RECAP

- COOK'S THEOREM: BCSAT $\in$ NPH
  - "simulate" $V_L(x, y)$ with $C_{V_L, x}(y)$

A certificate y_x for $x \in L$ becomes a satisfying assignment for $f(x) = \langle C_{V_L, x}(y) \rangle$

- FORMULA SATISFIABILITY (SAT):

  BC-SAT $\leq_p$ SAT:
  - Trivial reduction based on subexpressions not ptc (poly-time computable)

We need a more sophisticated approach!

$$C = (I \cup G \cup \{0\}, W)$$

x_i (for I); $y_g \ \forall g \in G$ (for G); y_0 (for $\{0\}$)

New variables

$y_g : g \in G$

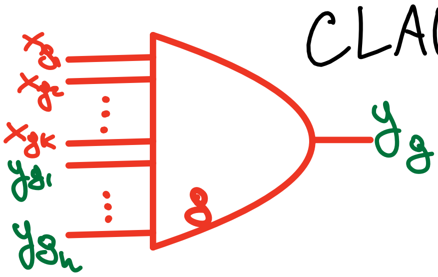
We add extra variables y_g associated to the gates, describing the output of each gate $g \in G$ and create a formula that reflects valid configurations of the circuit

RECALL: A configuration is the set of all bits on the wires under a valid switching of the circuit: $(\vec{s}_1, \vec{s}_2)$

$\vec{x}$ (for $\vec{s}_1$); $\vec{y}$ (gate outputs) (for $\vec{s}_2$)

For each gate $g \in G$ we introduce a sub-formula $\boxed{\Gamma_g}$ describing the input-output relation of the gate:

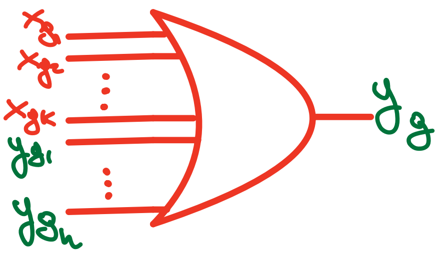
CLAUSE



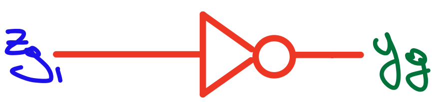
"equation"

$$\Gamma_g : y_g \leftrightarrow x_{g_1} \wedge \dots \wedge x_{g_k} \wedge y_{g_1} \wedge \dots \wedge y_{g_h}$$

input di circuito | output di altri gate



$$\Gamma_g : y_g \leftrightarrow x_{g_1} \vee \dots \vee x_{g_k} \vee y_{g_1} \vee \dots \vee y_{g_h}$$



$$\Gamma_g : y_g \leftrightarrow \neg (z_{g_1})$$

Observe that subexpression Γ_g is true under a truth assignment $\Leftrightarrow$ y_g is assigned the correct output value w.r.t. the values assigned to the gate's inputs

We can then obtain the formula

$$\boxed{\psi(\vec{x}, \vec{y}) = \bigwedge_{g \in G} \Gamma_g}$$

Formula ψ describes a configuration of the circuit C:

$\psi(\vec{b}_1, \vec{b}_2) = 1$ $\Leftrightarrow$ the values of the outputs of the gates of $C(\vec{b}_1)$ are $(b_2)_g : g \in G$

$\vec{x}$ $\quad$ $\vec{y}$

$\Rightarrow$ $(\vec{b}_1, \vec{b}_2)$ is a valid configuration non basta ψ per riduzione ⇒ formula sempre satisfiable

OBSERVATION: $\psi(\vec{x}, \vec{y})$ is always satisfiable (HOMEWORK: how many satisfying assignments?)

We have that $\boxed{|\langle \psi(\vec{x},\vec{y})\rangle| = \Theta(|\langle C(\vec{x})\rangle|)}$

since each subexpression uses a number of symbols linear in the size of the corresponding gate!

The reduction function f uses ψ as follows:

$$\boxed{f(\langle C(\vec{x})\rangle) = \phi_c(\vec{x},\vec{y}) = \psi(\vec{x},\vec{y}) \wedge y_0}$$

variable associated to gate of output wire

$$\boxed{|\langle \phi_c(\vec{x},\vec{y})\rangle| = \Theta(|\langle\psi(\vec{x},\vec{y})\rangle|) = \Theta(|\langle C(x)\rangle|)}$$

thus f is ptc

We are left to prove that

$$\boxed{x \in L_1 \Leftrightarrow f(x) \in L_2}$$

$$\begin{cases} x \in L_1 \Rightarrow f(x) \in L_2 \\ f(x) \in L_2 \Rightarrow x \in L_1 \end{cases}$$

OR difficoltà dipende dal caso

$$\begin{cases} x \in L_1 \Rightarrow f(x) \in L_2 \\ x \notin L_1 \Rightarrow f(x) \notin L_2 \end{cases}$$

$$\boxed{\langle C(\vec{x}) \rangle \in \text{BC-SAT} \Leftrightarrow f(\langle C(\vec{x}) \rangle) = \langle \phi_C(\vec{x}, \vec{y}) \rangle \in \text{SAT}}$$

$\Rightarrow$ If $\boxed{\langle C(\vec{x}) \rangle \in \text{BC-SAT}}$ then $\boxed{\exists \vec{b}_1 \in \{0,1\}^n : C(\vec{b}_1) = 1}$

Consider the configuration of the circuit under $\vec{b}_1$, and let $\vec{b}_2$ be the values carried by the wires at the outputs of the gates

We have $\boxed{\psi(\vec{b}_1, \vec{b}_2) = 1}$

Moreover, since $\boxed{C(\vec{b}_1) = 1}$ we have $\boxed{b_0 = 1}$ y_0

$$\Rightarrow \boxed{\phi_C(\vec{b}_1, \vec{b}_2) = \psi(\vec{b}_1, \vec{b}_2) \wedge b_0 = 1} \Rightarrow \boxed{f(\langle C(\vec{x}) \rangle) \in \text{SAT}}$$

$\Leftarrow$ $\boxed{f(\langle C(\vec{x}) \rangle) \in \text{SAT} \Rightarrow \exists \vec{b}_1, \vec{b}_2 : \phi_C(\vec{b}_1, \vec{b}_2) = 1}$

operandi di ϕ_C

$\Rightarrow$ 1. $\boxed{\psi(\vec{b}_1, \vec{b}_2) = 1}$: the truth assignment represent a legal configuration $(\vec{b}_1, \vec{b}_2)$ of the circuit C

$\Rightarrow$ 2. Under $\boxed{(\vec{b}_1, \vec{b}_2) : b_0 = 1}$ $\Rightarrow$ $\boxed{C(\vec{b}_1) = 1}$

Thus $\boxed{\langle C(\vec{x}) \rangle \in \text{BC-SAT}}$

LESSON LEARNED: We have to make sure that the reduction is poly-time computable!

SIMILAR PITFALL: $f_{L' \to L}$ cannot possibly know whether $x \in L'$ or $x \notin L'$ unless $L' \in P$!

non si può fare riduzioni diverse in base a $\in L$

This is never the case if you are trying to prove that $L \in NPC$

IMPORTANT

Next reduction is a special case of a large family: $\boxed{\text{reductions by restriction}}$

These reductions are used to prove that an NPC problem stays NPC even if we restrict the set of admissible instances

DEF Given a set of boolean variables $\{x_1, x_2, \dots, x_n\}$:

- A $\boxed{\text{literal}}$ is either a variable x_i or its negation $\neg x_j$ ($\bar{x}_j$)

We will use y to denote a literal

- A $\boxed{\text{clause}}$ is a disjunction of 3 distinct literals, e.g. $C_i = x_1 \vee \bar{x}_3 \vee x_5$

- A $\boxed{\text{3-CNF formula } \phi(x_1, \dots, x_n)}$ is a conjunction of clauses built on the n variables

e.g. $\phi(x_1, x_2, x_3, x_4) = (x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_3)$

3-CNF-SAT:

$$\begin{cases} \text{I}: \langle \phi(x_1, x_2, \dots, x_n) = \bigwedge_{i=1}^{m} C_i \rangle, \\ \quad \phi \text{ 3-CNF formula} \\ \text{Q}: \text{Is } \phi(\bar{x}) \text{ satisfiable?} \end{cases}$$

We will prove that $\boxed{\text{3-CNF-SAT} \in \text{NPC}}$.

Meaning:

Even restricting the shape of the formula to a very simple one, the problem stays difficult!

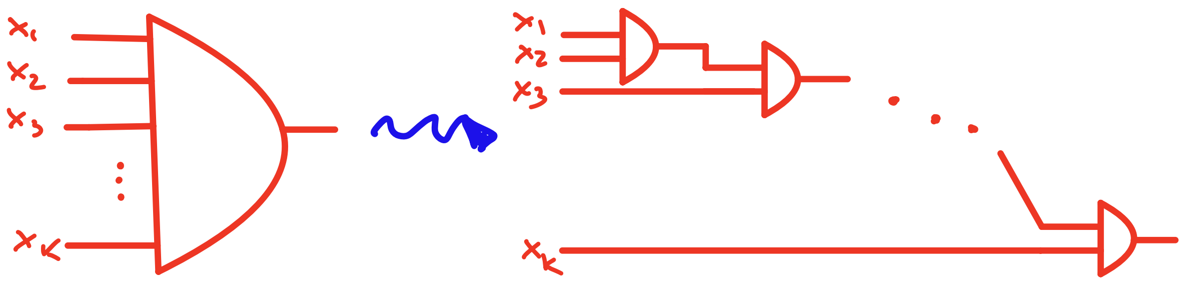
3-CNF-SAT $\in$ NP (trivial, exercise)

3-CNF-SAT $\in$ NPH. può essere difficile scrivere formula in CNF

We reduce again from BC-SAT, using a similar idea to BC-SAT $\leq_p$ SAT

- PREPROCESSING STEP: We modify C to get an equivalent one:

Replace each AND or OR gate with fan-in ≥ 3 with a cascade of gates with fan-in 2, e.g.,



The resulting circuit C' is equivalent to C. Moreover $\boxed{|\langle C'\rangle| = \Theta(|\langle C\rangle|)}$
(# new gates $\leq$ # wires of C)

- Next, we write the formula $\psi(\vec{x},\vec{y}) \wedge y_0$ as done for formula satisfiability. We have four types of subexpressions Γ:

$$y_i \longleftrightarrow w_i^1 \vee w_i^2 \qquad y_i \longleftrightarrow w_i^1 \wedge w_i^2$$

$$y_i \longleftrightarrow \overline{w_i} \quad (w_i \text{ either } x_k \text{ or } y_j) \qquad y_0$$

- We can re-write each subexpression as a conjunction of disjunctions:

$$y_i \longleftrightarrow w_i^1 \vee w_i^2 \equiv (y_i \vee \overline{w_i^1}) \wedge (y_i \vee \overline{w_i^2}) \wedge (\overline{y_i} \vee w_i^1 \vee w_i^2)$$

$$y_i \longleftrightarrow w_i^1 \wedge w_i^2 \equiv (\overline{y_i} \vee w_i^1) \wedge (\overline{y_i} \vee w_i^2) \wedge (y_i \vee \overline{w_i^1} \vee \overline{w_i^2})$$

$$y_i \longleftrightarrow \overline{w_i} \equiv (y_i \vee w_i) \wedge (\overline{y_i} \vee \overline{w_i})$$

Equivalence verified via truth tables (HOMEWORK)

e.g.,

$y_i w_i$	$y_i \leftrightarrow \overline{w_i}$	A: $y_i \vee w_i$	B: $\overline{y_i} \vee \overline{w_i}$	$A \wedge B$
0 0	0	0	1	0
0 1	1	1	1	1
1 0	1	1	1	1
1 1	0	1	0	0

=

- After this transformation, $\psi(x,y) \wedge y_0$ has been rewritten as a conjunction of disjunctive clauses. However, some of the clauses have < 3 literals!

- We add "dummy" variables to bring all clauses to 3 literals:

$$\boxed{\begin{aligned}(y \vee w) &= (y \vee w) \vee 0 = (y \vee w) \vee (z \wedge \overline{z}) = \\ &= (y \vee w \vee z) \wedge (y \vee w \vee \overline{z})\end{aligned}}$$

(For clause y_0 repeat twice)

- We have obtained a 3-CNF-FORMULA $\phi_{C'}(\vec{x},\vec{y},\vec{z})$ equivalent to $\psi_{C'}(\vec{x},\vec{y}) \wedge y_0$

$$\boxed{\begin{aligned}&C \in \text{BC-SAT} \Leftrightarrow C' \in \text{BC-SAT} \Leftrightarrow \\ &\psi_{C'}(\vec{x},\vec{y}) \wedge y_0 \in \text{SAT} \Leftrightarrow \phi_{C'}(\vec{x},\vec{y},\vec{z}) \in \text{3-CNF SAT}\end{aligned}}$$

LESSON LEARNED: Reductions by restriction must "modify" f so that it maps into the set of restricted instances.

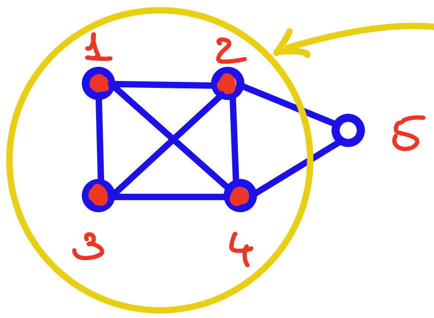
PROBLEMS ON GRAPHS

We will prove the NP-Completeness of three decision problems on graphs, related to very important optimization problems.

DEF Given $G=(V,E)$ undirected, a $\boxed{\text{clique}}$ of size K is a subset $V' \subseteq V$ of K nodes such that each pair of distinct nodes in V' is connected by an edge (clique = complete subgraph)

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

EXAMPLE:



$\{1,2,3,4\}$ is a clique of size 4

We have:

CLIQUE

$$\begin{cases} I: \langle G=(V,E), K\rangle, \ G \text{ undirected graph}, \ 1 \leq K \leq |V| \\ Q: \text{Does } G \text{ contain a clique of size } K? \end{cases}$$

The optimization version of this problem requires determining the clique of max size.

QUESTION: Why doesn't the decision problem ask "size $\geq k$"? [Argue that the two questions are equivalent]

Important applications in bioinformatics, computational chemistry, social network analysis

Let's prove that CLIQUE $\in$ NPC

- 1. CLIQUE $\in$ NP trivial. Candidate certificate: set of k nodes.
  [Write $V_{CLIQUE}(x, y)$ as an exercise]
- 2. CLIQUE $\in$ NPH
  Reduction from 3-CNF-SAT.

$$\underset{\text{known NPH}}{\text{3-CNF-SAT}} \leq_p \underset{\text{candidate}}{\text{CLIQUE}}$$

We start from $\phi(x_1, x_2, \dots, x_n) = C_1 \wedge C_2 \wedge \dots \wedge C_m$
with $C_i = y_1^i \vee y_2^i \vee y_3^i$, $y_j^i \in \{x_k, \bar{x}_k : 1 \leq k \leq n\}$

We have to define a reduction function that takes ϕ and returns an instance of CLIQUE in poly-time:

$$f(\langle \phi(x_1, \dots, x_n) \rangle) = \langle G_\phi, k_\phi \rangle$$

1. We set $k_\phi = m$ (# of clauses of ϕ)

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

V_ϕ:

(one node for each literal y^i_j $\begin{smallmatrix} 1\le i\le m \\ 1\le j\le 3 \end{smallmatrix}$):

$$V_\phi = \{\sigma^i_j : 1\le i\le m,\ 1\le j\le 3\} \Rightarrow |V_\phi| = 3m$$

E_ϕ:

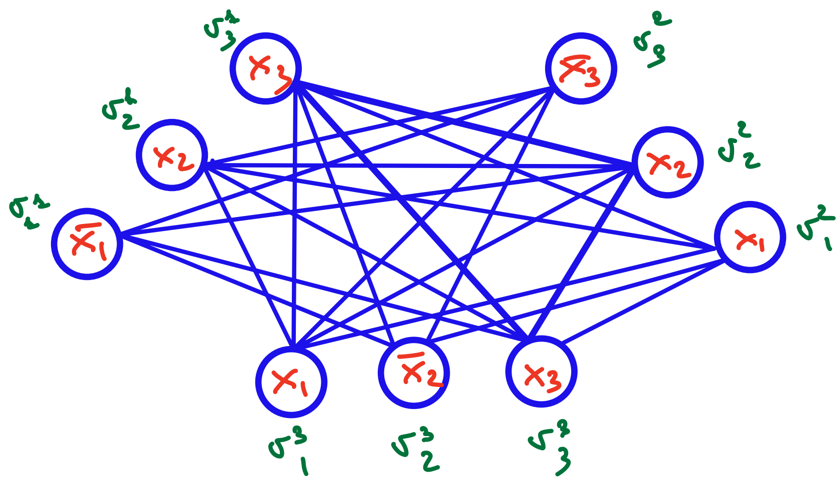
$$\{\sigma^{i_1}_{j_1}, \sigma^{i_2}_{j_2}\} \in E_\phi \Leftrightarrow (i_1 \ne i_2) \wedge (y^{i_1}_{j_1} \ne \neg y^{i_2}_{j_2})$$

There is an edge between two nodes if their corresponding literals belong to distinct clauses and are not the negation of each other (e.g. $y^{i_1}_{j_1} = x_1,\ y^{i_2}_{j_2} = \bar{x}_1$)

INTUITION: Two literals corresponding to an edge can be made both true under the same truth assignment

e.g. $y^{i_1}_{j_1} = x_3$ —— $y^{i_2}_{j_2} = \bar{x}_5$ both true if $b_3 = 1,\ b_5 = 0$

EXAMPLE: $\phi(x_1, x_2, x_3) = (\bar{x}_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \bar{x}_3) \wedge (x_1 \vee \bar{x}_2 \vee x_3)$



Running time: $\Theta(n^2) = \Theta(|\langle\phi\rangle|^2)$

Each node is compared to all other nodes to determine the edges

$\Rightarrow f$ is ptc