Michael A. Forbes
MiForbes@illinois.edu
2023-09-12.1

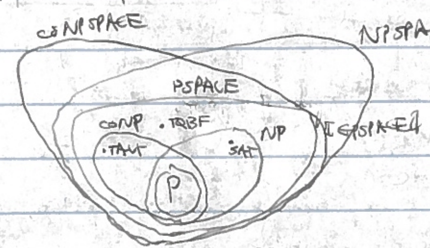
CS579 Computational Complexity: Lecture 7 (2023-09-12)

logistics: - pset1 soln
- pset2 due 09-21

last lecture: - space complexity [[def]] [[non-det]] [[space vs time]]
- PSPACE [[def]]
[[NP ⊆ PSPACE, via NP-completeness of SAT]]
[[coNP ⊆ PSPACE, via swapping acc/rej]]
- TQBF [[def]]
[[⊇ SAT, TAUT, alternating quantifiers]]
[[∈ PSPACE, via reusing space]]

today: P vs NP, vs PSPACE vs NPSPACE

the world:



[[now]] [[conj P ≠ NP ⟹ PSPACE ≠ NPSPACE?]]
Q: PSPACE vs NPSPACE?
vs coNPSPACE? [[conj NP ≠ coNP ⟹ NPSPACE ≠ coNPSPACE?]]
[[new phenomena?]]
[[more basic]]
Q: NPSPACE examples?

eg: convert beer to wine [[one letter at time, in english]] money, bone
[[long board =]]
beer → beer → bear → sear → sent → went → want → wane → wine [[word ladder]]
→ bent →
[[shape ladder]]

$\mathrm{LADDER}_{DFA} = \{\langle M,s,t\rangle : M$ DFA [[recall]] [[def non-cancel]] $L(M)$ contains ladder $s \rightsquigarrow t\}$ 2021

$x_0 \mapsto x_1 \mapsto \cdots \mapsto x_m$
$x_0 = s$, $x_m = t$
- $x_i \in L(M)$
- $|x_i| = |x_{i+1}|$
- $\Delta(x_i, x_{i+1}) = 1$ [[# changed chars]]

prop: $L_1 \in$ NPSPACE

pf: algo: "on input $\langle M,s,t\rangle$
- check if $s,t \in L(M)$, $|s|=|t|$, reject if not
- $x_0 := s$, $n := |s|$
- for $i = 0, 1, \ldots$ ... $|\Sigma|^n$
- [[non-det]] guess $x_{i+1} \in \Sigma^n$ [[Σ = DFA alphabet]] [[same length]]
- if $\Delta(x_i, x_{i+1}) \neq 1$, or $x_{i+1} \notin L(M)$, reject [[run DFA]]
- if $x_{i+1} = t$ accept"

correctness: clear: ladder ⟹ accept

complexity - space: only store M, s, t, x_i, x_{i+1} [[main resource]] ← $O(n \lg|\Sigma| + |\langle M\rangle|)$ ← non constant
forget x_j, $j<i$ [[not needed]]

[Q] halt: does ~~not~~ halt, as is [[part of def]]

clm: $s \rightsquigarrow t$ ladder exists iff — " — of length $\le |\Sigma|^n$
[[prune loops]]
sketch: $s \to t \mapsto s \to t$ simple path via pigeonhole principle [[space vs time]]

Michael A. Forbes
miforbes@illinois.edu
2023-09-12.2

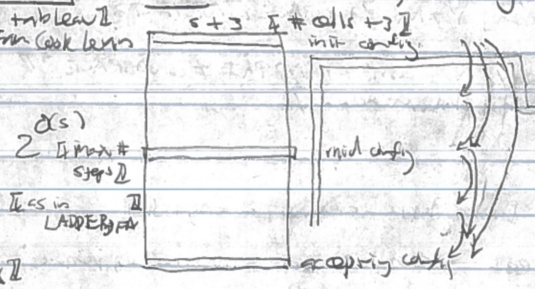
idea: add a clock
$\Rightarrow$ [$i \le |\Sigma|^n$]   correctness: accept $\Leftarrow$ ladder ~~ladder~~ $\exists$ ladder, length $\le |\Sigma|^n \Rightarrow$ accept
complexity: ~~space~~ store $i \le |\Sigma|^n \in O(n \lg|\Sigma|)$
halt: yes

rmk: - same proof shows $\mathrm{SPACE}_{\infty}(s(n)) \subseteq \mathrm{TIME}(2^{O(s(n))})$ [ ]
TM may reject by $\boxed{\text{not}}$ halting [vs last lecture] [add clock]
- ladders may be $\boxed{\text{long}}$, not clearly NP algo

Q: PSPACE algo? [PSPACE vs NPSPACE?]

thm [Savitch]: $s(n): \mathbb{N} \to \mathbb{N}$ w/ - $s(n) \ge n$ [space to store input] - $s(n)$ weakly space constructible ["nice" space bound] [$1^n \to 1^{s(n)}$ in $O(s)$ space]
then $\mathrm{NPSPACE}(s) \subseteq \mathrm{SPACE}(s^2)$ [part 1] [simplifies proof]
rmk: - not needed [full proof in book]
cor: $\mathrm{NPSPACE} = \bigcup_c \mathrm{NPSPACE}(n^c) \subseteq \bigcup_d \mathrm{SPACE}(n^d) = \mathrm{PSPACE}$ [reindex]
$\subseteq \mathrm{SPACE}(n^{2c})$ [surprising!] [space so powerful non-det adds little $\Rightarrow$ no clear P vs NP implications]

pf [Savitch]: idea: recursion, reusing space [same as TQBF $\in$ PSPACE]



[details]
lang A decided by NTM N in space $s(n)$
want: TM M accepting A in space $O(s^2)$ [recall]
def: C_1, C_2 space s config of N. $C_1 \to_t C_2$ if N starting on C_1 can reach C_2 in $\le t$ $\boxed{\text{non-det}}$ steps   [Q = init $\to_{2^{O(s)}}$ accept]
reachable vs not needed?
clm: wlog [convenience] N has unique space s accepting config $\#q_{acc}\underbrace{\sqcup \cdots \sqcup}_{s}\#$
sketch-idea: once you want to accept, erase the tape
N' = "simulate N, if it accepts - overwrite first s cells w/ $\sqcup$
- move head to start of tape
- accept
else reject"
via space constructibility
need 1^n
correctness: clear
complexity: $s + O(s(n))$ space [algo] [space construct.]
make more critical/principled
prop: [recall] N acc x iff $C_{init_x} \to_t$ [unique] w/ $t \le 2^{O(s(n))}$ [depends on x] [LADDER_DFA proof]

Michael A. Forbes
miforbes@illinois.edu
2023-09-12.3

algo: "on input ⟨C_{start}, C_{end}, t⟩ [Q: $C_{start} \to_t C_{end}$]
- if $t=1$, accept iff
  - $C_{start} = C_{end}$ [0 step]
  - $C_{start} \to C_{end}$ in 1 transition of N [NTM] [2x3 windows from Cook-Levin]
- else, - forall C_{mid} [requires knowing s]
  - test $C_{start} \to_{t/2} C_{mid}$ [recursion] [rounding]
  - test $C_{mid} \to_{t/2} C_{end}$
  - if $\boxed{\text{both}}$ possible, accept

correctness:
- accept ⇒ exist tableau
- exist tableau ⇒ $\boxed{\text{same}}$ C_{mid} works ⇒ accept (reject)

complexity: recursion depth $D(t) \le 1 + D(t/2) \le \cdots \le O(\lg t)$ [rounding]

space: $S(t) \le \underbrace{O(s(n))}_{\text{recursion stack, } C_{mid}} + \underbrace{S(t/2)}_{}$ ⇒ halts

recursion calls
- $C_{start} \to C_{mid}$
- $C_{mid} \to C_{end}$

$\boxed{\text{reuse}}$ space

$\le O(\underbrace{D(t)}_{O(\lg t) \le O(s) \text{ as } t \le 2^{O(s)}} \cdot S(t))$

$= O(s^2)$ [QED] ∎

Q: P vs PSPACE? [evidence?] [understand via completeness]

def: B PSPACE-complete iff [≈ NP-completeness]
- $B \in$ PSPACE
- $\forall A \in$ PSPACE, $A \le_p B$ ← poly $\boxed{\text{time}}$ reduction

prop: B PSPACE-complete, $B \in P$ iff $P =$ PSPACE [as before]

thm: TQBF $= \{\langle\varphi\rangle : \varphi$ is a true quantified boolean formula$\}$ [recall] is PSPACE-complete.

quantified boolean formula $= Q_1 x_1 \cdots Q_n x_n\ \psi(x_1, \ldots, x_n)$, $Q_i \in \{\exists, \forall\}$

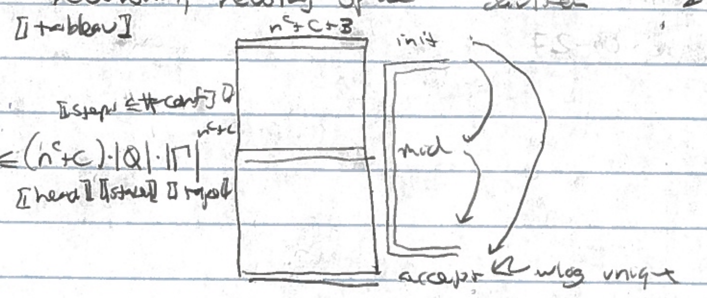
pf: $\in$ PSPACE: last lecture

PSPACE-hard: $A \in$ PSPACE decided by $n^c + c$ space TM [technical lemma]

want: $A \le_p$ TQBF

$x \mapsto \langle\varphi_x\rangle =$ "M acc x" [like Cook-Levin]

idea: recursion, reusing space [≈ TQBF ∈ PSPACE, Savitch]



$\varphi_{C_{start}, C_{end}, t} \approx$ "M starting at C_{start} yields C_{end} in $\le t$ steps"

attempt: $t = 1$: $\underbrace{(C_{start} = C_{mid})}_{\text{convert to boolean logic}} \vee \underbrace{\text{"}C_{start} \to_1 C_{end}\text{"}}_{\varphi_{\text{transition}} \text{ from Cook-Levin}}$

Michael A. Forbes
miforbes@illinois.edu
2023-05-12.4

$t > 1$: $\exists c_{mid}$ $(\varphi_{c_{start}, c_{mid}, t/2} \wedge \varphi_{c_{mid}, c_{end}, t/2})$ [[and]]
$O(n^c)$ boolean vars

[[clear]]

correctness: clm: M acc x iff $\varphi_{c_{init_x}, c_{acc}, 2^{O(n^c)}} \in$ TQBF [[like Savitch]]

complexity [[Q]]: $|\varphi_t| \approx 2 \cdot |\varphi_{t/2}| \approx t = 2^{O(n^c)}$

$\Rightarrow$ consequently φ_t $\boxed{\text{not}}$ polytime

idea: use '$\forall$' to simulate

$\varphi_{c_{start}, c_{end}, t}$ = $t = 1$: as before

$t > 1$: $\exists c_{mid}$ $\underbrace{\forall (c_1, c_2) \in \{(c_{start}, c_{mid}), (c_{mid}, c_{end})\}}_{\text{[[convert to } \exists, \forall \text{]]}} \varphi_{c_1, c_2, t/2}$

$\forall (c_1, c_2) ((c_1, c_2) = (c_{start}, c_{mid}) \vee (c_1, c_2) = (c_{mid}, c_{end}))$

$\rightarrow \varphi_{c_1, c_2, t/2}$

w/ "$a \to b$" $\equiv \neg a \vee b$

"$a \equiv b$" $= (a \to b) \wedge (b \to a)$

correctness: clear [[like Savitch]]

complexity: recursion depth: $D(t) \le 1 + D(t/2) \le O(\lg t)$

$S(i)$ output size: $S'(t) \le \underbrace{O(n^c)}_{\text{constraints on } c_1, c_2} + \underbrace{S'(t/2)}_{\text{recursion}}$

$\le O(D(t) \cdot n^c) \le \cdots \le O(n^{2c})$ [[Savitch]]

time: $\le$ poly ( ) [[structurally equivalent]] similarly proof

today: LADDER$_{DFA}$ $\in$ NPSPACE [[closer to host TM]] [[ex]]

NSPACE(s) $\subseteq$ SPACE(s^2) [[Savitch]] [[start $\to$ mid $\to$ end, $\boxed{\text{reuse}}$ space]]

TQBF PSPACE-complete [[NPSPACE = PSPACE, vs P $\neq$ NP]]

[[Savitch-version of Cook-Levin]]

next lecture: space complexity beyond TQBF

logistics: pset 1 soln

pset 2 due 05-27

• thenvald