Intro to Theory of Computation

CS 332

LECTURE 25

Last time

- Cook-Levin Theorem
- Examples of NP-complete languages

Today

- Space complexity
- The class PSPACE

Sofya Raskhodnikova

Question

1. $3SAT \leq_p A_{TM}$
2. A_{TM} is NP-complete

A. (1) and (2) are both true

B. (1) and (2) are both false

C. (1) is true and (2) is false

D. (2) is true and (1) is false

E. At least one of (1) and (2) is an open question

Space analysis

If M is a TM and $f: \mathbb{N} \to \mathbb{N}$ then

"M runs in space $f(n)$" means

for **every** input $w \in \Sigma^*$ of length n,

M on w uses at most $f(n)$ tape cells.

- If M is a nondterministic TM that halts on all inputs then $f(n)$ is the maximum number of cells M uses on any input of length n.

Space complexity classes

SPACE($f(n)$) is a class of languages.

$A \in$ **SPACE($f(n)$)** means that

some 1-tape TM M

that runs in space O($f(n)$) decides A.

Prove: SAT $\in$ SPACE(n)

M = `` On input $\langle \phi \rangle$, where ϕ is a Boolean formula, with variables $x_1, \dots, x_\ell$:

1. For each truth assignment to $x_1, \dots, x_\ell$
2. Evaluate ϕ on that truth assignment.
3. Accept if ϕ ever evaluates to 1. O.w. reject.”

- If n is the input length, M uses space $O(n)$.

Space complexity classes

NSPACE($f(n)$) is a class of languages.

$A \in$ **NSPACE($f(n)$)** means that some 1-tape *nondeterministic* TM M that runs in space O($f(n)$) decides A.

CS 332

Prove: $\overline{\text{ALL}_{NFA}} \in \text{NSPACE}(n)$

- ALL_{NFA}=$\{\langle M\rangle |\ M$ is an NFA and $L(M) = \Sigma^*\}$

N = `` On input $\langle M\rangle$, where M is an NFA:

1. Place marker on the start state of M.
2. Repeat ▭ times where q is the # of states of M:
3. Nondeterministically select $a \in \Sigma$.
4. Adjust the markers to simulate M reading a.
5. Accept if at any point none of the markers are on an accept state. O.w. reject."

- If n is the input length, N is an NTM that uses space $O(n)$.

Savitch's theorem

Theorem. Let $f(n)$ be a function, where $f(n) \geq n$.
NSPACE($f(n)$)$\subseteq$ SPACE($f^2(n)$).

Savitch's theorem

Theorem. Let $f(n)$ be a function, where $f(n) \geq n$.
NSPACE($f(n)$)$\subseteq$ SPACE($f^2(n)$).

Proof:

- Let N be an NTM deciding a language A in $f(n)$ space.
- We give a deterministic TM M deciding A.
- More general problem:
  - **Given configurations c_1, c_2 of N and integer t, decide whether N can go from c_1 to c_2 in $\leq t$ steps on some nondeterministic path.**
  - **Procedure CANYIELD(c_1, c_2, t)**

Savitch's theorem

Theorem. Let $f(n)$ be a function, where $f(n) \geq n$. NSPACE($f(n)$)$\subseteq$ SPACE($f^2(n)$).

Proof: (the rest of the proof on the board)

CANYIELD = `` On input $\langle \boldsymbol{c_1}, \boldsymbol{c_2}, \boldsymbol{t} \rangle$:

1. **If $t = 1$, accept if $c_1 = c_2$ or c_1 yields c_2 in one transition. O.w. reject.**
2. **If $t > 1$, then $\forall$ configs c_{mid} of N with $\leq f(n)$ cells:**
3. **Run CANYIELD($\langle c_1, c_{mid}, t/2 \rangle$).**
4. **Run CANYIELD($\langle c_{mid}, c_2, t/2 \rangle$).**
5. **If both runs accept, accept.**
6. **Reject."**

The class PSPACE

PSPACE is the class of languages decidable in polynomial space on a ***deterministic*** TM:

$$\boldsymbol{PSPACE} = \bigcup_k SPACE(n^k).$$

- NPSPACE – the same, but for NTMs.
- By Savitch's Theorem,

$$\boldsymbol{PSPACE = NPSPACE}$$

CS 332 Relationships between classes

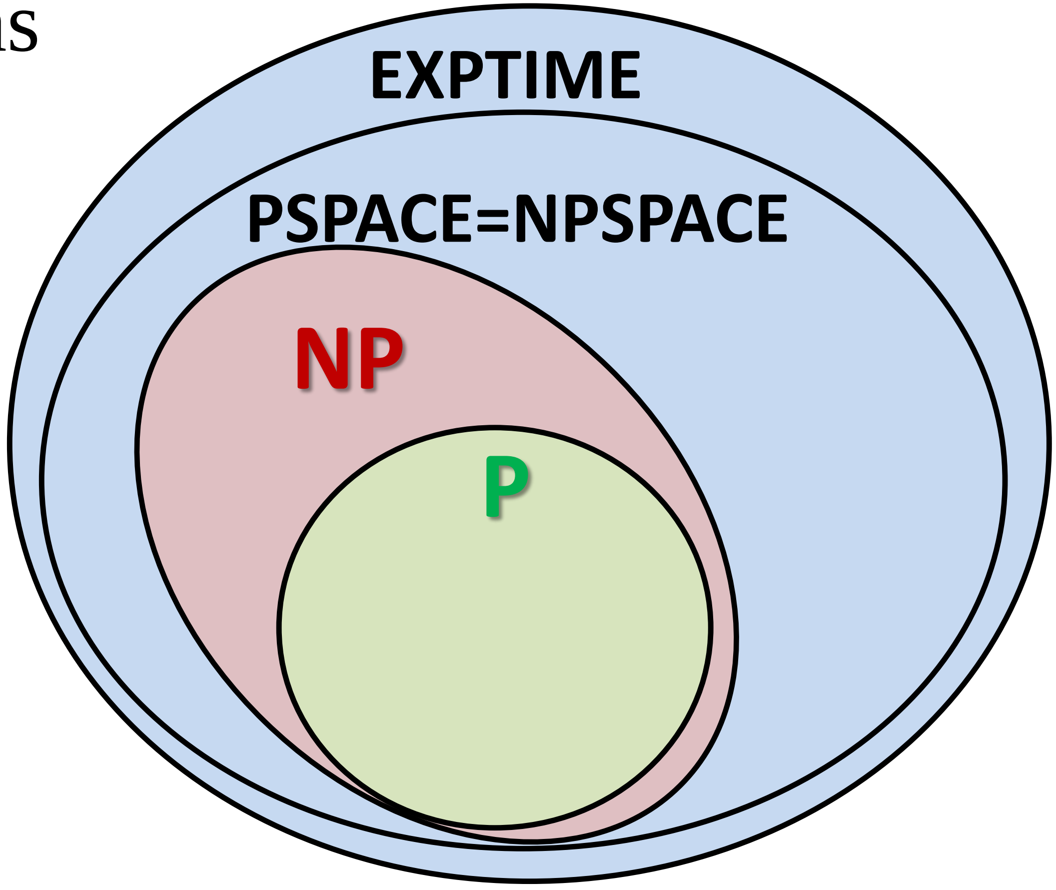
1. P ⊆ NP ⊆ PSPACE ⊆ EXPTIME

   Recall: a TM that runs in space $f(n)$ has $\leq f(n)2^{O(f(n))}$ configurations

2. P≠ EXPTIME

Which containments in **(1)** are proper?

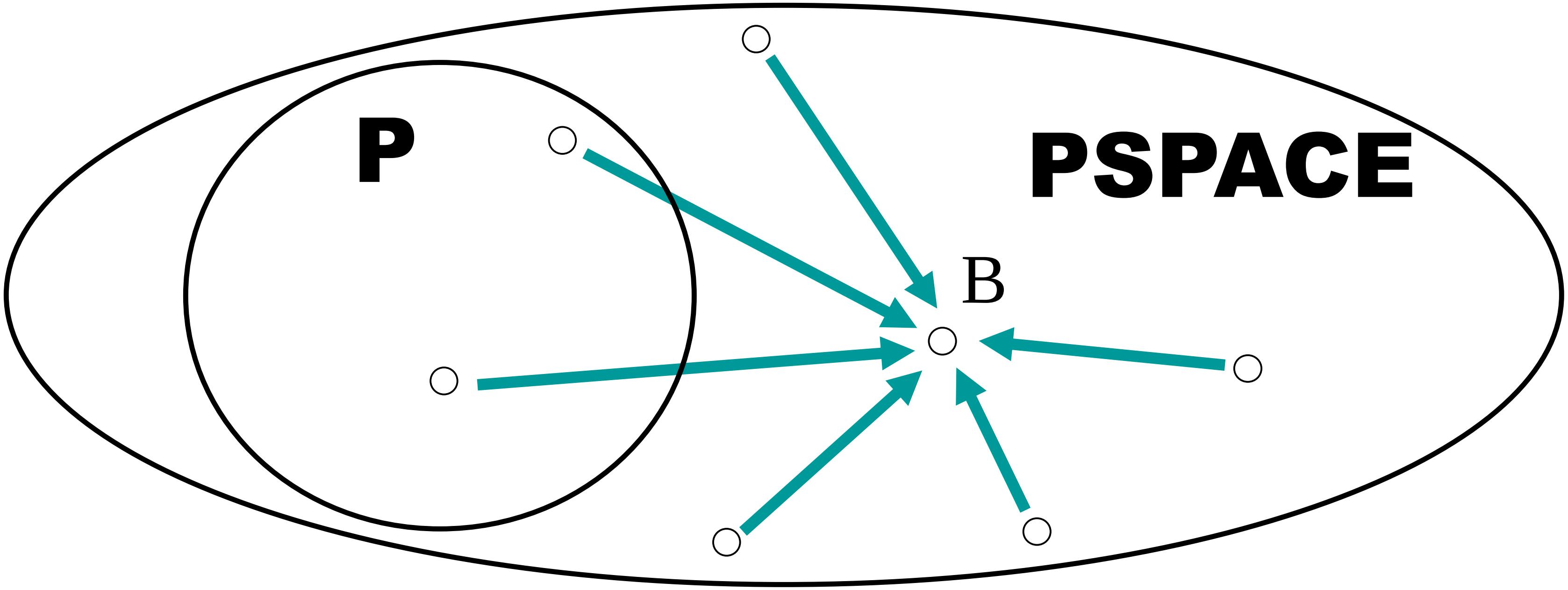
Unknown!



Hardest problems in PSPACE

A language B is **PSPACE-complete** if

1. B∈ **PSPACE**
2. B is **PSPACE-hard**,i.e.,
   every language in PSPACE is poly-time reducible to B.



CS 332

The TQBF problem

- **Boolean variables:** variables that can take on values T/F (or 1/0)
- **Boolean operations:** $\vee$, $\wedge$, and $\neg$
- **Boolean formula:** expression with Boolean variables and ops
- **Quantified Boolean formula:** Boolean formula with quantifiers ($\forall, \exists$)
- **Fully Quantified Boolean formula:** all variables have quantifiers ($\forall, \exists$)

We only consider the form where all quantifiers appear in the beginning.

Ex. $\forall x \exists y [(x \vee y) \wedge (\bar{x} \vee \bar{y})]$ **True**

$\exists y \forall x\ [(x \vee y) \wedge (\bar{x} \vee \bar{y})]$ **False**

- Each fully quantified Boolean formula is either true or false.
- The order of quantifiers matters!

TQBF = $\{\langle \phi \rangle \mid \phi$ is a true fully quantified Boolean formula$\}$

TQBF is PSPACE-complete

1. **TQBF is in PSPACE**
2. **TQBF is PSPACE-hard**

Prove: TQBF $\in$ PSPACE

T = `` On input $\langle \phi \rangle$,
where ϕ is a fully quantified Boolean formula:

1. If ϕ has no quantifiers, it has only constants (and no variables). Evaluate ϕ.
   If true, accept; o.w., reject.
2. If ϕ is of the form $\exists x\, \psi$, recursively call T on ψ with $x = 0$ and then on ψ with $x = 1$.
   If either call accepts, accept; o.w., reject.
3. If ϕ is of the form $\forall x\, \psi$, recursively call T on ψ with $x = 0$ and then on ψ with $x = 1$.
   If both calls accept, accept; o.w., reject."

- If n is the input length, T uses space $O(n)$.

CS 332

Question

If TQBF is in P then it implies that

A. **P = NP**

B. **P = PSPACE**

C. **P = EXPTIME**

D. **(A) and (B) are true**

E. **(A), (B), (C) are true**