTM N — " — $\}$
$PSPACE = \bigcup_k SPACE(n^k) = SPACE(poly(n))$ is polynomial space
NPSPACE — " — N — " — N — " — non-det — " —

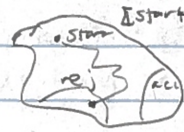
[time vs space?] [time is fourth dimension]

prop: (a) $TIME(t(n)) \subseteq SPACE(t(n))$ [no penalty]
(b) $SPACE(s(n)) \subseteq TIME(2^{O(s(n))})$ [conversion lurs] [not space efficient: clock is $\geq$ ... space constructible]

pf: (a) clear [again: ≤ t steps ⟹ ≤ t cells]
(b) recall: [last lecture] configuration of TM contains [TM vs NTM configs]

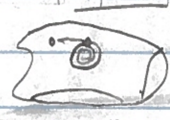
		# [choice?]	[Γ, Q constant]
- tape contents		$\leq \lvert Q\rvert \lvert\Gamma\rvert^{s(n)}$	$\leq 2^{O(s(n))}$
- state		$\leq \lvert Q\rvert$	$\leq O(1)$
- head position		$\leq s(n)$	$\leq 2^{O(s(n))}$
total			$\leq 2^{O(s(n))}$

space of configurations [start depends on x]



[Q: loops?]

clm: TM M on x $\boxed{\text{repeats}}$ config ⟹ M on x does not halt
pf. [by picture] [first repeat ⟹ infinite repeats]



M on x space s(n) ⟹ $\leq C = 2^{O(s(n))}$ distinct configurations
$> C$ steps ⟹ (pigeonhole principle) $\boxed{\text{some}}$ config repeated ⟹ [clm] does not halt — forbidden by def [space]
contrapositive ⟹ $\leq C = 2^{O(s(n))}$ time

∎

Michael A. Forbes
mforbes@illinois.edu
2023-09-07.1

cor: P ⊆ PSPACE

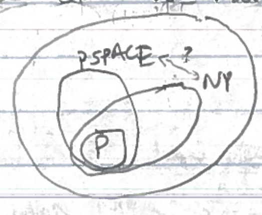
prop: (a) NTIME($t(n)$) ⊆ NSPACE($t(n)$)
(b) NSPACE($s(n)$) ⊆ NTIME($2^{O(s(n))}$)

sketch: [same proof]
(a) $\boxed{\text{each}}$ branch, ≤ t steps. ⇒ ≤ t cells
(b) $\boxed{\text{some}}$ branch, taking $> C = 2^{O(s(n))}$ steps
⇒ $\boxed{\text{some}}$ branch has repeated configs ← may halt! [nondet]
⇒ $\boxed{\text{some}}$ branch [possibly diff] does not halt [violates defn] ∎

cor: NP ⊆ NPSPACE

= [Q]

Q: deterministic space vs. non-deterministic time? [time ≤ space det ≤ nondet look at cross compare]



[NP vs PSPACE]

prop: NP ⊆ PSPACE [direct proof possible] [let's use reductions]

lem: SAT ∈ PSPACE

sketch: [Q] algo: "on input ⟨φ⟩:
[store loop counter] $O(n)$ — for all $x \in \{0,1\}^n$ [# vars]
$O(|\langle\varphi\rangle|)$ — if $\varphi(x) = 1$, accept
— reject"

correctness: clear

complexity: $(x \vee \bar{x}) \wedge \bar{x}$ [naive algo]

[resource of interest]
space: $O(n + |\langle\varphi\rangle|) = O(|\langle\varphi\rangle|)$ ← wlog only use variables $\boxed{\text{appearing}}$ in φ

lem: $\underset{x \mapsto f(x)}{A \leq_p B}$, $\underbrace{B \in \text{PSPACE}}_{M}$ ⇒ $A \in$ PSPACE [mapping reduction]

pf: [instead]
algo: "on input x:
poly time ⇒ poly space — compute $f(x)$
poly $|f(x)|$ space — run M on $f(x)$, accept iff M acc"

correctness: clear

complexity: [space] $|f(x)| \leq \text{poly}|x| \leq \text{poly}(n)$ [as before]
⇒ $\text{poly}(n) + \underbrace{\text{poly}(\text{poly}(n))}_{\text{poly}(n)}$ space ∎

Michael A. Forbes
miforbes@illinois.edu
2023-09-07.2

pf [NP ⊆ PSPACE]:

$A \in NP \Rightarrow A \leq_p SAT \overset{\text{lem}}{\Rightarrow} A \in PSPACE$ ∎

$\uparrow$ lem PSPACE

=

def: [saw] $coNP = \{A : \bar{A} \in NP\}$ $= \Sigma^* \setminus A$

def: $TAUTOLOGY = \{\langle\varphi\rangle : \varphi$ boolean formula, $\forall x\ \varphi(x) = 1\}$

lem: $TAUTOLOGY \leq_p \overline{SAT} \leq_p TAUTOLOGY$ [poly-time equiv] miforbes@illinois.edu

[valid encodings]

sketch: $\langle\varphi\rangle \mapsto \langle\neg\varphi\rangle$ [not TAUT] $\neg\forall x\, \varphi(x) = 1 \equiv \exists x\ \varphi(x) = 0$ [SAT] $(\neg\varphi)(x) = 1$

$\langle\psi\rangle \mapsto \langle\neg\psi\rangle$ [same]

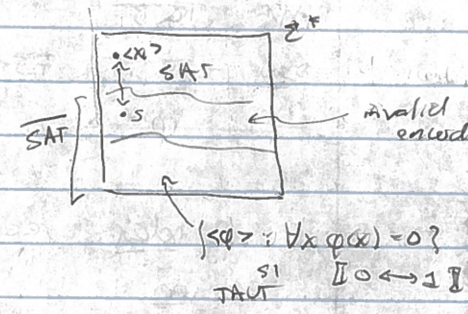
[encoding is efficient]

[and] map invalid encodings appropriately:

invalid encod. $\rightsquigarrow s \mapsto \langle x_1\rangle$ is SAT [one var]

$\notin$ TAUT, $\notin \overline{SAT} \equiv \in SAT$

—— // —— [similar]



lem: L NP-complete iff $\bar{L}$ coNP-complete

pf: – $L \in NP$ iff $\bar{L} \in coNP$

– $A \in NP$ $A \leq_p L$ via f $\bar{A} \in coNP$,

$x \in A$ iff $f(x) \in L$ $\equiv$ $x \in \bar{A}$ iff $f(x) \in \bar{L}$

$\Rightarrow \bar{A} \leq_p \bar{L}$ ∎

cor: TAUTOLOGY is coNP complete

prop: $A \in PSPACE$ iff $\bar{A} \in PSPACE$ [already claimed for P]

sketch: $\Rightarrow$ TM M also: M' = "on input x:

– run M on x

– accept iff M rejects"

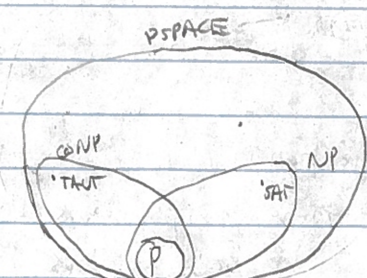
complexity: clear

correctness: clear [only for determinism!]

$\Leftarrow$: —— // —— ∎

cor: $coNP \subseteq PSPACE$

the world:



conj: $P \neq NP$ iff $SAT \notin P$

$NP \neq coNP$

$NP \neq PSPACE \Rightarrow P \neq PSPACE$

iff ?

Q: PSPACE completeness?

def: a true quantified boolean formula is an expression

$\varphi = Q_1 x_1 Q_2 x_2 \cdots Q_n x_n\ \psi$ || that is true prenex

$Q_i \in \{\exists, \forall\}$ ← boolean formula on $x_1, \ldots, x_n$ [no free vars]

Michael A. Forbes
miforbes@illinois.edu
2023-09-07.3

eg: $\varphi = \forall x \exists y \forall z \; (x \vee y) \wedge (\bar{y} \vee z)$ is $\boxed{\text{false}}$

φ true $\Rightarrow$ $\varphi|_{x=0}$ true

$\exists y \forall z \; \underbrace{y \wedge (\bar{y} \vee z)}$ $\overset{\text{distrib}}{=}$ [cook-levin] $(\underbrace{y \wedge \bar{y}}_{=0}) \vee (y \wedge z)$ $= y \wedge z$

$= \exists y \forall z \; y \wedge z \overset{\text{[cases]}}{=} y=0 \; \forall z \; \underbrace{0 \wedge z}_{0}$ is $\boxed{\text{false}}$

$= y=1 \; \forall z \; \underbrace{1 \wedge z}_{z}$ is $\boxed{\text{false}}$ [hard problem]

def: TQBF $= \{ \langle \varphi \rangle : \varphi$ is a true quantified boolean formula $\}$

· TAUT vs TAUTOLOGY

eg: SAT, TAUT $\subseteq$ TQBF

[next lecture]

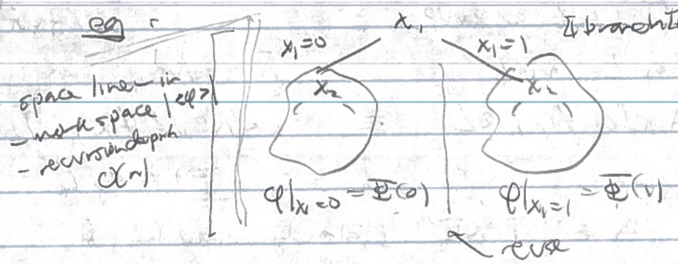
thm: TQBF is PSPACE-complete

prop: TQBF $\in$ PSPACE

pf: $\varphi = Q_1 x_1 \underbrace{Q_2 x_2 \cdots Q_n x_n \psi}_{\Phi(x_1)}$

idea: recurse on variables, $\boxed{\text{reusing}}$ space [space is reusable, time is not] [eg classroom]

eg:



$\varphi = \Phi(0) \wedge \Phi(1)$

algo: "on input $\langle \varphi \rangle$:

[no vars]

- if no quantifiers, eval φ directly, accept iff φ true
- if $\varphi = \exists x_1 \Phi(x_1)$ - recursively solve if $\Phi(0)$ is true
  - — " — 1 — " —
  - accept iff $\boxed{\text{either}}$ is true
- — " — $\forall$ — " —
  - — " —
  - — " —
  - — $\boxed{\text{both}}$ —

correctness: clear

complexity: clm: always halts

pf: start w/ n vars, recursion decreases by 1

0 vars $\Rightarrow$ halt

space: $O(|\langle\varphi\rangle|)$ to store φ

$O(n)$ to store recursion stack - partial results - prefix $x_1 - x_i$

· next lecture: TQBF PSPACE-complete

logistics: pset 1 due, pset 2 out

today: space complexity [def] [non-det] [space vs time]

PSPACE [def] [NP, coNP $\subseteq$ PSPACE]

TQBF [def] [$\in$ PSPACE]

. SC(?) ...