Michael Forbes
miforbes@illinois.edu

2019-01-24.4 → 2019-01-24.1
2019-01-24.2 cs579

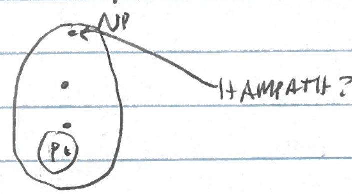
cs 579 Computational Complexity: Lecture 4 19?

<u>admin</u>: ps 1 due 01-31

<u>last time</u>: nondeterminism
NP
HAMPATH

<u>today</u>: reducibility
completeness
SAT
[everything you need for ps 1]

<u>Q</u>: evidence HAMPATH is hard?
[vs eulerian path]



thm: HAMPATH $\in$ P iff P = NP
[it is "one of hardest" problems in NP]
[captures all of NP = nondeterminism in a single computational problem]
[how to prove? infinite # of probs in NP]

<u>idea</u>: use <u>reductions</u>

ie reduce solving problem A to solving problem B
[view solving B as computational resource, ask how hard to solve A]
[we do this in everyday life - programming libraries]

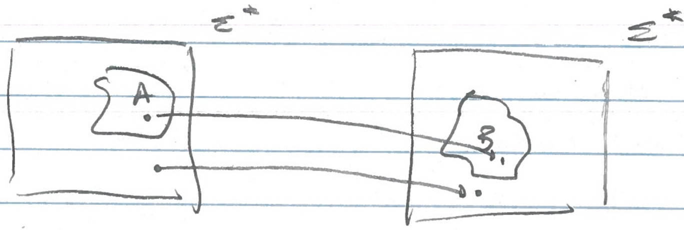
<u>def</u>: a TM M <u>with output</u> is a [deterministic] TM w/ distinguished states - q_{start}, - q_{halt}
[no accept/reject]
it computes by - initialization [as before]
- iterating transition function [as before]
- if ever reach q_{halt}: - halt
- output tape contents
↳ longest prefix of tape in Σ^*
ie $w \in \Gamma \setminus \Sigma$

the <u>time</u> of M on input x is # transitions until halting
a function $f: \Sigma^* \to \Sigma^*$ is <u>poly time computable</u> if some TM M has $\forall w \in \Sigma^*$ outputs $f(w)$, in time $t(|w|)$ ← some polynomial

<u>def</u>: $A \subseteq \Sigma^*$ is poly time mapping (/many-one) reducible to $B \subseteq \Sigma^*$
denoted $A \leq_p B$ [p for polynomial], if $f: \Sigma^* \to \Sigma^*$ poly time computable
w/ $\forall w$, $w \in A$ iff $f(w) \in B$

Michael Forbes
mforbes@illinois.edu
2019-01-24.2 $\leftrightarrows$ 2019-01-24.1
2019-01-24.3

eg



Rmk: - f is not necessarily injective, nor surjective
- other types of reductions exist [more general: - ask many B-questions
- this is "right" notion for us]

Prop. $A \leq_p B$, $B \in P \Rightarrow A \in P$

Pf. [ask]
polytime reduction $f: \Sigma^* \to \Sigma^*$ w/
polytime TM M for B

N = "on input x:
$\text{poly}(|x|)$ 1) Compute $f(x)$
$\text{poly}(|f(x)|) \leq \text{poly}|x|$ 2) Run M on $f(x)$ and accept iff M does."

correctness: $x \in A \Rightarrow f(x) \in B \Rightarrow M \text{ acc} \Rightarrow N \text{ acc}$
$\quad \notin \qquad\qquad\qquad\qquad$ rej

runtime: [composition of polynomials is polynomial.] ∎

Cor: $A \leq_p B$. $A \notin P \Rightarrow B \notin P$.

Prop: $A \leq_p B$. $B \in NP \Rightarrow A \in NP$

Sketch. $x \in A \Rightarrow f(x) \in B \Rightarrow$ some branch of M acc $\Rightarrow$ some branch N acc $\Rightarrow$ N acc
$\quad \notin \qquad \notin \qquad$ no " $\qquad$ no " $\qquad \Rightarrow N$ rej ∎

def. language B is NP-complete if
- $B \in NP$ [remember this for psets]
- $\forall A \in NP$, $A \leq_p B$.

Cor: B NP-complete. $B \in P$ iff $P = NP$. [evidence of computational hardness]

Cor. B NP-complete. $C \in NP$, $B \leq_p C \Rightarrow C$ NP-complete

Pf: - $C \in NP$
- any $A \in NP$. $A \overset{f}{\leq_p} B \overset{g}{\leq_p} C \Rightarrow A \leq_p C$
$x \in A \Rightarrow f(x) \in B \Rightarrow g(f(x)) \in C$ [runtimes compose, still polynomial]
$\quad \notin \qquad\quad \notin \qquad\quad \notin$

[are there NP-complete problems?]

def. a boolean formula φ is an expression involving boolean variables [0,1] and AND, OR, NOT operations. eg $\varphi = (\underset{1}{\bar{\overset{0}{x}}} \wedge \underset{0}{y}) \vee \overset{1}{\overline{(\underset{0}{x} \wedge \underset{0}{\bar{\overset{0}{z}}})}} = 1$

φ is satisfiable if there is a $\{0,1\}$ assignment to the variables that makes φ evaluate to 1, eg $x = 0$, $y = 0$, $z = 1$

$SAT = \{\langle \varphi \rangle : \varphi \text{ is satisfiable boolean formula}\}$

Prop: $SAT \in NP$.

Michael Forbes
miforbes@illinois.edu

<u>def</u> a boolean formula φ is in <u>conjunctive normal form</u> (CNF) if

$$\varphi = C_1 \wedge \cdots \wedge C_m$$

each C_i is a <u>clause</u> $C = (\ell_1 \vee \cdots \vee \ell_k)$

$\nwarrow$ literals $\in \{ x_i, \bar{x}_i \}$

φ is a k-CNF formula if it is in CNF form w/ each clause having $\leq k$ literals

$3SAT = \{ \langle \varphi \rangle : \varphi$ is a satisfiable boolean formula in $\underbrace{3CNF}_{\text{easily checkable}} \}$ [$\in NP$]

[independently works during cold war]

thm [Cook 71, Levin 73] There exist NP-complete problems [nontrivial, but direct proof on ps1] and further 3SAT is NP-complete. [hence SAT is also]

= Questions

Prop: $CLIQUE = \{ \langle G, k \rangle :$ G undirected graph with $\underbrace{\text{clique}}$ of size $\geq k \}$ is NP-complete

$S \subseteq V$ w/ $(i,j) \in E$ $\forall i \neq j \in E$

<u>Pf</u>: <u>$\in NP$</u>: easy

<u>NP-hard</u>: will give reduction $3SAT \leq_p CLIQUE$.

<u>eg</u> $\varphi = (x_1 \vee x_2 \vee \bar{x}_3) \wedge (x_1 \vee \bar{x}_2 \vee \bar{x}_4) \wedge \cdots \wedge (\cdots)$

$f \downarrow$

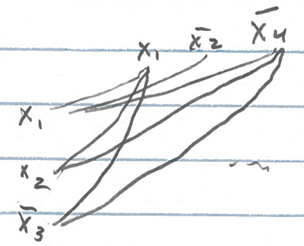
$V = \overbrace{\overset{x_1}{\bullet} \quad \overset{x_2}{\bullet} \quad \overset{\bar{x}_3}{\bullet}} \quad \overbrace{\overset{x_1}{\bullet} \quad \overset{\bar{x}_2}{\bullet} \quad \overset{\bar{x}_4}{\bullet}} \quad \cdots \quad \overbrace{\bullet \ \bullet \ \bullet}$

k = # clauses

$V = \{$ literals appearing in clauses, w/ duplicates $\} \Rightarrow |V| \leq 3k$

E = all edges except
- self loops
- between literals in same clause
- between literals that are negations of each other

<u>eg</u>



<u>Clm</u>: φ satisfiable iff $\langle G, k \rangle \in CLIQUE$

<u>Pf</u> $\underline{\varphi \text{ satisfiable}}$, say $\varphi(\alpha) = 1$ $\alpha \in \{0,1\}^{n = \#\text{vars}}$

each clause $C = (\ell \vee \ell' \vee \ell'') \Rightarrow \geq$ one literal is true

$S := \{$ set of literals made true by $\alpha \}$ [w/ duplicates]

$\hookrightarrow |S \cap C_i| \geq 1$ all i

pick $T \subseteq S$ w/ $|S \cap C_i| = 1$ all i

Michael Forbes
mforbes@illinois.edu
2019-01-24.4 ⇆ 2019-01-24.3 → 2019-01-29.1
CS579

Clm: $T \subseteq V$ is a k-clique
Pf - $|T| = k$ by construction
- $\ell \neq \ell' \in T$: - not a self loop
  - not in same clause
  - both made true by α ⇒ not negations of each other,] ⇒ $(\ell, \ell') \in E$ ☒

$\langle G, k\rangle \in$ CLIQUE: want to extract α st $\varphi(\alpha) = 1$

say S is k-clique
note $\underbrace{|S \cap C_i| \leq 1}$ [no edges between literals in same clause]
⇒ $|S \cap C_i| = 1$ ⇒ S picks exactly 1 literal per clause
↳ cannot pick both x and $\bar{x}$ [no edge]

define $\alpha(x) = \begin{cases} 1 & x \in S \\ 0 & \bar{x} \in S \\ 0 & x, \bar{x} \notin S \end{cases}$

Clm: $\varphi(\alpha) = 1$
Pf. - α is well defined
- α makes all literals in S true
- S has 1 literal per clause ⇒ all clauses true ⇒ φ true ☒

hence φ satisfiable iff $\langle G, k\rangle \in$ CLIQUE [so reduction works]

Rmk: 3SAT $\leq_p$ HAMPATH in textbook [so hampath NP-complete]

next time: Cook Levin Thm: 3SAT is NP-complete