CS314 - More NP-Completeness

Note Title 10/30/2013

Announcements

- HW due Friday
- Oral grading next week (hopefully)

nondeterministic polynomial time

P, NP, + co-NP

Consider decision problems: Yes or No.

P: set of decision problems that can be solved in polynomial time.

Ex: is this list sorted?

: is s connected to t in G?

NP: set of problems s.t., if the answer is yes, this can be checked in poly. time.

(So can verify a yes answer.)

NP-Hard

Π is NP-hard $\iff$ If Π can be solved in polynomial time, then P=NP

So if an NP-Hard problem can be solved in polynomial time, then any problem in NP can be solved in polynomial time.

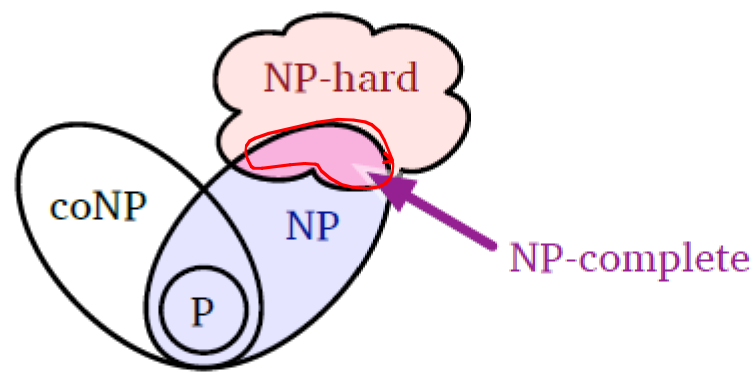
(Paths story ...)

P vs NP

NP-Completeness

A problem is NP-Complete if it is both:

- in NP
- NP-Hard

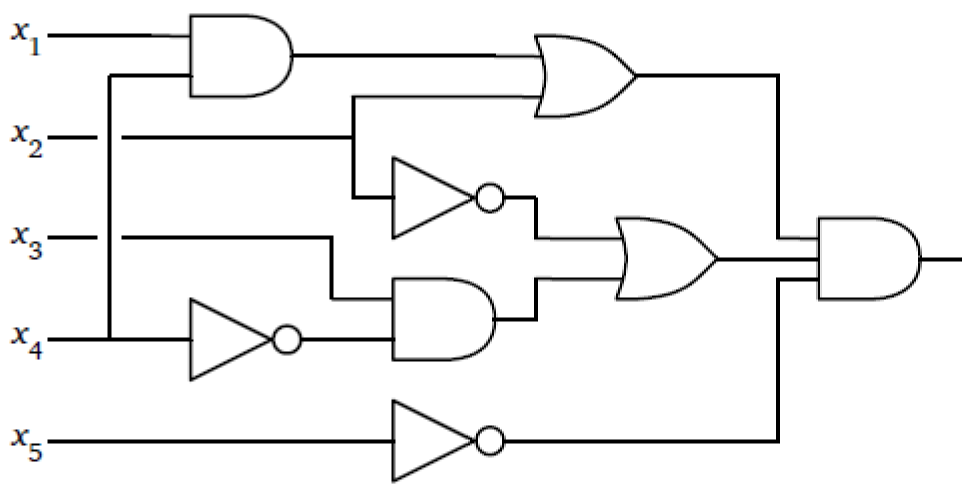


More of what we *think* the world looks like.

polynomal heirarchy

Thm (Cook-Levin) : (The first known NP-Hard problem)

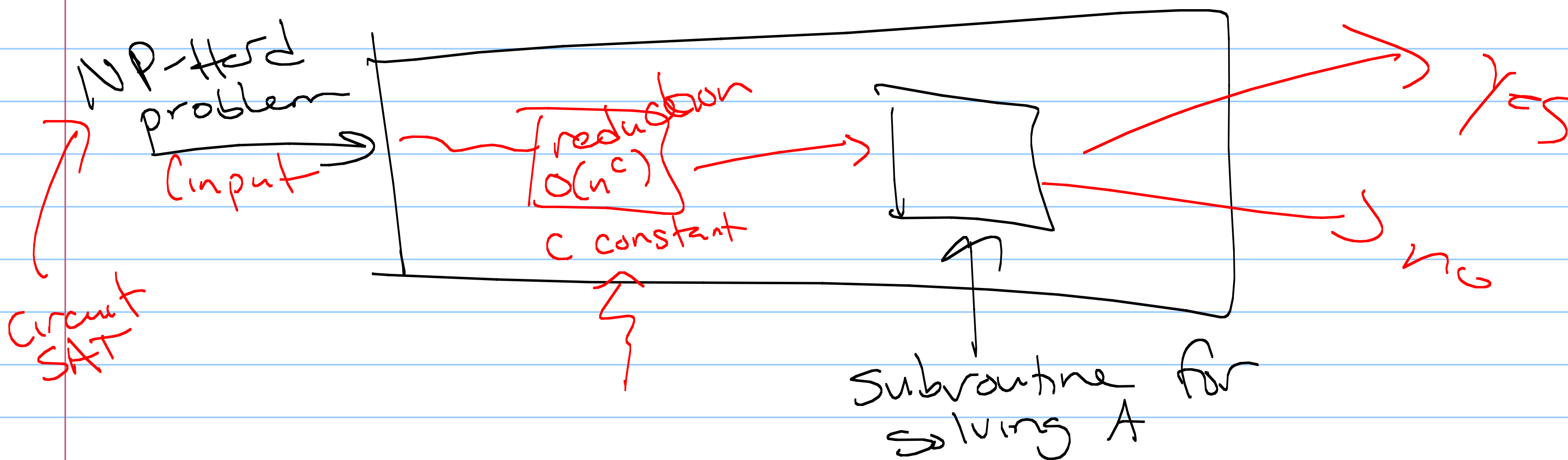
Circuit Satisfiability is NP-Complete.



"pf": any turing machine can be turned into a circuit

To prove NP-Hardness of A:

☆ Reduce a known NP-Hard problem to A.



Let me repeat this:

To prove your problem is hard, solve a different problem using your problem as a subroutine!

Known NP-Hard

Dfn: SAT takes a boolean formula & asks if it is possible to to assign booleans to the formula is true.

Ex:

$$(a \vee b \vee c \vee \bar{d}) \Leftrightarrow ((b \wedge \bar{c}) \vee \overline{(\bar{a} \Rightarrow d)} \vee (c \neq a \wedge b))$$

m variables

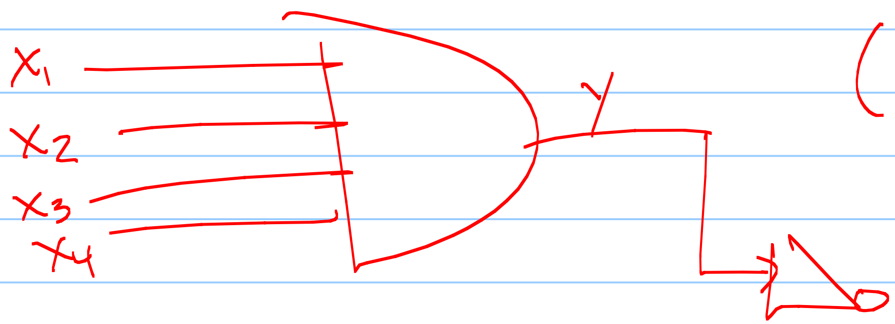
n clauses

in NP: given assignment a, b, c, d, can check if it evaluates to true in $O(m+m)$ time

Thm SAT is NP-Complete

Pf: Reduction:

Reduce circuit SAT to SAT

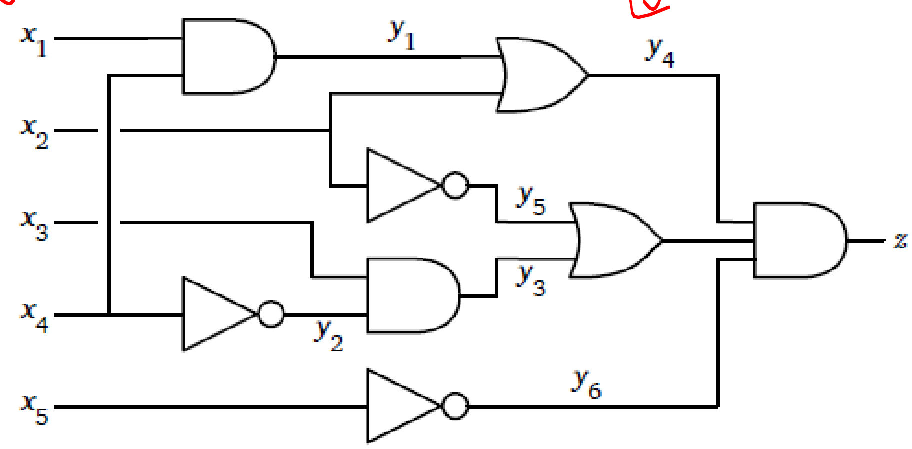


$$(x_1 \wedge x_2 \wedge x_3 \wedge x_4 = y)$$

$$\wedge\ (\overline{y}) \ \ldots$$

Picture:

known hard one



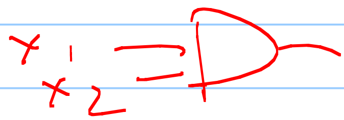
change

$$(y_1 = x_1 \wedge x_4) \wedge (y_2 = \overline{x_4}) \wedge (y_3 = x_3 \wedge y_2) \wedge (y_4 = y_1 \vee x_2) \wedge$$
$$(y_5 = \overline{x_2}) \wedge (y_6 = \overline{x_5}) \wedge (y_7 = y_3 \vee y_5) \wedge (z = y_4 \wedge y_7 \wedge y_6) \wedge z$$

z must be true + intermediate variable correspond to circuit

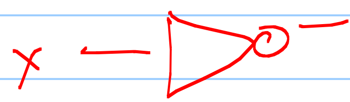
① Any gate: write equivalent equation

$\Rightarrow$



: $(y = x_1 \wedge x_2)$

$(y = x_1 \vee x_2)$



$(y = \overline{x})$

② "And" these together, & "and" on final output

But careful — formula is bigger!
only n inputs to circuit. ☺
(with m gates)
How many variables / clauses in the SAT instance?

each gate makes 1 clause
each gate introduces 1 new variable
+ z

$n+m$ ⟶ $O(m+n)$

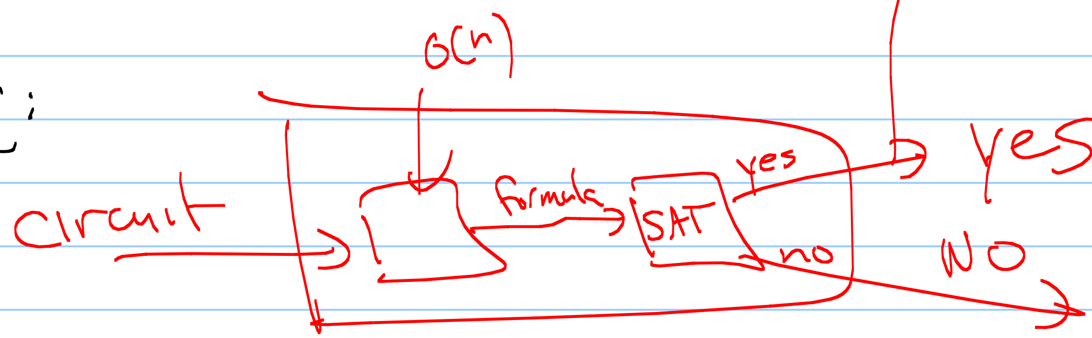
So our reduction looks like this:

circuit SAT

$$\begin{tikzcd}
\boxed{\text{boolean circuit}} \arrow[r, "O(n)"] & \boxed{\text{boolean formula}} \arrow[d, Rightarrow, "\text{SAT}"] \\
\boxed{\text{True or False}} & \boxed{\text{True or False}} \arrow[l, "\text{trivial}"']
\end{tikzcd}$$

"subroutine" for SAT

or:



3SAT: a restriction of SAT

Dfn: Conjunctive normal form (CNF)

$$\overbrace{(a \vee b \vee c \vee d)}^{\text{clause}} \wedge (b \vee \bar{c} \vee \bar{d}) \wedge (\bar{a} \vee c \vee d) \wedge (a \vee \bar{b})$$

each clause is an "OR"

"and"s between clauses

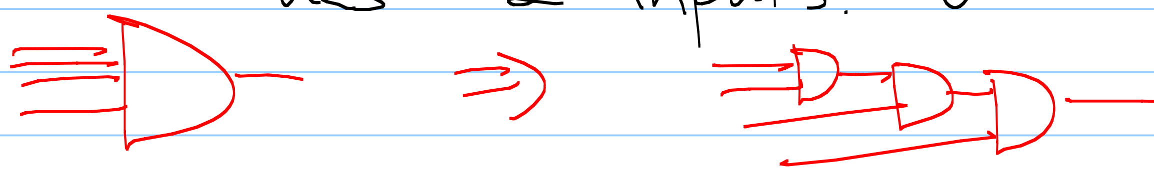
3SAT: SAT where we have CNF + exactly 3 literals per clause

Thm: 3SAT is NP-Hard.

pf: Reduce circuit SAT to 3SAT.

Need to show any circuit can be written in CNF form.

Steps: (1) Make sure each gate has 2 inputs.



(2) Write down formula, 1 clause per gate.
(Same as last one)

$$\begin{cases} y = x_1 \vee x_2 \\ y = x_1 \wedge x_2 \\ y = \overline{x} \end{cases}$$

③ Need each gate to be in 3CNF form!
(not $y = x_1 \vee x_2$, etc.)

3 types:

$$a = b \wedge c \longmapsto (a \vee \bar{b} \vee \bar{c}) \wedge (\bar{a} \vee b) \wedge (\bar{a} \vee c)$$

$$a = b \vee c \longmapsto (\bar{a} \vee b \vee c) \wedge (a \vee \bar{b}) \wedge (a \vee \bar{c})$$

$$a = \bar{b} \longmapsto (a \vee b) \wedge (\bar{a} \vee \bar{b})$$

(exercise)

4 Need exactly 3 per clause.

Solution:

$$\begin{aligned} a &\longmapsto (a \vee x \vee y) \wedge (a \vee \bar{x} \vee y) \wedge (a \vee x \vee \bar{y}) \wedge (a \vee \bar{x} \vee \bar{y}) \\ a \vee b &\longmapsto (a \vee b \vee x) \wedge (a \vee b \vee \bar{x}) \end{aligned}$$

Note: even bigger! Last gate we saw:

$$
\begin{aligned}
&(y_1 \vee \overline{x_1} \vee \overline{x_4}) \wedge (\overline{y_1} \vee x_1 \vee z_1) \wedge (\overline{y_1} \vee x_1 \vee \overline{z_1}) \wedge (\overline{y_1} \vee x_4 \vee z_2) \wedge (\overline{y_1} \vee x_4 \vee \overline{z_2}) \\
\wedge &(y_2 \vee x_4 \vee z_3) \wedge (y_2 \vee x_4 \vee \overline{z_3}) \wedge (\overline{y_2} \vee \overline{x_4} \vee z_4) \wedge (\overline{y_2} \vee \overline{x_4} \vee \overline{z_4}) \\
\wedge &(y_3 \vee \overline{x_3} \vee \overline{y_2}) \wedge (\overline{y_3} \vee x_3 \vee z_5) \wedge (\overline{y_3} \vee x_3 \vee \overline{z_5}) \wedge (\overline{y_3} \vee y_2 \vee z_6) \wedge (\overline{y_3} \vee y_2 \vee \overline{z_6}) \\
\wedge &(\overline{y_4} \vee y_1 \vee x_2) \wedge (y_4 \vee \overline{x_2} \vee z_7) \wedge (y_4 \vee \overline{x_2} \vee \overline{z_7}) \wedge (y_4 \vee \overline{y_1} \vee z_8) \wedge (y_4 \vee \overline{y_1} \vee \overline{z_8}) \\
\wedge &(y_5 \vee x_2 \vee z_9) \wedge (y_5 \vee x_2 \vee \overline{z_9}) \wedge (\overline{y_5} \vee \overline{x_2} \vee z_{10}) \wedge (\overline{y_5} \vee \overline{x_2} \vee \overline{z_{10}}) \\
\wedge &(y_6 \vee x_5 \vee z_{11}) \wedge (y_6 \vee x_5 \vee \overline{z_{11}}) \wedge (\overline{y_6} \vee \overline{x_5} \vee z_{12}) \wedge (\overline{y_6} \vee \overline{x_5} \vee \overline{z_{12}}) \\
\wedge &(\overline{y_7} \vee y_3 \vee y_5) \wedge (y_7 \vee \overline{y_3} \vee z_{13}) \wedge (y_7 \vee \overline{y_3} \vee \overline{z_{13}}) \wedge (y_7 \vee \overline{y_5} \vee z_{14}) \wedge (y_7 \vee \overline{y_5} \vee \overline{z_{14}}) \\
\wedge &(y_8 \vee \overline{y_4} \vee \overline{y_7}) \wedge (\overline{y_8} \vee y_4 \vee z_{15}) \wedge (\overline{y_8} \vee y_4 \vee \overline{z_{15}}) \wedge (\overline{y_8} \vee y_7 \vee z_{16}) \wedge (\overline{y_8} \vee y_7 \vee \overline{z_{16}}) \\
\wedge &(y_9 \vee \overline{y_8} \vee \overline{y_6}) \wedge (\overline{y_9} \vee y_8 \vee z_{17}) \wedge (\overline{y_9} \vee y_8 \vee \overline{z_{17}}) \wedge (\overline{y_9} \vee y_6 \vee z_{18}) \wedge (\overline{y_9} \vee y_6 \vee \overline{z_{18}}) \\
\wedge &(y_9 \vee z_{19} \vee z_{20}) \wedge (y_9 \vee \overline{z_{19}} \vee z_{20}) \wedge (y_9 \vee z_{19} \vee \overline{z_{20}}) \wedge (y_9 \vee \overline{z_{19}} \vee \overline{z_{20}})
\end{aligned}
$$

How big?

Each gate was transformed into at most 5 (6?) clauses.
(& introduced ≤ 3 new variables scattered throughout)

Can transform in polynomial time, & has $O(n)$ size.

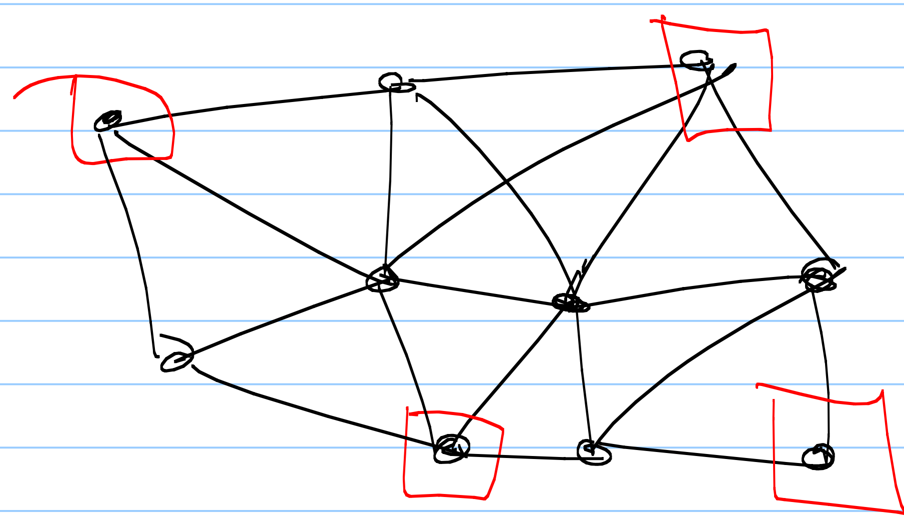
So we have:

$$\begin{tikzcd}
\boxed{\text{boolean circuit}} \arrow[r, "O(n)"] & \boxed{\text{3CNF formula}} \arrow[d, Rightarrow, "\text{3SAT}"] \\
\boxed{\textsc{True or False}} & \boxed{\textsc{True or False}} \arrow[l, "\text{trivial}"']
\end{tikzcd}$$

$$T_{\mathrm{CSAT}}(n) \le O(n) + T_{\mathrm{3SAT}}(O(n)) \quad \Longrightarrow \quad T_{\mathrm{3SAT}}(n) \ge T_{\mathrm{CSAT}}(\Omega(n)) - O(n)$$

Next problem: Independent Sets

A set of vertices in a graph with no edges between them.



decision version: is there an indep set of size = k in G?

↳ in NP

Challenge: Nothing like a boolean!

To show NP-Hard, need to reduce
SAT, circuit SAT, or 3-SAT
to a <u>graph</u> problem.

We'll use 3SAT
(but you should marvel a bit..)

$(\quad\quad) \wedge (\quad\quad)$

$(a \vee \underline{b} \vee \bar{c}) \wedge (\underline{\bar{b}} \vee c \vee d)$

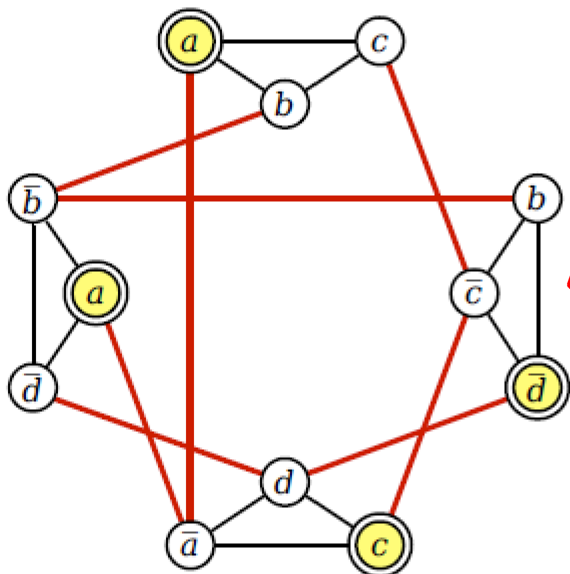
Construct a graph:

① A vertex for each literal in each clause.

n clauses w/ m inputs
↳ 3n vertices

② Connect two vertices if:
- they are in the same clause
- they are a variable & its inverse

Ex: $(a \vee b \vee c) \wedge (b \vee \bar{c} \vee \bar{d}) \wedge (\bar{a} \vee c \vee d) \wedge (a \vee \bar{b} \vee \bar{d})$

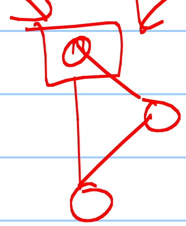


Claim: formula is satisfiable $\Longleftrightarrow$ G has indep. set of size k

pf: $\Rightarrow$ Suppose 3CNF was satisfiable.

$(\_ \vee \_ \vee \_) \wedge (\_\ \_\ \_) \wedge (\quad)$

in indep set



Each clause evaluates to true under this satisfying assignment,
$\Rightarrow$ at least one literal in each was true
Pick corresponding vertex to be in I.S.

1 per clause $\Rightarrow$ one per Δ.

Can't have vertex corresponding to variable & its inverse since if x is true in 3SAT, then $\bar{x}$ would have been false.

$\Leftarrow$: Spps ind. set of size n. must take at most 1 vertex per Δ.
Since n Δ's, must take exactly 1 per Δ.
Set that variable = true in 3SAT.

So:

$$\begin{tikzcd}
\boxed{\text{3CNF formula with } k \text{ clauses}} \arrow[r, "O(n)"] & \boxed{\text{graph with } 3k \text{ nodes}} \arrow[d, Rightarrow, "\textsc{MaxIndSet}"] \\
\boxed{\textsc{True or False}} & \boxed{\text{maximum independent set size}} \arrow[l, "O(1)"']
\end{tikzcd}$$

$$T_{\text{3SAT}}(n) \le O(n) + T_{\textsc{MaxIndSet}}(O(n)) \quad\Longrightarrow\quad T_{\textsc{MaxIndSet}}(n) \ge T_{\text{3SAT}}(\Omega(n)) - O(n)$$