Intro to Theory of Computation

CS 464

LECTURE 28

Last time

- Examples of NP-complete problems

Today

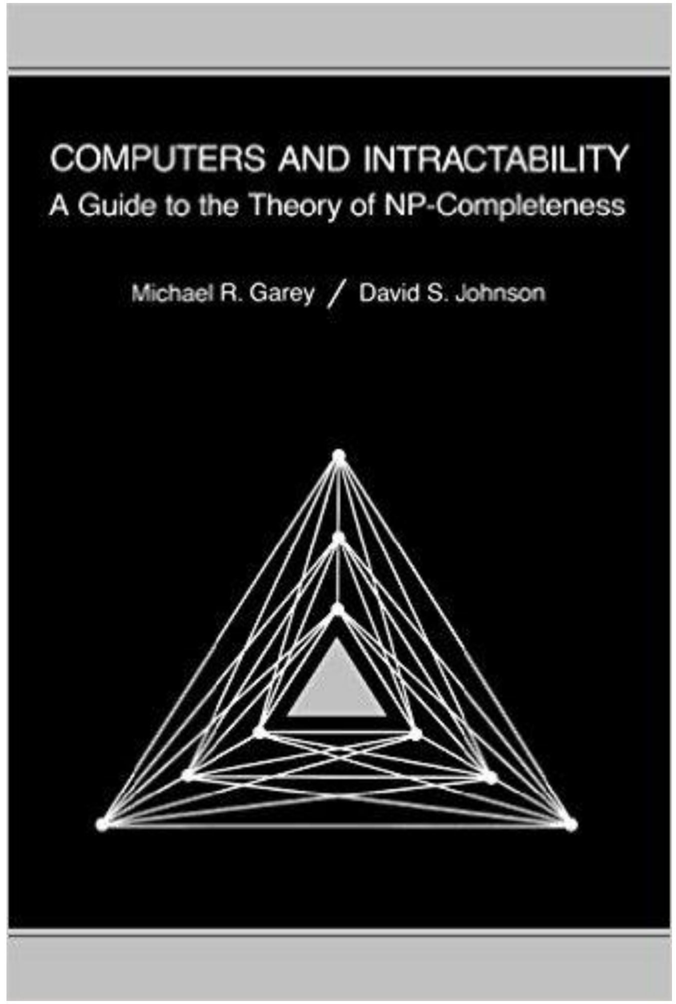
- Space complexity

Homework 11 due
Homework 12 out
Extra credit programming assignment

Sofya Raskhodnikova

CS 464

An encyclopedia of NP-complete problems



CS 464

From Garey and Jonhson

``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.”

I-clicker question (frequency: AC)

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

CS 464

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 ∈ SPACE(n)

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

1. **For each truth assignment to $x_1, \ldots, x_\ell$**
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.

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

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

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 uses nondeterministic space $O(n)$.

Savitch's theorem

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

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)**

CS 464

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 $\langle c_1, c_2, t\rangle$:

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:

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

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

$$\boldsymbol{PSPACE = NPSPACE}$$

- P ⊆ NP ⊆ PSPACE = NPSPACE ⊆ EXPTIME