RECAP

- BC-SAT $\in$ NPC (Cook, 1971)
- SAT $\in$ NPC (BC-SAT $\leq_p$ SAT, make sure that $f(x)$ is polynomial)
- 3-CNF-SAT $\in$ NPC (reduction by restriction)
- CLIQUE $\in$ NPC (3-CNF-SAT $\leq_p$ CLIQUE, "context switch"):

$$f(\langle \phi(x_1, \ldots, x_m) = \bigwedge_{i=1}^{m} C_i \rangle) = \langle G_\phi, k_\phi \rangle$$

$C_i = y_1^i \vee y_2^i \vee y_3^i$

1. $k_\phi = m$
2. $G_\phi = (V_\phi, E_\phi)$

$V_\phi = \{ v_j^i : 1 \leq i \leq m, 1 \leq j \leq 3 \} \Rightarrow |V_\phi| = 3m$

one node per literal occurrence

E_ϕ:

$\{ v_{j_1}^{i_1}, v_{j_2}^{i_2} \} \in E_\phi \Leftrightarrow (i_1 \neq i_2) \wedge (y_{j_1}^{i_1} \neq \neg y_{j_2}^{i_2})$

edges between literals of different clauses that can both be true under the same truth assignment

We have to show that:

$$\langle \phi \rangle \in \text{3-CNF-SAT} \iff f(\langle \phi \rangle) = \langle G_\phi, k_\phi \rangle \in \text{CLIQUE}$$

$\Rightarrow$

$\langle \phi \rangle \in$ 3-CNF-SAT : there exists a satisfying assignment $\vec{b} \in \{0,1\}^m : \phi(\vec{b}) = 1$. Since ϕ is in 3-CNF, all clauses must be true under $\vec{b}$ $\Rightarrow$ under $\vec{b}$ there is a true literal in each clause.

Let these literals be:

$$\{ y^1_{j_1}, y^2_{j_2}, \dots, y^m_{j_m} \} \qquad 1 \le j_i \le 3, \; 1 \le i \le m$$

and let $V' = \{ v^1_{j_1}, v^2_{j_2}, \dots, v^m_{j_m} \}$

(corresponding nodes in G_ϕ.)

For two distinct nodes $v^{i_1}_{j_{i_1}}, v^{i_2}_{j_{i_2}} \in V'$,

we have:

$$(i_1 \neq i_2) \wedge \left(y^{i_1}_{j_{i_1}} \text{ and } y^{i_2}_{j_{i_2}} \right) \text{ both true under } \vec{b}$$

$$\Longrightarrow \{ v^{i_1}_{j_{i_1}}, v^{i_2}_{j_{i_2}} \} \in E$$

Thus V' is a CLIQUE of size $k_\phi = m$

$\Rightarrow f(\langle \phi \rangle) = \langle G_\phi, k_\phi \rangle \in$ CLIQUE

⟸ meglio supporre che grafo venga da f

$$f(\langle\phi\rangle) = \langle G_\phi, K_\phi \rangle \in \text{CLIQUE}$$

Let us build a satisfying assignment $\vec{b}$ starting from a clique $V' = \{v^1_{j_1}, v^2_{j_2}, \dots, v^m_{j_m}\}$:

1 nodo per clausola

For $1 \le i \le m$:

1. If $y^i_{j_i} = x_k$ then set $b_k = 1$
2. If $y^i_{j_i} = \bar{x}_k$ then set $b_k = 0$

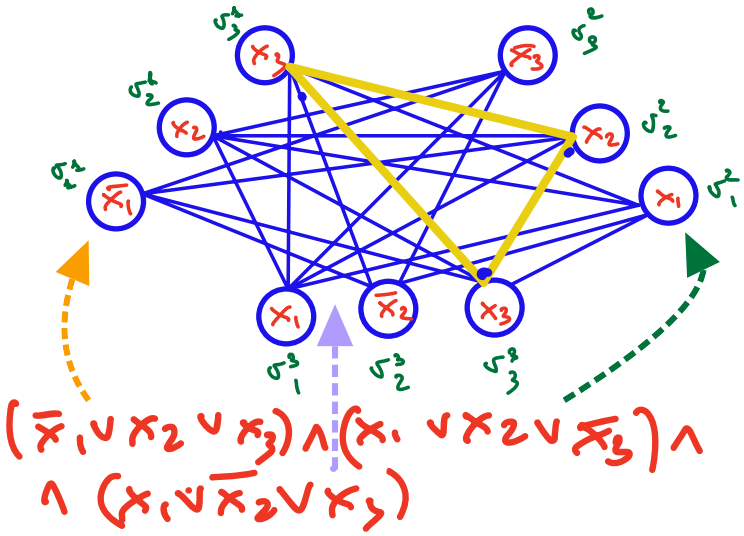
If some value b_ℓ is still unfixed, set it to an arbitrary value

Assignment $\vec{b}$ is well-defined since I cannot set $b_k = 0$ for some i_1 and $b_k = 1$ for some i_2, or otherwise $y^{i_1}_{j_{i_1}} = \bar{y}^{i_2}_{j_{i_2}}$

(impossible, since V' is a clique)

Under $\vec{b}$ there is at least one true literal in each clause

$\Rightarrow \phi(\vec{b}) = 1 \Rightarrow \langle\phi\rangle \in$ 3-CNF-SAT



A 3-clique contains 3 literals (one per clause) that can be made all true under the same truth assignment

$v^1_3 \leftrightarrow x_3 \rightarrow b_3 = 1$

$v^2_2 \leftrightarrow x_2 \rightarrow b_2 = 1$

$v^3_3 \leftrightarrow x_3 \rightarrow b_3 = 1$

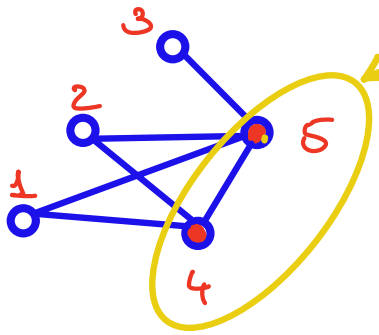
$b_1 = 0/1$

We introduce two extra problems on graphs:

DEF: Given $G=(V,E)$ undirected, a **vertex cover** of size K is a subset of nodes $V' \subseteq V$, $|V'|=K$ such that each edge in E has at least one endpoint in V'

$$V' \subseteq V, |V'|=K : \forall u \neq v \in V : \{u,v\} \in E \Rightarrow (u \in V') \vee (v \in V')$$

EXAMPLE:



$\{4,5\}$ is a vertex cover of size 2

We have:

VERTEX COVER (VC)

- I: $\langle G=(V,E), K \rangle$, G undirected graph, $1 \leq K \leq |V|$
- Q: Does G contain a vertex cover of size K?

The optimization version of this problem requires determining the vertex cover of min size

A vertex cover identifies a (possibly small) set of nodes from where all edges can be controlled. Crucial in networking, distributed computing, etc...

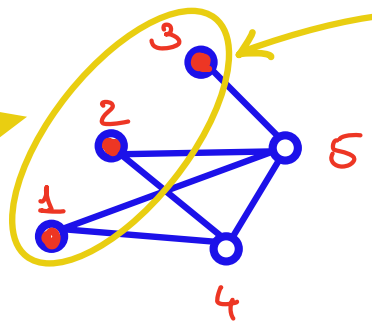
Second problem:

DEF: Given $G=(V,E)$ undirected, an **independent set** of size K is a subset of nodes $V' \subseteq V$, $|V'|=K$ such that there is no edge in E with both endpoints in V':

$$V' \subseteq V, |V'|=K : \forall u \neq v \in V : (u \in V') \wedge (v \in V') \Rightarrow \{u,v\} \notin E$$

EXAMPLE:

complement of the vertex-cover $\{4,5\}$!



$\{1, 2, 3\}$ is an independent set of size 3

We have:

INDEPENDENT SET (IS)

- I: $\langle G=(V,E), K \rangle$, G undirected graph, $1 \leq K \leq |V|$
- Q: Does G contain an independent set of size K?

The optimization version of this problem requires determining the independent set of max size

An independent set is somewhat the "negative" of a clique, with no edges connecting any two of its nodes. Important

primitive for graph-based computations.

The three problems share many similarities. We will see that reductions will be mostly "syntactic", based on logical manipulation of the definitions of clique, vertex-cover, independent-set.

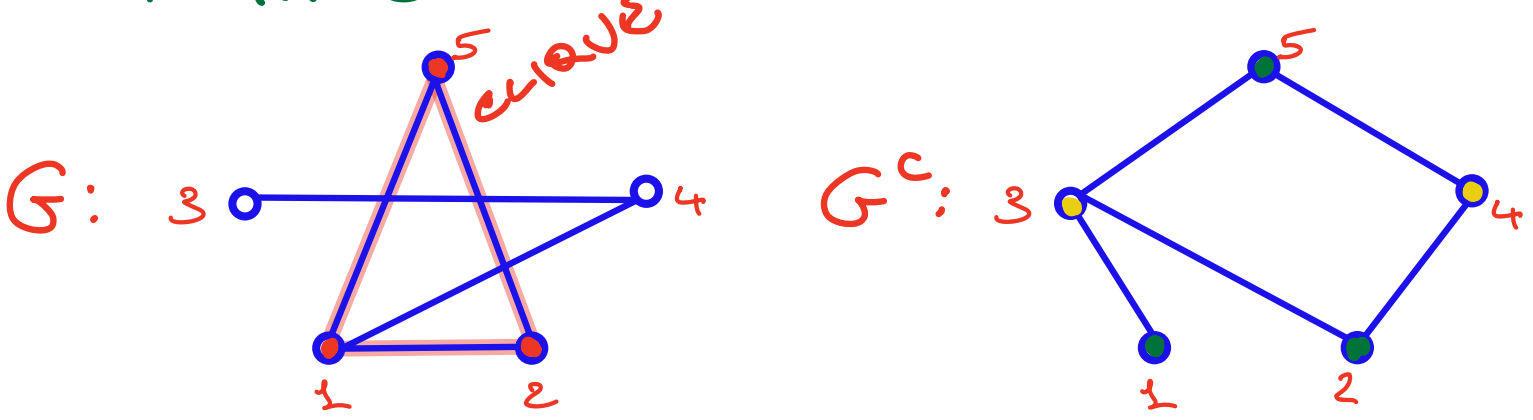
We will make use of the complement of a graph

DEF: Given $G=(V,E)$ its complement $G^c=(V,E^c)$ is the graph with the same node-set and

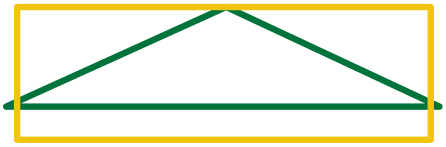
$$E^c: \{\{u,v\} : u \neq v \in V,\ \{u,v\} \notin E\}$$

G^c is the "negative" of G w.r.t. edges

EXAMPLE



"Gadgets"



OBSERVE: Clique $\{1,2,5\}$ in G implies that:

1. $V - \{1,2,5\} = \{3,4\}$ is a vertex cover in G^c (all edges between $\{1,2,5\}$ disappear in E^c, thus each edge in E^c has one endpoint in $\{3,4\}$)
2. $\{1,2,5\}$ is an independent set in G^c (there cannot be edges between $\{1,2,5\}$ in E^c)

These observations suggest the following two reductions:

1. CLIQUE $\leq_p$ VC using:

$$f_{C \to VC}(\langle G=(V,E), K\rangle) = \langle G^c=(V,E^c), |V|-K\rangle$$

2. CLIQUE $\leq_p$ IS using:

$$f_{C \to IS}(\langle G=(V,E), K\rangle) = \langle G^c=(V,E^c), K\rangle$$

Clearly, $f_{C \to VC}$ and $f_{C \to IS}$ are ptc (the construction of $\langle G^c, |V|-K\rangle$ and $\langle G^c, K\rangle$ requires time $O(|\langle G, K\rangle|^2)$)

Let's prove that $f_{C \to VC}$ is a reduction by logical manipulations of the definition of CLIQUE:

$X = \langle G=(V,E), K \rangle \in$ CLIQUE

$\Leftrightarrow \exists V' \subseteq V, (|V'|=K) \wedge (\forall u \neq v \in V: \underline{(u \in V') \wedge (v \in V') \Rightarrow \{u,v\} \in E}$

{transform using counterpositive}

$\Leftrightarrow \exists V' \subseteq V, (|V'|=K) \wedge (\forall u \neq v \in V: \underline{\{u,v\} \notin E \Rightarrow (u \notin V') \vee (v \notin V')}$

$\{\neg(a \wedge b) = \neg a \vee \neg b\}$

$\Leftrightarrow \exists V' \subseteq V, (|V'|=K) \wedge (\forall u \neq v \in V: \{u,v\} \in E^c \Rightarrow \begin{matrix}(u \in V-V') \\ \vee \\ (v \in V-V')\end{matrix}$

{now, set $V''=V-V'$: $|V''| = |V|-K$}

$\Leftrightarrow \exists V'' \subseteq V, (|V''|=|V|-K) \wedge (\forall u \neq v \in V: \{u,v\} \in E^c \Rightarrow \begin{matrix}(u \in V'') \\ \vee \\ (v \in V'')\end{matrix}$

{this says that V'' is a VC of G^c}

$\Leftrightarrow f(x) = \langle G^c = (V, E^c), |V|-K \rangle \in VC$

REMARK: The proof proceeds by logical equivalences

Let's use the same technique to prove that $f_{C \to IS}$ is a reduction:

$X = \langle G=(V,E), K \rangle \in CLIQUE$

$\Leftrightarrow \exists V' \subseteq V, (|V'|=K) \wedge (\forall u \neq v \in V: (u \in V') \wedge (v \in V') \Rightarrow \{u,v\} \in E$

$\Leftrightarrow \exists V' \subseteq V, (|V'|=K) \wedge (\forall u \neq v \in V: (u \in V') \wedge (v \in V') \Rightarrow \{u,v\} \notin E^c$

{this says that V' is an IS of G^c}

$\Leftrightarrow f(x) = \langle G^c=(V,E^c), K \rangle \in IS$

EXERCISE Using the same technique, prove that $VC \leq_p IS$

These reductions suggest that CLIQUE, VC and IS are mostly equivalent problems

BUT

this is true only for what concerns exact solutions! We will argue that the optimization versions of the three problems are radically different w.r.t. approximation. In particular:

MINIMUM VC is "easy" to approximate

MAXIMUM CLIQUE/IS are "hard" to approximate

We now introduce a problem in the realm of COMPUTER ARITHMETIC:

- SUBSET-SUM (SS)

$$\begin{cases} I : \langle S, t \rangle : S \subseteq \mathbb{N} \text{ finite}, \ t \in \mathbb{N} \ (\text{"target"}) \\ Q : \exists S' \subseteq S : \sum_{s \in S'} s = t \ ? \end{cases}$$

The problem asks whether there is a subset of the input set whose sum is equal to t (the target)

EXAMPLE:

$\langle \{10, 15, 4, 3, 17\}, 24 \rangle \in L_{SS}$

since $24 = 4 + 3 + 17$ $(S' = \{3, 4, 17\})$

REMARK: $|\langle S, t \rangle|$ is the total number of bits needed to encode $s \in S$ $\forall s$ and t:

$$|\langle S, t \rangle| = \Theta\left(\sum_{s \in S} \log s + \log t\right)$$

Clearly, $SS \in NP$:

candidate certificate: $y = \langle S' \rangle$ $S' \subseteq S$

Let us prove that $SS \in NPH$ by proving that 3-CNF-SAT $\leq_P$ SS

The reduction is much more complex than the other ones seen before.

Given a 3-CNF formula:

$$\phi(x_1, x_2, \ldots, x_n) = C_1 \wedge C_2 \wedge \ldots \wedge C_m$$

with $C_i = y_1^i \vee y_2^i \vee y_3^i$, $y_j^i \in \{x_k, \bar{x}_k : 1 \le k \le n\}$

The reduction f must create $\langle S_\phi, t_\phi \rangle$ such that $\langle \phi \rangle \in$ 3-CNF-SAT $\Leftrightarrow \langle S_\phi, t_\phi \rangle \in SS$

f creates a set S_ϕ of $2n+2m$ numbers, each with $n+m$ decimal digits:

$S_\phi = \{$ [n+m] [n+m] [n+m] .... [n+m] $\}$ ($2n+2m$)

The n most significant digits correspond to the n variables $x_1, \ldots, x_n$. The m least significant digits correspond to the m clauses

CRUCIAL PROPERTY: the sum of all digits in each position i, $1 \le i \le n+m$ does never generate a carry

this is important to relate the satisfiability of ϕ to the achievement of the target t_ϕ

With no loss of generality:

1. Assume that each variable x_k of $\phi(x_1, \ldots, x_n)$ appears in at least one clause (as $y_j^i = x_k$ or $y_j^i = \bar{x}_k$)
2. Assume that no clause contains x_k and $\bar{x}_k$

Otherwise:

Eliminate variable (1) or clause (2)

$x_k \vee \bar{x}_k = 1$!

We create:

- two numbers per variable x_i: v_i, v_i' $\quad 1 \le i \le n$ $\quad 2n$

- two numbers per clause C_j: s_j, s_j' $\quad 1 \le j \le m$ $\quad 2m$

$\forall 1 \le i \le m$:

- s_i, s'_i: have a single 1 in the ith position of the m most significant digits
- s_i has 1's in all positions j of the m least significant digits, where c_j contains literal x_i
- s'_i has 1's in all positions j of the m least significant digits, where c_j contains literal $\overline{x_i}$
- All other digits are 0

$\forall 1 \le j \le m$:

- s_j has a single 1 in the jth position of the m least significant ones
- s'_j has a single 2 in the jth position of the m least significant ones
- All other digits are 0

Finally, $t_\phi = \underbrace{1\ldots1}_{n}\underbrace{4\ldots4}_{m}$

$n=2$
$m=2$ 1144

(Recall that all numbers have to be interpreted to the base 10)

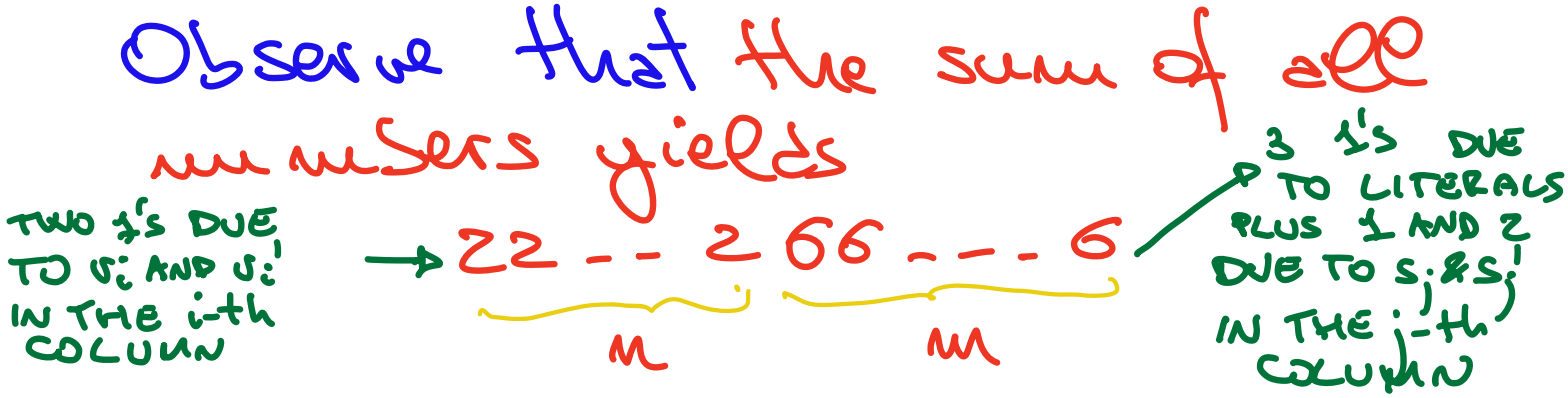
EXAMPLE:

$$\phi(x_1,x_2,x_3) = (x_1 \vee \bar{x}_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3)$$
$$C_1 \qquad\qquad C_2$$

S_ϕ:

	x_1	x_2	x_3	C_1	C_2		
s_1:	1	0	0	1	1	10.011	there are exactly 3 1's in each clause column
s_1':	1	0	0	0	0	10.000	
s_2:	0	1	0	0	1	1.001	
s_2':	0	1	0	1	0	1.010	
s_3:	0	0	1	1	1	111	
s_3':	0	0	1	0	0	100	
s_1:	0	0	0	1	0	10	the sum in each clause column is 3
s_1':	0	0	0	2	0	20	
s_2:	0	0	0	0	1	1	
s_2':	0	0	0	0	2	2	
t_ϕ:	1	1	1	4	4	11.144	

Observe that the sum of all numbers yields



No carry can ever be generated by summing any subset of numbers

$\Rightarrow$ no "interference" between different columns

The running time of the reduction is proportional to the number of digits generated:

$$\underbrace{(2m}_{v_i, v_i'} + \underbrace{2m+1}_{s_j s_j'} + \underbrace{t_\phi}_{t_\phi}) \cdot (n+m) = O((n+m)^2) = O(|\langle \phi \rangle|^2)$$

NOTE: After we generate the numbers in decimal, we can encode them in binary without changing the order of magnitude of the size of the encoding