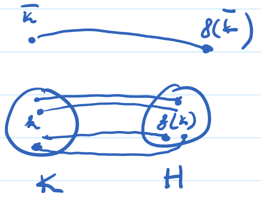
SATISFIABILITY: THE ORIGINAL NP-HARD PROBLEM

Thursday, November 29, 2018 5:07 PM

Assumptions: 3-COLOR, HP/HC, PARTITION, LP are NP-hard problems

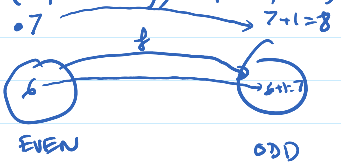
To show that a new problem H is NP-hard, we REDUCE a known NP-hard $\overset{K}{\text{problem}}$ to H.
In symbols, $K \le H$.



To show $K \le H$ we must do 3 things

1) give a poly-time function f such that
2) if $k \in K$ then $f(k) \in H$, and
3) if $f(k) \in H$, then $k \in K$.
   (equivalently, if $k \notin K$, then $f(k) \notin H$)

Ex:



EVEN ≤ ODD

$f(n) = n+1$

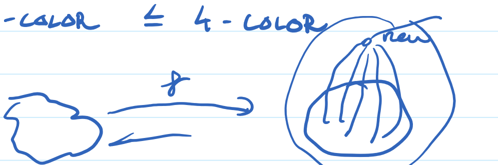
$n \notin EVEN \Rightarrow f(n) \notin ODD$

is equivalent to

$f(n) \in ODD \Rightarrow n \in EVEN$

$A \Rightarrow B$
eg $\neg B \Rightarrow \neg A$

3-COLOR ≤ 4-COLOR



G is 3-colorable → G' is 4-color
←

Cook
Karp

THE ORIGINAL NP-HARD PROBLEM: SATISFIABILITY (SAT)

Every other NP-hard problem was shown to be hard by reducing SAT to it.

Def: A boolean formula is an expression involving variables $x_1, x_2, \dots, x_n, \dots$ whose values are either T or F, and operators $\neg$ (NOT), $\vee$ (OR), $\wedge$ (AND) and ( ). $\neg$ has the highest precedence, followed by $\wedge$, and followed by $\vee$

Ex.

x_1

$\neg x_1$

$x_1 \vee x_2$

$x_1 \wedge x_2$

$\neg x_1 \vee x_2 \wedge \neg x_3$

$(\neg x_1 \vee x_2) \wedge \neg x_3$

SAT

INPUT: a boolean formula X using n variables $x_1, \dots, x_n$

OUTPUT: TRUE if there is an assignment $a: \{x_1, \dots, x_n\} \to \{T, F\}$ such that $a(X) = T$

Ex.: $X = x_1 \lor x_2 \land \neg x_3$

$\boxed{T \lor T \land \neg F}$

$T \lor \underbrace{T \land T}$

$\underbrace{T \lor T}$

$\boxed{T}$

$a(x_1) = T$
$a(x_2) = T$
$a(x_3) = F$

X is SATISFIABLE

Ex: $X = x_1 \land \neg x_1$

$a(x_1) = T$ or $a(x_1) = F$

$a(X) = T \land \neg T = T \land F = F$
$a(X) = F \land \neg F = F \land T = F$

X is NOT SATISFIABLE

OPEN: Is SAT in P? Brute force $\Omega(2^n \times n)$

SIMPLER VERSIONS OF SAT:

1) CNF-SAT: input is a boolean formula in CONJUNCTIVE NORMAL FORM (CNF)

Def. A literal is a variable or the negation of a variable
Ex: $x_1, \neg x_5$

Def. A clause is a disjunction of literals: $C = \ell_1 \lor \ell_2 \lor \ell_3 \lor \dots \lor \ell_k$
Ex. $C = x_1 \lor \neg x_1 \lor x_2 \lor \neg x_5 \lor x_7$

Def: A boolean formula in CNF is a conjunction of clauses
$X = C_1 \land C_2 \land \dots \land C_m$
Ex: $x_1 \land (x_1 \lor \neg x_2) \land (x_1 \lor \neg x_4 \lor x_5 \lor \neg x_7) \land (x_2 \lor \neg x_3)$

OPEN: CNF-SAT is in P?

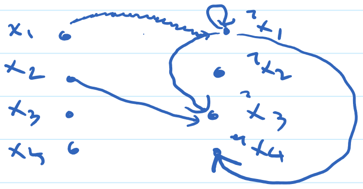
2) $\boxed{\text{3-CNF}}$: input is a CNF formula, where each clause has EXACTLY 3 literals
Ex: $X = (x_1 \lor \neg x_1 \lor x_2) \land (x_1 \lor x_3 \lor x_4) \land (\neg x_2 \lor \neg x_3 \lor \neg x_5) \land (x_1 \lor \neg x_2 \lor \neg x_3)$

OPEN: 3-CNF is in P?

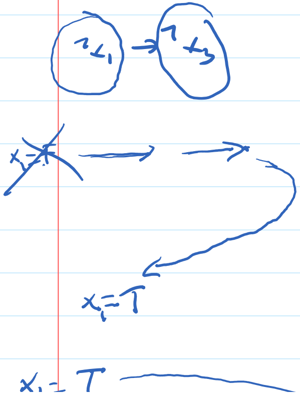
3) 2-CNF: input is a CNF formula, where each clause has EXACTLY 2 literals
Ex: $X = (x_1 \lor \neg x_1) \land (x_1 \lor \neg x_3) \land (\neg x_2 \lor \neg x_3) \land (x_1 \lor \neg x_4)$

KNOWN: 2-CNF $\in$ P !!!

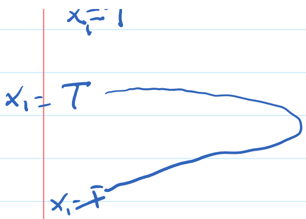
Proof sketch: 1) Convert each clause $(x \lor y)$ into $\neg x \to y$
2) Construct a graph G with $2n$ vertices
$x_1, \neg x_1, x_2, \neg x_2, \dots x_n, \neg x_n$



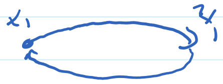
$x_1 \lor \neg x_1 \equiv \neg x_1 \to \neg x_1$
$x_1 \lor \neg x_3 \equiv \neg x_1 \to \neg x_3$
$\neg x_2 \lor \neg x_3 \equiv x_2 \to \neg x_3$
$x_1 \lor \neg x_4 \equiv \neg x_1 \to \neg x_4$



3) X is satisfiable $\Leftrightarrow$ G has no cycles containing a variable x and its negation $\neg x$



3) X is satisfiable $\Leftrightarrow$ G has no cycles containing a variable x and its negation $\neg x$



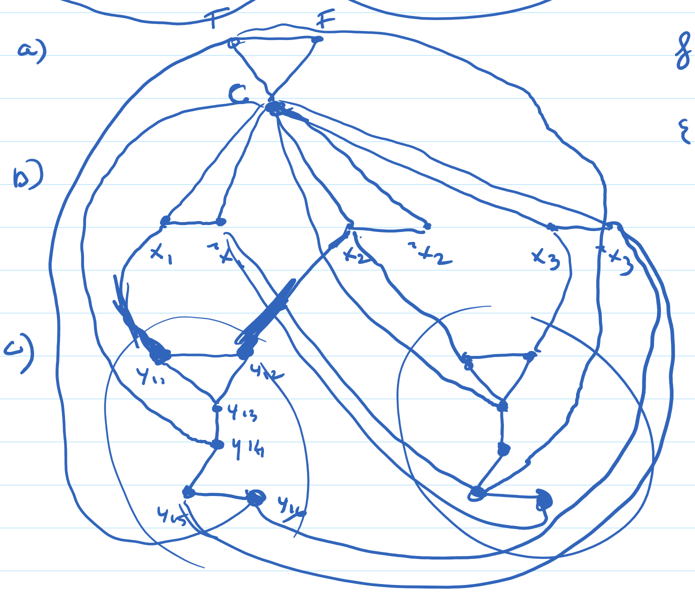
THEOREM (KARP) : 3-CNF $\leq$ 3-COLOR

1) Poly-time reduction f :

Ex:

$$X = (\overset{T}{x_1} \vee \overset{T}{x_2} \vee \overset{F}{\neg x_3}) \wedge (\overset{T}{x_2} \vee \overset{T}{x_3} \vee \overset{F}{\neg x_1})$$

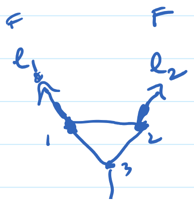
$f(X)$: a)



```
f(X = $C_1 \wedge C_2 \wedge \cdots \wedge C_m$)
      on $x_1, \cdots, x_n$
{
    V = {C, T, F}                                   O(1)
    E = {{C,T}, {C,F}, {T,F}}
    for (i = 1; i ≤ n; ++i)                         O(n)
    {
        V += {$x_i$, $\neg x_i$}
        E += {{$x_i$, C}, {$x_i$, $\neg x_i$}
              {$\neg x_i$, C}}
    }
    for (i = 1; i ≤ m; ++i)
    {
        V += {$y_{i1}$, ..., $y_{i6}$}
        E += {{$y_{i1}$, $y_{i2}$}                  O(m)
              {$y_{i1}$, $y_{i3}$}
              {$y_{i2}$, $y_{i3}$}
              {$y_{i3}$, $y_{i4}$}
              {$y_{i4}$, $y_{i5}$}
              {T, $y_{i6}$}
              {$y_{i5}$, $y_{i6}$}
              {$y_{i1}$, $l_{i1}$}
              {$y_{i2}$, $l_{i2}$}
              {$y_{i5}$, $l_{i3}$}
              {$y_{i6}$, $l_{i3}$}
              {$y_{i4}$, C}
        }
    }
    return (V, E)
}
```

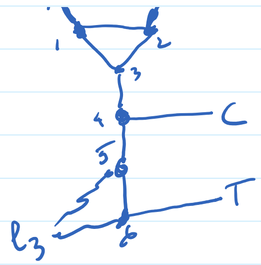
Running time: $O(n+m)$

Theorem :



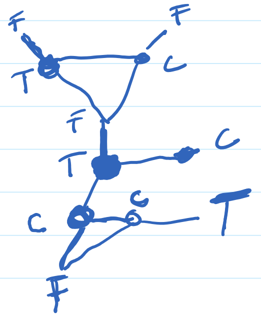
not 3-colorable if and only if $l_1 = l_2 = l_3 = F$

Theorem :



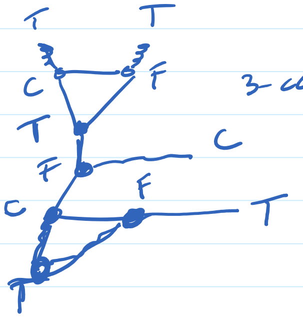
not 3-colorable if and only if $\ell_1 = \ell_2 = \ell_3 = F$

FFF :



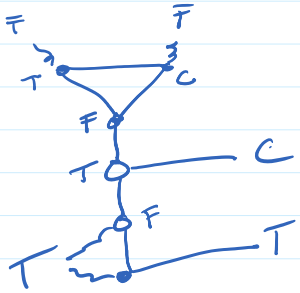
NOT 3-colorable

TTT

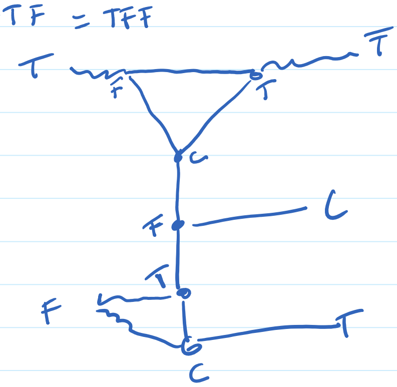


3-colorable

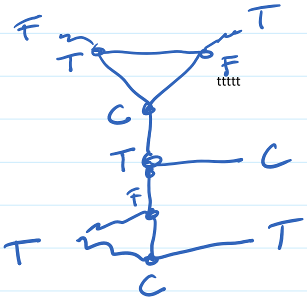
FF T:



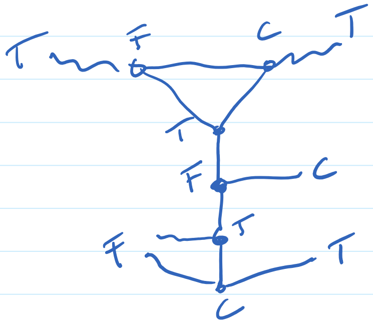
FTF = TFF



FTT = TFT



TT F



If X is satisfiable, then there is an assignment such that each clause has at least 1 TRUE literal. Hence each gadget is 3-colorable and G is 3-colorable

Conversely, if G is 3-colorable, then each gadget is 3 colorable, so each clause has at least one T literal, so X is satisfiable.