Lecture 26

Last time

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

Today

- Space complexity
- The class PSPACE
- Hierarchy theorems

Sofya Raskhodnikova
An encyclopedia of NP-complete problems
``We need an efficient algorithm that constructs a design of iThingy that meets the maximum # of requirements at lowest cost.”

``I thought about it for weeks, but I can’t come up with an efficient algorithm. I guess I’m just too dumb.”

“I can’t find an efficient algorithm, but neither can all these famous people.”
3SAT is an example of a problem that cannot be solved by an algorithm.

A. True
B. False
C. It is an open question
D. None of the above
Question

3SAT is an example of a problem that cannot be solved by a polynomial time algorithm.

A. True
B. False
C. It is an open question
D. None of the above
1. $3\text{SAT} \leq_p A_{TM}$
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 \( f : \mathbb{N} \rightarrow \mathbb{N} \) 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 nondeterministic TM that halts on all inputs then \( f(n) \) is the maximum number of cells M uses on any input of length \( n \).
SPACE($f(n)$) is a class of languages. $A \in \text{SPACE}(f(n))$ means that some 1-tape TM M that runs in space $O(f(n))$ decides A.
Prove: SAT ∈ SPACE(\(n\))

M = ```
On input \(\langle \phi \rangle\), where \(\phi\) is a Boolean formula, with variables \(x_1, \ldots, x_\ell\):

1. For each truth assignment to \(x_1, \ldots, x_\ell\):
2. Evaluate \(\phi\) on that truth assignment.
3. Accept if \(\phi\) ever evaluates to 1. O.w. reject.”
```

- If \(n\) is the input length, M uses space \(O(n)\).
NSPACE\( (f(n)) \) is a class of languages.\\n
\( A \in \text{NSPACE}(f(n)) \) means that some 1-tape nondeterministic TM M that runs in space \( O(f(n)) \) decides A.
Prove: $\text{ALL}_{\text{NFA}} \in \text{NSPACE}(n)$

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

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

1. Place marker on the start state of $M$.
2. Repeat $q$ 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, $N$ uses nondeterministic space $O(n)$. 
Savitch’s theorem

**Theorem.** Let $f(n)$ be a function, where $f(n) \geq n$. 
$\text{NSPACE}(f(n)) \subseteq \text{SPACE}(f^2(n))$. 
The class PSPACE

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

\[ PSPACE = \bigcup_k \text{SPACE}(n^k). \]

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

\[ PSPACE = NPSPACE \]
1. \( P \subseteq NP \subseteq PSPACE \subseteq EXPTIME \)
   
   **Recall:** a TM that runs in space \( f(n) \) has \( \leq f(n)2^{O(f(n))} \) configurations

2. \( P \neq EXPTIME \)
   
   Which containments in (1) are proper? **Unknown!**
Motivating question

Is there a decidable language which is provably not in P?
A function $f: \mathbb{N} \rightarrow \mathbb{N}$ is **space constructible** if the function that maps the string $1^n$ to the binary representation of $f(n)$ is computable in space $O(f(n))$.

- Fractional functions are rounded down.
- **Examples:** $\log_2 n, \sqrt{n}, n \log_2 n, n^2, 2^n$, all commonly occurring functions that are $\Omega(\log n)$
Hierarchy Theorems

**Corollary 1:** For all $\alpha, \beta \geq 0$, where $\alpha < \beta$, \[ \text{SPACE}(n^\alpha) \text{ is a strict subset of } \text{SPACE}(n^\beta) . \]

**Corollary 2:** PSPACE is a strict subset of EXPSPACE, where \[ \text{EXPSPACE} = \bigcup_k \text{SPACE}(2^{n^k}) \]
A function \( f : \mathbb{N} \rightarrow \mathbb{N} \) is **time constructible** if the function that maps the string \( 1^n \) to the binary representation of \( f(n) \) is computable in time \( O(f(n)) \).

- Fractional functions are rounded down.
- **Examples:** \( n \log_2 n, n^2, 2^n \), all commonly occurring functions that are \( \Omega(n \log n) \).
Hierarchy Theorems

Time Hierarchy Theorem
For any time constructible function \( f \),
   a language exists that is decidable in \( O(f(n)) \) time,
   but not in \( o\left(\frac{f(n)}{\log f(t)}\right) \) time.

- **Corollary 1:** For all \( \alpha, \beta \geq 0 \), where \( \alpha < \beta \),
  \( \text{TIME}(n^\alpha) \) is a strict subset of \( \text{TIME}(n^\beta) \).
- **Corollary 2:** \( P \) is a strict subset of \( \text{EXPTIME} \)
A language \( B \) is \textbf{PSPACE-complete} if

1. \( B \in \text{PSPACE} \)
2. \( B \) is \textbf{PSPACE-hard}, i.e., every language in PSPACE is poly-time reducible to \( B \).
A language $B$ is EXPSPACE-complete if

1. $B \in \text{EXPSPACE}$
2. $B$ is EXPSPACE-hard, i.e., every language in EXPSPACE is poly-time reducible to $B$. 
The $EQ_{REX}\uparrow$ problem

- Regular operations: $\cap$, $\circ$, $\ast$
- Regular expressions: use regular operations, $\emptyset$, $\varepsilon$, $\Sigma$, members of $\Sigma$
- We can test equivalence of two regular expressions in poly space (problem in the book).
- Exponentiation operation: $R^k = R \uparrow k = R \circ \cdots \circ R$ ($k$ times)

$EQ_{REX}\uparrow = \{ \langle Q, R \rangle \mid Q \text{ and } R \text{ are equivalent regular expressions with exponentiation} \}$

$EQ_{REX}\uparrow$ is EXPSPACE-complete.
Recognizable

Decidable

EXPSPACE

EXPTIME

PSPACE = NPSPACE

P

CFL

regular

NP

coNP

EQ_{REX}↑

Generalized Chess

Generalized Geography