Recall:

$$\left.\begin{array}{l} L' \in NP \\ L \text{ NP-complete} \\ L \leq_P L' \end{array}\right\} \Rightarrow L' \text{ NP-complete.}$$

At this moment, we know that CIRCUIT-SAT is NP-complete.

We will prove that $\underbrace{\text{3SAT}}_{\text{page 193}}$ is NP-complete.

It is sufficient to show:

* 3SAT $\in$ NP : for a given truth-assignment of the variables, we can verify in polynomial time if the Boolean formula is true.

* CIRCUIT-SAT $\leq_P$ 3SAT.

We need a function f such that

① $f: \underbrace{\text{Boolean circuit } B}_{\text{as on page 210}} \rightarrow \underbrace{\text{Boolean formula } \varphi}_{\text{as on page 192}}$

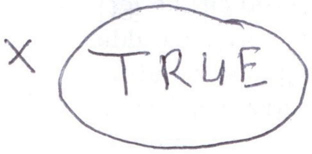
② ∃ truth-values for the unknown input gates such that B's output is true $\iff$ ∃ truth-values for the variables such that φ is true

③ $\varphi = f(B)$ can be computed in time that is polynomial in the length of B.

Consider a Boolean circuit B.

- one variable for each gate.
- describe the effect of each gate using a few clauses.
- connect all clauses by $\wedge$'s.

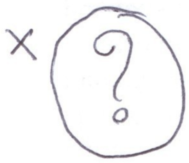
known input gates:



$\longleftrightarrow$ clause: x

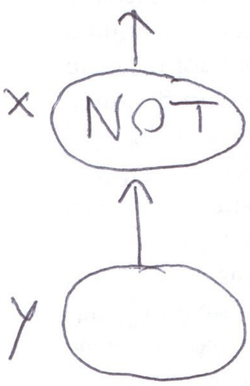
$\longleftrightarrow$ clause: $\neg x$

unknown input gates:



$\longleftrightarrow$ clause: $x \vee \neg x$

NOT-gates:

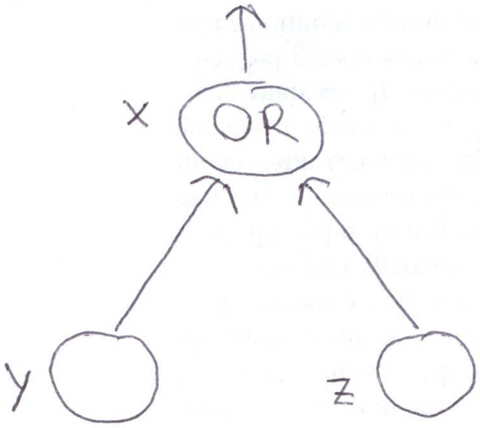


$\longleftrightarrow$ clauses: $(y \Rightarrow \neg x) \wedge (\neg y \Rightarrow x)$

same as

$$(\neg y \vee \neg x) \wedge (y \vee x)$$

OR-gates:



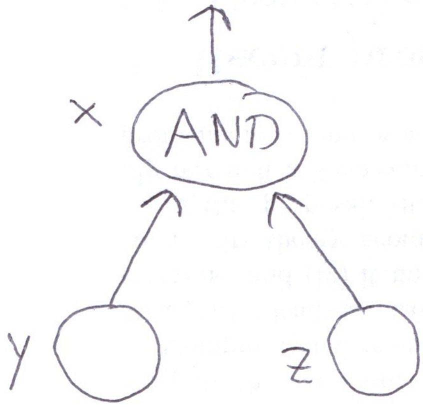
$\longleftrightarrow$ clauses:

$$(x \Rightarrow (y \vee z)) \wedge ((y \vee z) \Rightarrow x)$$

same as

$$(\neg x \vee y \vee z) \wedge (\neg y \vee x) \wedge (\neg z \vee x)$$

AND-gates:



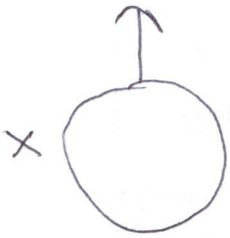
$\longleftrightarrow$ clauses:

$$(x \Rightarrow (y \wedge z)) \wedge ((y \wedge z) \Rightarrow x)$$

same as

$$(\neg x \vee y) \wedge (\neg x \vee z) \wedge (\neg y \vee \neg z \vee x)$$

output-gate:



$\longleftrightarrow$ clause: x

Let φ be the conjunction of all these clauses.

By construction:

$$B \in \text{CIRCUIT-SAT} \iff f(B) = \varphi \in 3SAT.$$

Size of φ:

* number of variables in φ = number of gates in B
* each clause has ≤ 3 literals
* number of clauses ≤ 3 * number of gates in B

$\therefore$ size of φ = O(size of B) : polynomial

φ can be computed in time that is polynomial in the size of B.

<u>Conclusion</u>: 3SAT is NP-complete.

$$\left.\begin{array}{l}\text{3SAT NP-complete}\\ \text{3SAT} \le_{P} \text{INDEP-SET (page 193)}\\ \text{INDEP-SET} \in \text{NP (exercise)}\end{array}\right\} \therefore \begin{array}{l}\text{INDEP-SET}\\ \text{NP-complete.}\end{array}$$

$$\left.\begin{array}{l}\text{CLIQUE} \in \text{NP (exercise)}\\ \text{INDEP-SET} \le_{P} \text{CLIQUE (page 186)}\end{array}\right\} \therefore \begin{array}{l}\text{CLIQUE}\\ \text{NP-complete}\end{array}$$

$$\left.\begin{array}{l}\text{VERTEX-COVER} \in \text{NP (exercise)}\\ \text{CLIQUE} \le_{P} \text{VERTEX-COVER (page 188)}\end{array}\right\} \therefore \begin{array}{l}\text{VERTEX-}\\ \text{COVER}\\ \text{NP-complete.}\end{array}$$

0/1 integer programming

Input: Linear inequalities such as

$$\begin{cases} 3y_1 - 2y_2 - 5y_3 \leq -2 \\ -y_1 + 7y_2 - 2y_3 \leq -3 \end{cases}$$

Question: Are there binary values for the variables such that all inequalities hold?

for the example: $y_1 = 1$, $y_2 = 0$, $y_3 = 1$.

We can write the example as

$$\begin{pmatrix} 3 & -2 & -5 \\ -1 & 7 & -2 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \underset{\substack{\uparrow \\ \text{component-wise}}}{\leq} \begin{pmatrix} -2 \\ -3 \end{pmatrix}$$

$$Ay \leq c$$

- A: $m \times n$ matrix, $m = \#$ inequalities, $n = \#$ variables
- y: binary vector of length n : $y \in \{0,1\}^n$
- c: vector of length m

$0/1$-IP =

$\{(A, c) :$ A is an integer $m \times n$ matrix,
c is an integer column m-vector,
$\exists y \in \{0,1\}^n : Ay \le c\}$.

Exercise: Show that ~~[illegible]~~ $0/1$-IP $\in$ NP.

We will show: 3SAT $\le_P$ $0/1$-IP.

Since 3SAT is NP-complete:
$0/1$-IP is NP-complete.

We need a function f:

① f: input φ for 3SAT $\rightarrow$ input (A, c) for $0/1$-IP

② φ satisfiable $\Leftrightarrow$ $\exists$ binary vector y: $Ay \le c$

③ time to compute (A, c) is polynomial in the length of φ.

$$\varphi = \underbrace{(x_1 \vee \neg x_2 \vee x_3)}_{C_1} \wedge \underbrace{(\neg x_1 \vee x_2 \vee \neg x_3)}_{C_2}$$

We are going to write "φ is satisfiable" as a system of linear inequalities.

We will use variables $y_1, \dots, y_6$:

Correspondence:

$$x_1 \leftrightarrow y_1$$
$$\neg x_1 \leftrightarrow y_2$$
$$x_2 \leftrightarrow y_3$$
$$\neg x_2 \leftrightarrow y_4$$
$$x_3 \leftrightarrow y_5$$
$$\neg x_3 \leftrightarrow y_6$$

Remember: x_1, x_2, x_3: Boolean variables: true or false

$y_1, \dots, y_6$: binary variables: 0 or 1

φ is satisfiable $\iff$

① for $i=1,2,3$: we can give x_i a truth-value

② for $j=1,2$: clause C_j is true.

① x_1 has a truth-value

$\iff \quad y_1 + y_2 = 1$

$\iff \quad y_1 + y_2 \leq 1$ and $y_1 + y_2 \geq 1$

$$\iff \quad \boxed{\begin{array}{l} y_1 + y_2 \leq 1 \\ -y_1 - y_2 \leq -1 \end{array}}$$

x_2 has a truth-value $\iff \boxed{\begin{array}{l} y_3 + y_4 \leq 1 \\ -y_3 - y_4 \leq -1 \end{array}}$

x_3 has a truth-value $\iff \boxed{\begin{array}{l} y_5 + y_6 \leq 1 \\ -y_5 - y_6 \leq -1 \end{array}}$

2. $C_1 = x_1 \vee \neg x_2 \vee x_3$ is true

$\Leftrightarrow$ at least one of $x_1, \neg x_2, x_3$ is true

$\Leftrightarrow \quad y_1 + y_4 + y_5 \geq 1$

$\Leftrightarrow \quad \boxed{-y_1 - y_4 - y_5 \leq -1}$

$C_2 = \neg x_1 \vee x_2 \vee \neg x_3$ is true

$\Leftrightarrow \quad \boxed{-y_2 - y_3 - y_6 \leq -1}$

Thus: φ is mapped to

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 & -1 & 0 \\ 0 & -1 & -1 & 0 & 0 & -1 \end{pmatrix}, \quad c = \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \\ -1 \\ -1 \end{pmatrix}$$

In general:

Boolean variables $x_1, \dots, x_n$

$\varphi = C_1 \wedge \dots \wedge C_m$, $C_i = l_i^1 \vee l_i^2 \vee l_i^3$,

each l_i^j is either x_k or $\neg x_k$ for some k.

Our linear inequalities will use binary variables $y_1, \dots, y_{2n}$.

A will be a $(2n+m) \times 2n$ matrix.

c will be a $(2n+m)$-vector.

$$c = \begin{array}{l} \left(\begin{array}{c} 1 \\ -1 \\ 1 \\ -1 \\ \vdots \\ 1 \\ -1 \\ -1 \\ -1 \\ \vdots \\ -1 \end{array}\right) \begin{array}{l} \left.\vphantom{\begin{array}{c} 1 \\ -1 \\ 1 \\ -1 \\ \vdots \\ 1 \\ -1 \end{array}}\right\} 2n \\ \left.\vphantom{\begin{array}{c} -1 \\ -1 \\ \vdots \\ -1 \end{array}}\right\} m \end{array} \end{array}$$

for $i = 1, \dots, n$:

row $2i-1$ of A : $\underbrace{0 \text{———} 0}_{2(i-1)}\ 1\ 1\ \underbrace{0 \text{———} 0}_{2(n-i)}$

row $2i$ of A : $0 \text{———} 0\ {-1}\ {-1}\ 0 \text{———} 0$

for $i = 1, \dots, m$: what is row $2n+i$ of A:

for $j = 1, 2, 3$:

if $\ell_i^j = x_k$: entry in column $2k-1$ is -1

if $\ell_i^j = \neg x_k$: entry in column $2k$ is -1

all other entries are 0

This defines the function $f : \varphi \to (A, c)$.

By construction:

$$\varphi \text{ satisfiable} \iff \exists\, y \in \{0,1\}^{2n} : Ay \le c.$$

Time to construct (A, c):

$$O((n+m)n) = O((n+m)^2)$$

$$= O((\text{length of } \varphi)^2) \quad \text{polynomial}$$