NP-Completeness (cont)

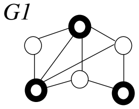
VC (VERTEX-COVER)

Instance: A graph $G = (V,E)$ and an integer m.

Question: Does G have a vertex cover of size m or less ?

(V' is a *vertex cover* for $G = (V,E)$ if for every edge $e=(u,v)$ either $u \in V'$ or $v \in V'$ or both.)

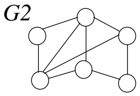
G1



$\in L_{\text{VC}}$?

m=3 YES

G2



$\in L_{\text{VC}}$?

m=2 NO

NP-Completeness (cont)

CLIQUE $\leq_P$ VC

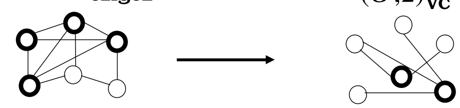
Step 1: Construct $f : \{(G,k)_{\text{CLIQUE}}\} \rightarrow \{(G',m)_{\text{VC}}\}$

Given $G = (V,E)$ and k, let $G' = \overline{G}$ and $m = |V| - k$.

$\overline{G} = (V, \overline{E})$ where $\overline{E} = \{ (u,v) \mid u,v \in V \text{ and } (u,v) \notin E \}$.

Example :

$(G,4)_{\text{CLIQUE}}$ $(\overline{G},2)_{\text{VC}}$



k=4 m=6-4=2

NP-Completeness (cont)

CLIQUE $\leq_P$ VC

Step 2: Show that $(G,k) \in L_{\text{CLIQUE}} \Leftrightarrow f(G,k) \in L_{\text{VC}}$

G has a clique of size $\geq k \Leftrightarrow \overline{G}$ has a vertex cover of size $\leq |V| - k$

Proof: Suppose V' is a clique of size at least k in G.

Pick any two vertices u and v in V.

If the edge (u,v) is in $\overline{E}$, then it is not in E.

So u and v cannot both be in V' (a clique of G).

Either u or v is in $V'' = V - V'$.

V'' is a vertex cover for $\overline{G}$ of size at most $|V|-k$.

NP-Completeness (cont)

CLIQUE $\leq_P$ VC

Proof:(cont) Now suppose V'' is a vertex cover of size at most $|V|-k$ in $\overline{G}$.

Pick any two vertices u and v in $V' = V - V''$.

Then both u and v are not in V'' (a vertex cover for $\overline{G}$).

So the edge (u,v) is not in $\overline{E}$, and hence is in E.

V' is a clique in G of size at least $|V| - (|V|-k) = k$.

QED

NP-Completeness (cont)

CLIQUE $\leq_P$ VC

Step 3: Show that f is computable in polynomial time.

If G has a vertices and b edges then $f(G,k)$ has a vertices and $O(a^2)$ edges.

Whether an edge exists or not in $\overline{G}$ can be determined by inspection of G.

f is computable in polynomial time.

Finished **CLIQUE $\leq_P$ VC**

NP-Completeness (cont)

SUBSET-SUM

Instance: A set of positive integers $S = \{s_1, s_2, \ldots, s_k\}$ and an integer t.

Question: Does S have a subset S' which totals exactly to t ?

Examples: $S = \{2,9,3,5,11,24\}$ and $t = 23$

YES: 11+9+3=23

$S = \{2,9,3,5,11,24\}$ and $t = 15$

NO

NP-Completeness (cont)

VC $\leq_P$ SUBSET - SUM

Step 1: Construct $f : \{(G,m)_{\mathbf{VC}}\} \to \{(S,t)_{\mathbf{SUBSET\text{-}SUM}}\}$

Given $G = (V,E)$ and m where $E = \{e_1, e_2, \ldots, e_{|E|}\}$

create for each $v_i \in V$ the integer

$$s_i = 4^{|E|} + \sum_{j=1}^{|E|} 4^{j-1} \cdot b_{ij} \text{ where } b_{ij} = \begin{cases} 1 & \text{if } v_i \text{ is an endpoint of } e_j \\ 0 & \text{otherwise} \end{cases},$$

create for each $e_j \in E$ the integer $s_{j+|V|} = 4^{j-1}$,

and let $t = m4^{|E|} + \sum_{j=0}^{|E|-1} 2 \cdot 4^j$.

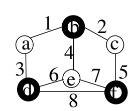
NP-Completeness (cont)

Radix 4 representation of S

VC $\leq_P$ SUBSET - SUM

Example :

$(G,3)_{\mathbf{VC}}$



$m=3$

$(S,t)_{\mathbf{SUBSET\text{-}SUM}}$

$V \downarrow$ / $E \rightarrow$		8	7	6	5	4	3	2	1
a	1	0	0	0	0	0	1	0	1
b	1	0	0	0	0	1	0	1	1
c	1	0	0	0	1	0	0	1	0
d	1	1	0	1	0	0	1	0	0
e	1	0	1	1	0	1	0	0	0
f	1	1	1	0	1	0	0	0	0
1	0	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0	1	0
3	0	0	0	0	0	0	1	0	0
4	0	0	0	0	0	1	0	0	0
5	0	0	0	0	1	0	0	0	0
6	0	0	0	1	0	0	0	0	0
7	0	0	1	0	0	0	0	0	0
8	0	1	0	0	0	0	0	0	0
t	3	2	2	2	2	2	2	2	2

NP-Completeness (cont)

VC $\leq_P$ SUBSET - SUM

Step 2: Show that $(G,m) \in L_{\mathbf{VC}} \Leftrightarrow f(G,m) \in L_{\mathbf{SUBSET\text{-}SUM}}$

G has a vertex cover of size $\leq m \Leftrightarrow$

S has a subset S' that totals to t

Proof: Suppose V' is a vertex cover of size m in G.

(If G has a vertex cover of size $\leq m$, then it has one of size m.)

Construct a subset $S' \subseteq S$ as follows :

for each $v_i \in V'$, add s_i to S'

for each e_j with only one endpoint in V', add $s_{j+|V|}$ to S'

NP-Completeness (cont)

VC $\leq_P$ SUBSET - SUM

Proof:(cont) Represent each of the integers in S' in radix 4.

By construction each digit position from 0 to $|E|$-1 has exactly two 1's among all the integers in S' since each edge has either one or two endpoints in V'.

Since V' has size exactly m, there are exactly m integers in S' which are $\geq 4^{|E|}$.

The integers in S' total to $m4^{|E|} + \sum_{j=0}^{|E|-1} 2 \cdot 4^j = t.$

Now suppose S has a subset S' totals to t.

NP-Completeness (cont)

VC $\leq_P$ SUBSET - SUM

Proof:(cont) Construct a subset of vertices V' :

for each $s_i \in S'$ with $1 \leq i \leq |V|$, add v_i to V'

Since the integers in S' total t and there are no carries possible among the least significant $|E|$ digit positions, for each of these digit positions there must be exactly two integers in S' with 1's in that position.

One of these 1's must be from an $s_i \in S'$ with $1 \leq i \leq |V|$.

For each edge $e_j = (v_a, v_b)$ either s_a or $s_b \in S'$.

For each edge $e_j = (v_a, v_b)$ either v_a or $v_b \in V'$.

NP-Completeness (cont)

VC $\leq_P$ SUBSET - SUM

Proof:(cont) V' is a vertex cover for G.

Since the integers in S' total to t and there are no carries possible among the least significant $|E|$ digit positions, there must be exactly m integers in $S' \geq 4^{|E|}$.

All $s_i \in S$ with $s_i \geq 4^{|E|}$ have $1 \leq i \leq |V|$.

V' is a vertex cover for G of size m.

QED

NP-Completeness (cont)

VC $\leq_P$ SUBSET - SUM

Step 3: Show that f is computable in polynomial time.

If G has a vertices and b edges then $f(G,k)=(S,t)$ and S has $a+b$ integers and each of these integers has $2b+2$ bits.

The integer t can be encoded in less than $2b+\lg a$ bits.

The integers can be constructed in $O(ab)$ time by first building the incidence matrix for G.

f is computable in polynomial time.

Finished **VC $\leq_P$ SUBSET - SUM**

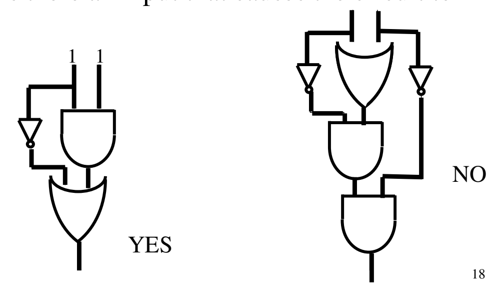
NP-Completeness (cont)

CIRCUIT-SAT

Instance: A boolean combinational circuit with one output, composed of **AND**, **OR**, and **NOT** gates.

Question: Is there an input that causes the circuit to output 1 ?

Examples:



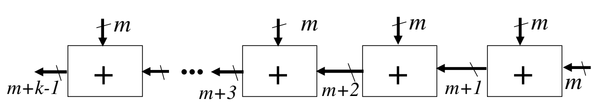
NP-Completeness (cont)

SUBSET - SUM $\leq_P$ CIRCUIT - SAT

Step 1: Construct $f:\{(S,t)\}\rightarrow\{C\}$

Given (S,t) where $S=\{s_1,s_2,\ldots,s_k\}$ and m is the maximum number of bits in any s_i in S and t,

Create a circuit for adding together k m-bit integers.



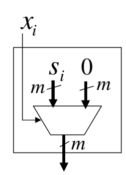
Size $\theta(k(m+k))$

NP-Completeness (cont)

SUBSET - SUM $\leq_P$ CIRCUIT - SAT

For each $s_i \in S$, create a new boolean variable (input) x_i

and a circuit which outputs $\begin{cases} s_i & \text{when } x_i = 1 \\ 0 & \text{when } x_i = 0 \end{cases}$

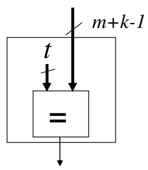


Each has size $\theta(m)$
Total size $\theta(mk)$

NP-Completeness (cont)

SUBSET - SUM $\leq_P$ CIRCUIT - SAT

Create a circuit which compares an *m+k-1* bit integer with *t* and outputs 1 only when they are equal

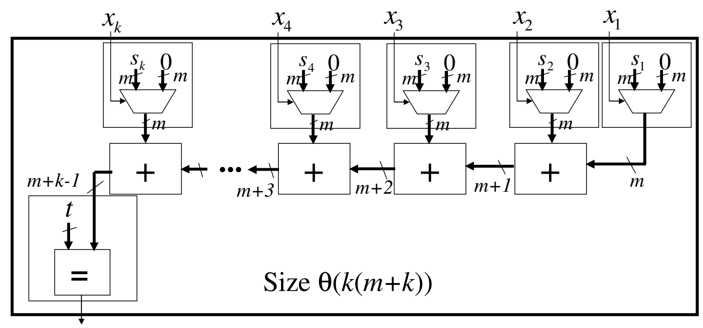


Size $\theta(m+k)$

NP-Completeness (cont)

SUBSET - SUM $\leq_P$ CIRCUIT - SAT

Combine circuits as follows:



NP-Completeness (cont)

SUBSET - SUM $\leq_P$ CIRCUIT - SAT

Step 2: Show that $(S,t) \in L_{\text{SUBSET-SUM}}$

$\Leftrightarrow f(S,t) = C \in L_{\text{CIRCUIT-SAT}}$

S has a subset S' that totals to t $\Leftrightarrow$

there are inputs for C that make its output 1

Proof (sketch)**:** Suppose $S' \subseteq S$ that totals to t.

Set the value of C's inputs as follows: for each $x_i = \begin{cases} 1 & \text{if } s_i \in S' \\ 0 & \text{if } s_i \notin S' \end{cases}$

Since S' totals to t the output of C will be 1.

NP-Completeness (cont)

SUBSET - SUM $\leq_P$ CIRCUIT - SAT

Proof (sketch cont**):** Now suppose $\vec{\mathbf{x}}$ is an input for C which makes its output 1.

Construct S' as follows: let $S' = \{s_i \mid x_i = 1 \text{ in } \vec{\mathbf{x}}\}$

Then S' totals to t since the output of C is only 1 when

$$\sum_{s_i \in S'} s_i = \sum_{i=1}^{k} x_i \cdot s_i = t$$

Step 3: f is computable in polynomial time since the size of C is O(m(k+m)) and it can be constructed in time O(m(k+m)) .

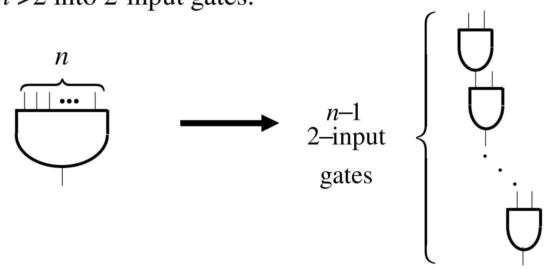
Finished **SUBSET - SUM $\leq_P$ CIRCUIT - SAT**

NP-Completeness (cont)

CIRCUIT - SAT $\leq_P$ 3SAT

Step 1: Construct $f : \{C\} \rightarrow \{\phi\}$

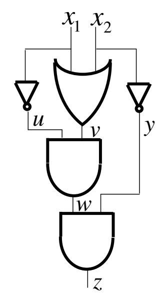
Given a circuit *C*, first break any *n*-input gates for *n* >2 into 2-input gates.



NP-Completeness (cont)

CIRCUIT - SAT $\leq_P$ 3SAT

Create a new variable for each gate output.

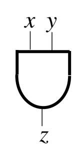


Circuit inputs are already labeled

NP-Completeness (cont)

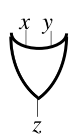
CIRCUIT - SAT $\leq_P$ 3SAT

Create a clause for each gate as follows:



$(x \vee y \vee \bar{z})(x \vee \bar{y} \vee \bar{z})(\bar{x} \vee y \vee \bar{z})(\bar{x} \vee \bar{y} \vee z)$

$(x \vee z)(\bar{x} \vee \bar{z})$



$(x \vee y \vee \bar{z})(x \vee \bar{y} \vee z)(\bar{x} \vee y \vee z)(\bar{x} \vee \bar{y} \vee z)$

Each clause will be 0 for an inconsistent set of values on the gate inputs and output

NP-Completeness (cont)

CIRCUIT - SAT $\leq_P$ 3SAT

Let ϕ be the conjunction of all the clauses created for the gates and *C*'s output.

ϕ is satisfiable $\Leftrightarrow$ *C* is satisfiable.

Final step is to convert all clauses in ϕ to 3-CNF.

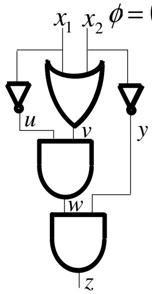
If a clause has only two literals $(x \vee z)$ make a new variable u and replace $(x \vee z)$ by $(x \vee z \vee u)(x \vee z \vee \bar{u})$

For a 1 literal clause (x) make 2 new variables u, v and replace (x) by $(x \vee u \vee v)(x \vee u \vee \bar{v})(x \vee \bar{u} \vee v)(x \vee \bar{u} \vee \bar{v})$

NP-Completeness (cont)

CIRCUIT - SAT $\le_P$ 3SAT

Example:



$$\phi = (x_1 \vee x_2 \vee \bar{v})(x_1 \vee \bar{x}_2 \vee v)(\bar{x}_1 \vee x_2 \vee v)(\bar{x}_1 \vee \bar{x}_2 \vee v)$$

$$(x_1 \vee u \vee c)(x_1 \vee u \vee \bar{c})(\bar{x}_1 \vee \bar{u} \vee c)(\bar{x}_1 \vee \bar{u} \vee \bar{c})$$

$$(x_2 \vee y \vee d)(x_2 \vee y \vee \bar{d})(\bar{x}_2 \vee \bar{y} \vee d)(\bar{x}_2 \vee \bar{y} \vee \bar{d})$$

$$(u \vee v \vee \bar{w})(u \vee \bar{v} \vee \bar{w})(\bar{u} \vee v \vee \bar{w})(\bar{u} \vee \bar{v} \vee w)$$

$$(w \vee y \vee \bar{z})(w \vee \bar{y} \vee \bar{z})(\bar{w} \vee y \vee \bar{z})(\bar{w} \vee \bar{y} \vee z)$$

$$(z \vee a \vee b)(z \vee a \vee \bar{b})(z \vee \bar{a} \vee b)(z \vee \bar{a} \vee \bar{b})$$

NP-Completeness (cont)

CIRCUIT - SAT $\le_P$ 3SAT

The final ϕ in 3-CNF form is satisfiable $\Leftrightarrow$ *C* is satisfiable.

The size of the final ϕ is $O(|C|)$.

All the steps used to construct ϕ are $O(|C|)$, so *f* is polynomial time computable.

Finished **CIRCUIT - SAT $\le_P$ 3SAT**

NP-Completeness (cont)

Summary

- **3SAT $\le_P$ CLIQUE**
- **CLIQUE $\le_P$ VC**
- **VC $\le_P$ SUBSET - SUM**
- **SUBSET - SUM $\le_P$ CIRCUIT - SAT**
- **CIRCUIT - SAT $\le_P$ 3SAT**

If anyone of these problems is in **P**, then they all are.