# Undergraduate Complexity Theory
# Lecture 13: Search-to-Decision, Padding, Dichotomy Theorems

Marcythm

July 16, 2022

## 1 Lecture Notes

What we are going to do:

1. Deciding $\mathsf{SAT}$ is hard. Is searching even harder?
2. Maybe $\mathsf{SAT} \notin \mathsf{P}$, how fast can we solve it?
3. $\mathsf{LIS} \in \mathsf{P}$, even $\in \mathsf{TIME}(n^2)$. Is it $\in \mathsf{TIME}(n^{1.9})$?
4. Other resources (e.g. space, random interaction)?
5. $\mathsf{2SAT} \in \mathsf{P}$, $\mathsf{3SAT}$ $\mathsf{NP}$-complete. Why?
6. Is every $L \in \mathsf{NP}$ either $\in \mathsf{P}$ or $\mathsf{NP}$-complete?
7. Maybe $\mathsf{SAT}$ "almost in" $\mathsf{P}$?

Suppose $\mathsf{P} = \mathsf{NP}$, then $\mathsf{CIRCUIT\text{-}SAT} \in \mathsf{P}$, exists poly-time algo $M_{\mathsf{CSAT}}$ s.t.

$$M_{\mathsf{CSAT}}(C) = \begin{cases} \text{yes}, & \text{if } C \text{ is satisfiable} \\ \text{no}, & \text{if } C \text{ is not satisfiable} \end{cases}$$

then we can decide a satisfying assignment bit-by-bit.

For $\mathsf{3COL}$, the subinstance to find solution is $\mathsf{3COL{+}UNARY}$ in homework 6.

$\mathsf{CSAT}$ is "downward" self-reducible.

**Theorem 1.1.** *Suppose $L \in \mathsf{NP}$, with verifier $V(x, y)$. Assume $\mathsf{P} = \mathsf{NP}$, then exists poly-time algo $S$ s.t. $\forall x \in L, V(x, S(x))$ accepts.*

*Proof.* Use the idea of Cook-Levin Theorem. Given $x \in L$, construct a circuit $C_x(y)$ that does the same as $V(x, y)$. Then $S(x) = C(x, y)$ is a solution. □

A complexity theory "trick": padding

**Theorem 1.2.** $\mathsf{P} = \mathsf{NP} \implies \mathsf{EXP} = \mathsf{NEXP}$ *(contrapositive:* $\mathsf{EXP} \neq \mathsf{NEXP} \implies \mathsf{P} \neq \mathsf{NP}$*)*

*Proof.* Asuume $\mathsf{P} = \mathsf{NP}$, need to show $\mathsf{NEXP} \subseteq \mathsf{EXP}$. Let $L \in \mathsf{NEXP}$, say $M$ is a nondet TM deciding $L$ in time $2^{n^k}$. Let $L_{pad} = \{\langle x, 1^{2^{|x|^k}} \rangle : x \in L\}$.

**Claim 1.3.** $L_{pad} \in \mathsf{NP}$.

*Proof.* Define $M'$ a nondet TM. Given input $y$, it first check if $y = \langle x, 1^{2^{|x|^k}} \rangle$ in time about $O(|y|)$, then throws away 1s, gets $x$, and simulate nondet $M(x)$ in time $O(2^{n^k}) = O(|y|)$. □

Then $L_{pad} \in \mathsf{P}$. Let $A$ be poly-time algo deciding $L_{pad}$.

**Claim 1.4.** $L \in \mathsf{EXP}$.

*Proof.* Let $A'$ be TM with input $x$, it runs $A(\langle x, 1^{2^{|x|^k}} \rangle)$ in time $O(\text{poly}(2^{|x|^k}) = 2^{O(n^k)}$, i.e. $A'$ is in exponential time, so $L \in \mathsf{EXP}$. $\square$

$\square$

1. World 1: $\mathsf{NP} = \mathsf{P}$ + $\mathsf{NP}$-hard + something else
2. World 2: $\mathsf{NP} = \mathsf{P}$ + $\mathsf{NP}$-hard
3. World 3: $\mathsf{P} = \mathsf{NP}$

**Theorem 1.5** (Schaefer's Dichotomy Theorem '78)**.** *Every boolean CSP is either in* $\mathsf{P}$ *(basically* $\mathsf{2SAT}$*,* $\mathsf{XOR\text{-}SAT}$*,* $\mathsf{Horn\text{-}SAT}$*) or* $\mathsf{NP}$*-complete (everything else).*

**Theorem 1.6** (Bulatov '06)**.** *Same theorem for ternary CSP, i.e.* $|D| = 3$*. e.g.* $\mathsf{3\text{-}COL}$*.*

**Conjecture 1.7** (Dichotomy Conjecture)**.** *Every CSP is either in* $\mathsf{P}$ *or* $\mathsf{NP}$*-complete.*

upcoming: a theorem that shows world 2 is generally impossible.

**Theorem 1.8** (Ladner's Theorem)**.** *Assume* $\mathsf{P} \neq \mathsf{NP}$*, then* $\exists L \in \mathsf{NP}$ *s.t.* $L \notin \mathsf{P}$ *and* $L$ *is not* $\mathsf{NP}$*-complete.*

such $L$s are mostly unnatural. Maybe a natural $L$: $\mathsf{GRAPH\text{-}ISOMORPHISM}$. Fastest algo is $\mathsf{TIME}(n^{\log^{10} n})$ (Babai '16).