

## RECAP

- COOK'S THEOREM : BCSAT  $\in$  NPH
  - "simulate"  $V_L(x, y)$  with  $C_{V_L, x}(y)$
- A certificate  $y$  for  $x \in L$  becomes a satisfying assignment for  $f(x) = \langle C_{V_L, x}(y) \rangle$
- FORMULA SATISFIABILITY (SAT) :
  - $SC\text{-SAT} \leq_p SAT$  :
  - Trivial reduction based on subexpressions not PTC (poly-time computable)

We need a more sophisticated approach!

$$C = (I \cup G \cup \{y_0\}, W)$$

Arrows point from  $x_i$  to  $y_g$  for  $g \in G$ , and from  $y_g$  to  $y_0$ . A yellow arrow points from the set  $G$  to the variable  $y_0$ .

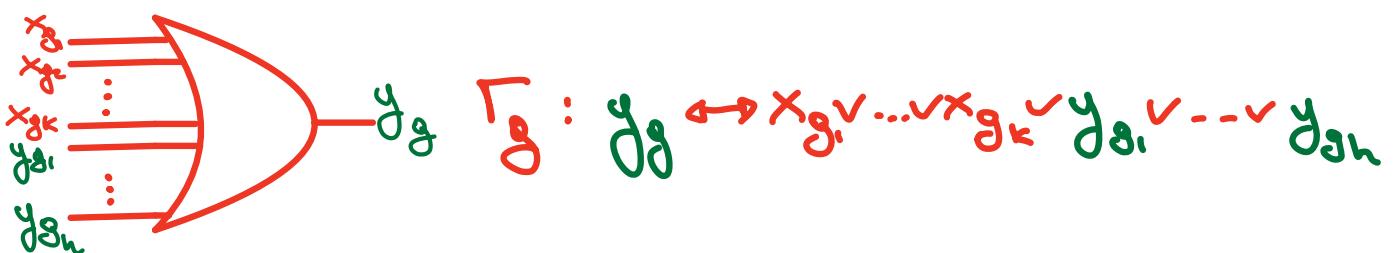
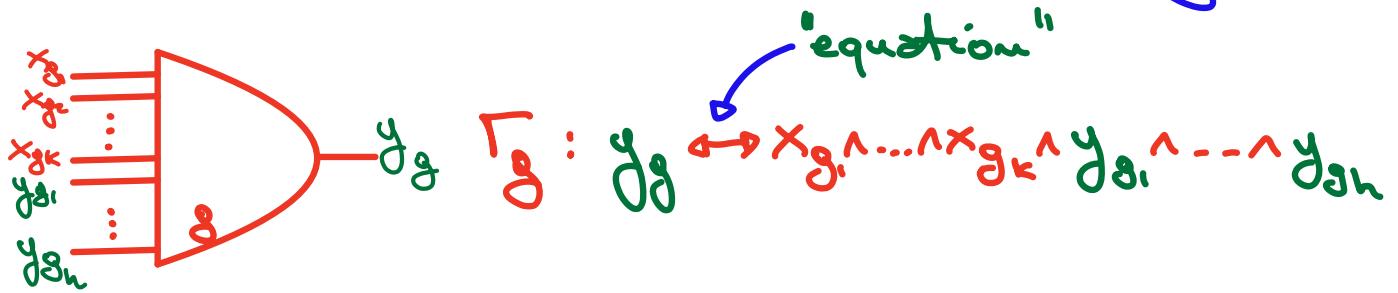
New variables  
 $y_g : g \in G$

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 :  $(\vec{S}_1, \vec{S}_2)$

Arrows point from  $x$  to  $\vec{S}_1$  and from  $y_g$  to  $\vec{S}_2$ . A yellow bracket groups  $\vec{S}_1$  and  $\vec{S}_2$  with the label "(gate outputs)".

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



Observe that subexpression  $\Gamma_g$  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

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

Formula  $\Psi$  describes a configuration of the circuit  $C$ :

$$\Psi(\vec{b}_1, \vec{b}_2) = 1 \Leftrightarrow \begin{array}{l} \text{the values at the} \\ \text{outputs of the gates} \\ \text{of } C(\vec{b}_1) \text{ are} \\ (b_2)_g : g \in G \end{array}$$

$\Leftrightarrow (\vec{b}_1, \vec{b}_2)$  is a valid configuration

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

We have that  $|<\psi(x, \bar{y})>| = \Theta(|<CC(x)>|)$

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

4

The reduction function  $f$  uses  $\psi$  as follows:

$$f(<CC(x)>) = \phi_c(x, \bar{y}) = \underline{\psi(x, \bar{y}) \wedge y_0}$$

variable associated to  
gate of output wire

$$|<\phi_c(x, \bar{y})>| = \Theta(|<\psi(x, \bar{y})>|) = \Theta(|<CC(x)>|)$$

thus  $f$  is ptc

We are left to prove that

$$x \in L_1 \Leftrightarrow f(x) \in L_2$$

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

OR

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

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

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

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

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

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

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

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

$\Rightarrow$  1.  $\psi(\vec{b}_1, \vec{b}_2) = 1$  : the truth assignments represent a legal configuration  $(S_1, B_2)$  of the circuit  $C$

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

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

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

**SIMILAR PITFALL:**  $f_{L \leq L'}$  cannot possibly know whether  $x \in L'$  or  $x \notin L'$  unless  $L \in P$ !  
This is never the case if you are trying to prove that  $L \in \text{NPC}$

### IMPORTANT

Next reduction is a special case of a large family: 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 literal is either a variable  $x_i$  or its negation  $\neg x_i$  ( $\bar{x}_i$ )

We will use  $y$  to denote a literal

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

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

We will prove that 3-CNF-SAT  $\in$  NP.

Meaning:

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

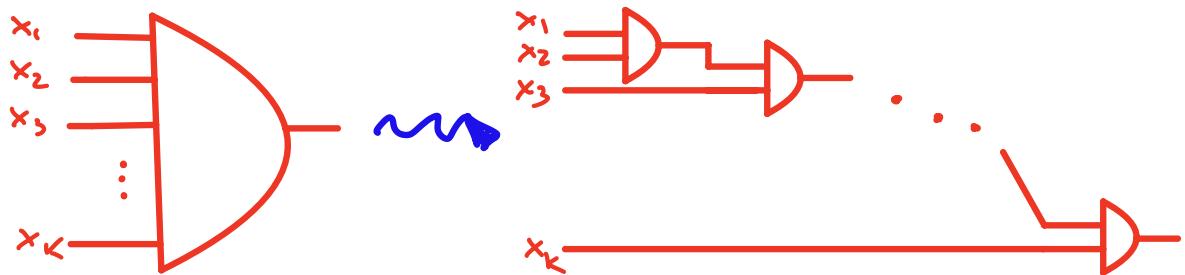
3-CNF-SAT  $\in$  NP (trivial, exercise)

3-CNF-SAT  $\in$  NPH.

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

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

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



The resulting circuit  $C'$  is equivalent to  $C$ . Moreover  $|<C'|| = \Theta(|<C||)$  ( $\#$  new gates  $\leq \#$  wires of  $C$ )

- Next, we write the formulae  $\psi(x, y) \wedge y_0$  as done for formulae satisfiability. We have four types of subexpressions  $\Gamma$ :

$$\begin{array}{ll} y_i \leftarrow w_i^1 \vee w_i^2 & y_i \leftarrow w_i^1 \wedge w_i^2 \\ \text{or } w_i \text{ either } & \\ y_i \leftarrow \bar{w}_i \wedge (x_k \text{ or } y_j) & y_0 \end{array}$$

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

$$y_i \leftarrow w_i^1 \vee w_i^2 \equiv (y_i \vee \bar{w}_i^1) \wedge (y_i \vee \bar{w}_i^2) \wedge (\bar{y}_i \vee w_i^1 \vee w_i^2)$$

$$y_i \leftarrow w_i^1 \wedge w_i^2 \equiv (\bar{y}_i \vee w_i^1) \wedge (\bar{y}_i \vee w_i^2) \wedge (y_i \vee \bar{w}_i^1 \vee \bar{w}_i^2)$$

$$y_i \leftarrow \bar{w}_i \equiv (y_i \vee w_i) \wedge (\bar{y}_i \vee \bar{w}_i)$$

Equivalence verified via truth tables (Homework)

e.g. 1

$y_i w_i$	$y_i \leftrightarrow \bar{w}_i$	A: $y_i v w_i$	B: $\bar{y}_i v \bar{w}_i$	$A \wedge B$
00	0	0	1	0
01	1	1	1	1
10	1	1	0	1
11	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:

$$\begin{aligned}
 (y v w) &= (y v w) v 0 = (y v w) v (z \wedge \bar{z}) = \\
 &= (y v w v z) \wedge (y v w v \bar{z})
 \end{aligned}$$

(For clause  $y_0$  repeat twice)

- We have obtained a 3-CNF-FORMULA  $\phi_C(x, y, z)$  equivalent to  $\psi(x, y) \wedge y_0$

$C \in \text{3C-SAT} \Leftrightarrow C' \in \text{3C-SAT} \Leftrightarrow$

$$\psi_{C'}(x, y) \wedge y_0 \in \text{SAT} \Leftrightarrow \phi_{C'}(x, y, z) \in \text{3CNF-SAT}$$

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

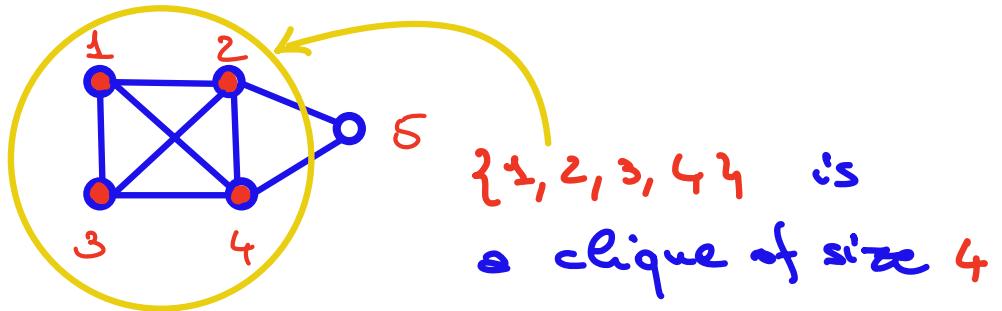
## PROBLEMS ON GRAPHS

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 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)

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

**EXAMPLE :**



We have:

**CLIQUE**

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

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 ENP

-1. CLIQUE ENP trivial. Candidate certificate: set of  $K$  nodes.

[Write  $V_{\text{CLIQUE}}(x, y)$  as an exercise]

-2. CLIQUE ENPH

Reduction from 3-CNF-SAT.

3-CNF-SAT  $\leq_p$  CLIQUE

Known NPH

candidate

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  $\phi$  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  $\phi$ )

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

$\vee\phi$ :

(one node for each literal  $y_j^i$   $i \in \{1, 2, 3\}$   $1 \leq j \leq 3$ ):

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

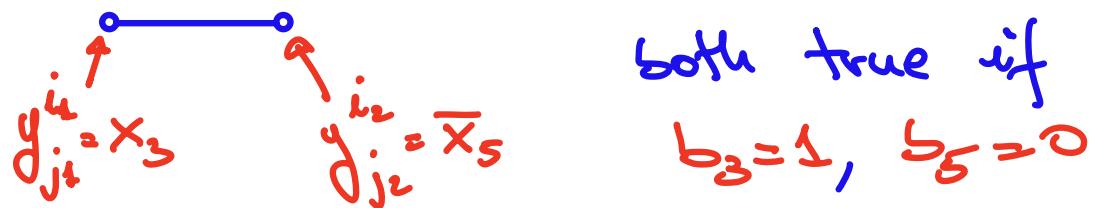
$E\phi$ :

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

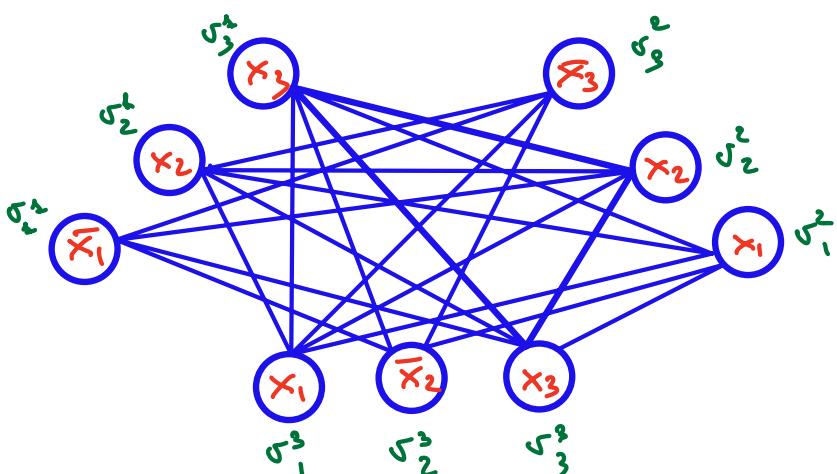
There is an edge between two nodes if their corresponding literals belong to distinct clauses and are not the negation of each other (e.g.  $y_{j_1}^{i_1} = x_1, y_{j_2}^{i_2} = \bar{x}_1$ )

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

e.g.



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:  $\Theta(n^2) = \Theta(|\phi|^2)$

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

$\Rightarrow f$  is ptc