Lecture 20 - NP-Complete Problems

Nov 12, 2020

Recap

A language L is a subset of $\{0,1\}^*$.

A relation R is a subset of $\{0,1\}^* \times \{0,1\}^*$ (0101 , 1010)

$L(R) = \{x : \exists y \text{ s.t. } (x,y) \in R\}$.

$P = \{L :$ Deciding whether $x \in L$ can be done in polynomial time$\}$

$NP = \{L(R) :$ R is an efficiently verifiable relation.$\}$

Given (x,y) we can check in poly$(|x|)$ time that $(x,y) \in R$.

Def'n:

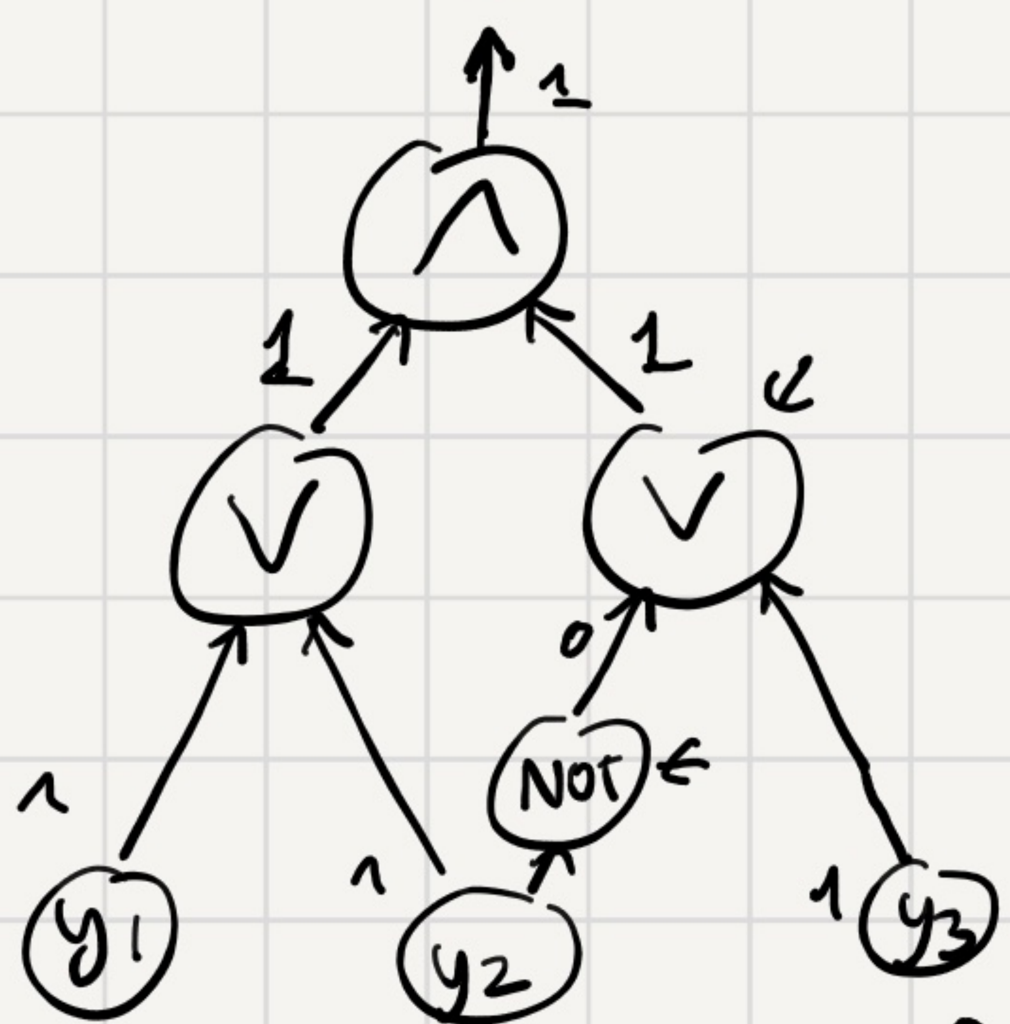
- A languge L is NP-Hard if $\forall M \in NP$ $M \to L$. "$M \leq L$"

- A language L is NP-complete if:
  1. L is NP-Hard
  2. $L \in NP$.

We "showed" that CSAT is NP-complete.

Def'n: A circuit is a directed acyclic graph with input nodes marked by $y_1, \dots, y_m$ & gates of three types: AND, OR, NOT. & one output.

Example:



The circuit computes the Boolean expression: $(y_1 \vee y_2) \wedge (\overline{y_2} \vee y_3)$.

size = # of gates

Def'n: CSAT (circuit satisfiability) Given circuit C on inputs $y_1, \dots, y_m$

Decide whether $\exists$ assignment to input variables s.t. the circuit accepts (i.e. $\exists y \in \{0,1\}^m$ s.t. $C(y) = 1$)

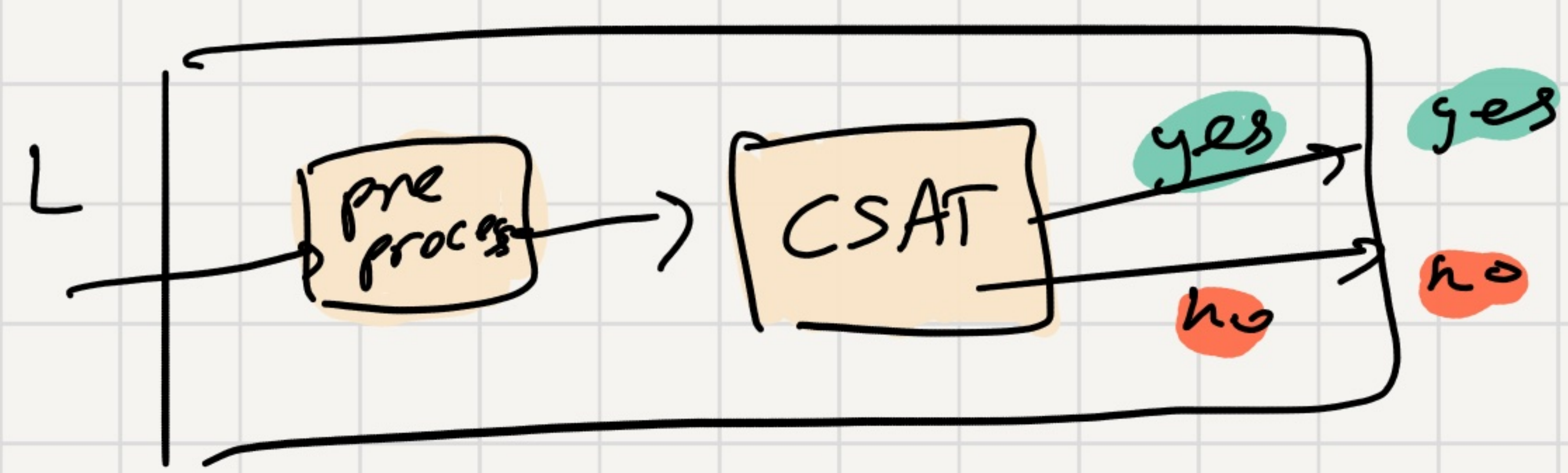
Thm: [Cook-Levin]

CSAT is NP-complete.

Corollary: $\mathrm{CSAT} \in P \iff P = NP$.

Proof:

- $P = NP \Rightarrow$ since $\mathrm{CSAT} \in NP \Rightarrow \mathrm{CSAT} \in P$
- $\mathrm{CSAT} \in P \Rightarrow$ since $\forall L \in NP \quad L \to \mathrm{CSAT} \Rightarrow \forall L \in NP : L \in P \Rightarrow P = NP$.



REDUCIBILITY AMONG COMBINATORIAL PROBLEMS[†]

1972

Richard M. Karp

University of California at Berkeley

We next introduce a concept of reducibility which is of central importance in this paper.

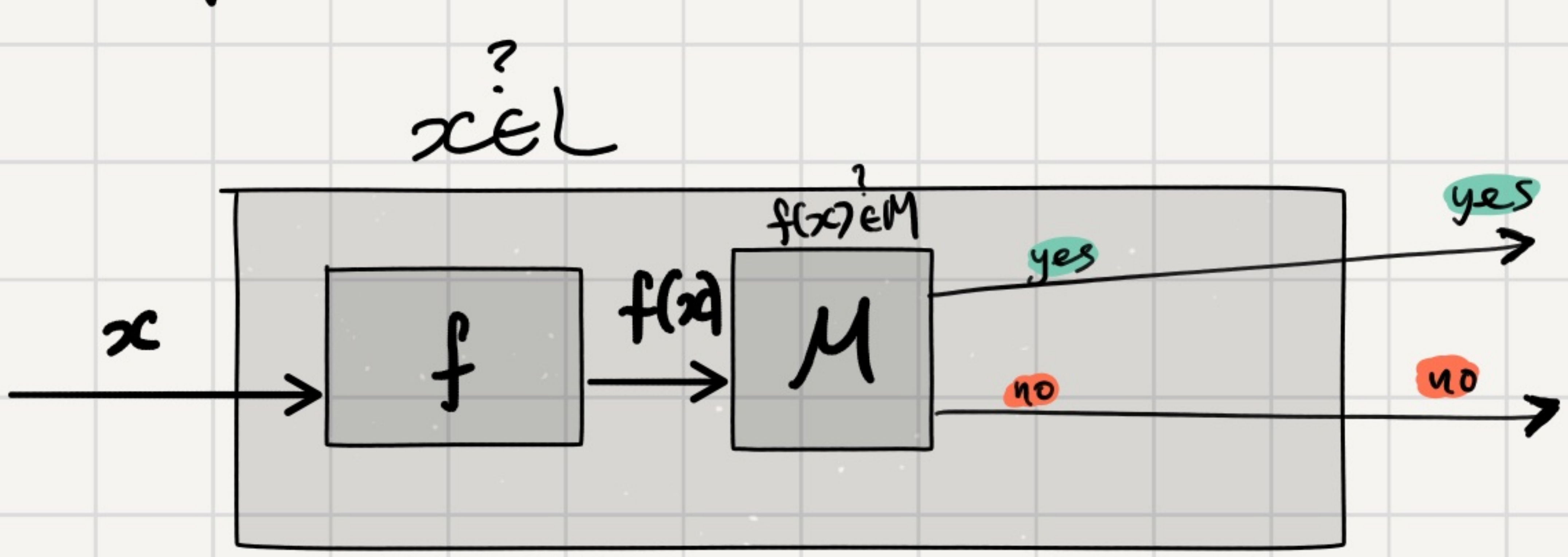
Definition 3. Let L and M be languages. Then $L \propto M$ (L is reducible to M) if there is a function $f \in \Pi$ such that $f(x) \in M \Leftrightarrow x \in L$.

"many-to-one reduction"

"Karp Reduction"

pre-processing

Π = computable in polynomial-time.



$x \in L \Leftrightarrow f(x) \in M.$

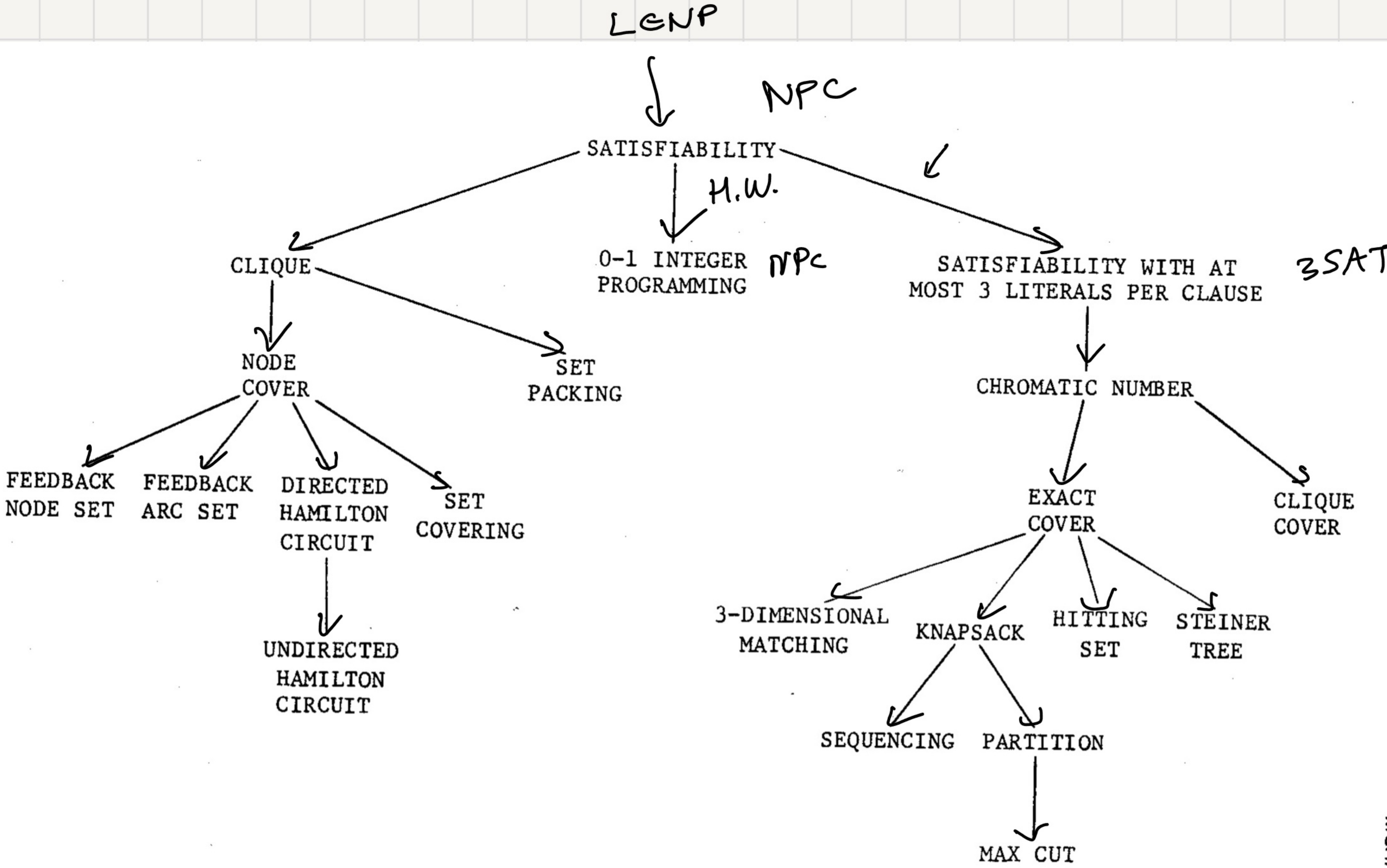


FIGURE 1 - Complete Problems

CSAT → 3SAT

Def'n: 3SAT:

Given a CNF formula ϕ on input $x_1, \dots, x_n$ where each clause involves at most 3 literals

$$\underbrace{(x_1 \vee x_7 \vee x_{15})}_{\text{clause}} \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_3) \wedge (x_3 \vee x_4 \vee \bar{x}_9) \wedge \dots \wedge (\quad)$$

Decide whether $\exists x \in \{0,1\}^n$ that satisfies ϕ.

3SAT → CSAT. easy

CSAT → 3SAT.

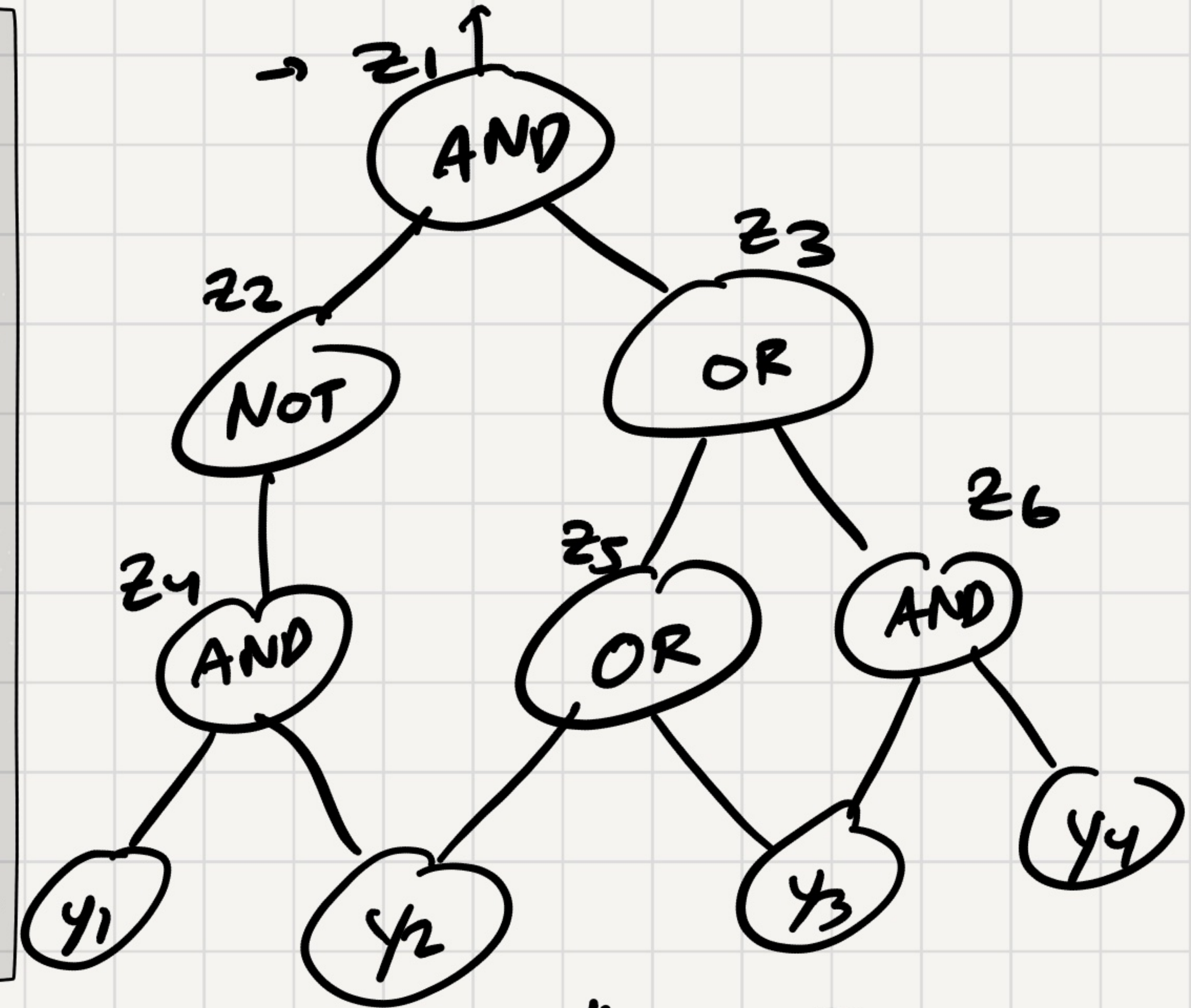
Example:

$$\phi = (x_1 \vee x_2 \vee x_3) \wedge \underline{(x_1 \vee \bar{x}_2)} \wedge (\underline{x_2} \vee \bar{x}_3) \wedge (x_3 \vee \bar{x}_1) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3).$$

$x_1 = 0 \to x_2 = 0 \to x_3 = 0$ reject.

$x_1 = 1 \to x_3 = 1 \to x_2 = 1$ reject.

Given a circuit, we introduce "helper" variables $z_1, z_2, \dots$ one per gate and we express the satisfiability of the circuit as the AND of many constraints, each involving ≤ 3 vars.



$$z_1 \wedge (\underbrace{\text{"}z_1 = z_2 \text{ AND } z_3\text{"}}) \wedge (\text{"}z_2 = \overline{z_4}\text{"}) \wedge (\text{"}z_4 = y_1 \text{ AND } y_2\text{"})$$

$$\wedge \dots$$

$$\underbrace{\text{"}z_2 = \overline{z_4}\text{"}} \equiv (\underbrace{z_2 \vee z_4}) \wedge (\underbrace{\overline{z_2} \vee \overline{z_4}})$$

$$g = h_1 \vee h_2 \equiv (\underbrace{\overline{g} \vee (h_1 \vee h_2)}) \wedge (\underbrace{g \vee \overline{(h_1 \vee h_2)}})$$

$$= (\overline{g} \vee h_1 \vee h_2) \wedge (g \vee \overline{h_1}) \wedge (g \vee \overline{h_2})$$

Claim: any constraint on 3 vars can be expressed as a 3CNF.

a	b	c	$f(a,b,c)$	f
0	0	0	0	$(a \vee b \vee c) \wedge$
0	0	1	0	$(a \vee b \vee \overline{c})$
0	1	0	1	
0	1	1	1	$\wedge$
1	0	0	1	
1	0	1	1	
1	1	0	0	$(\overline{a} \vee \overline{b} \vee c)$
1	1	1	1	

Special case:

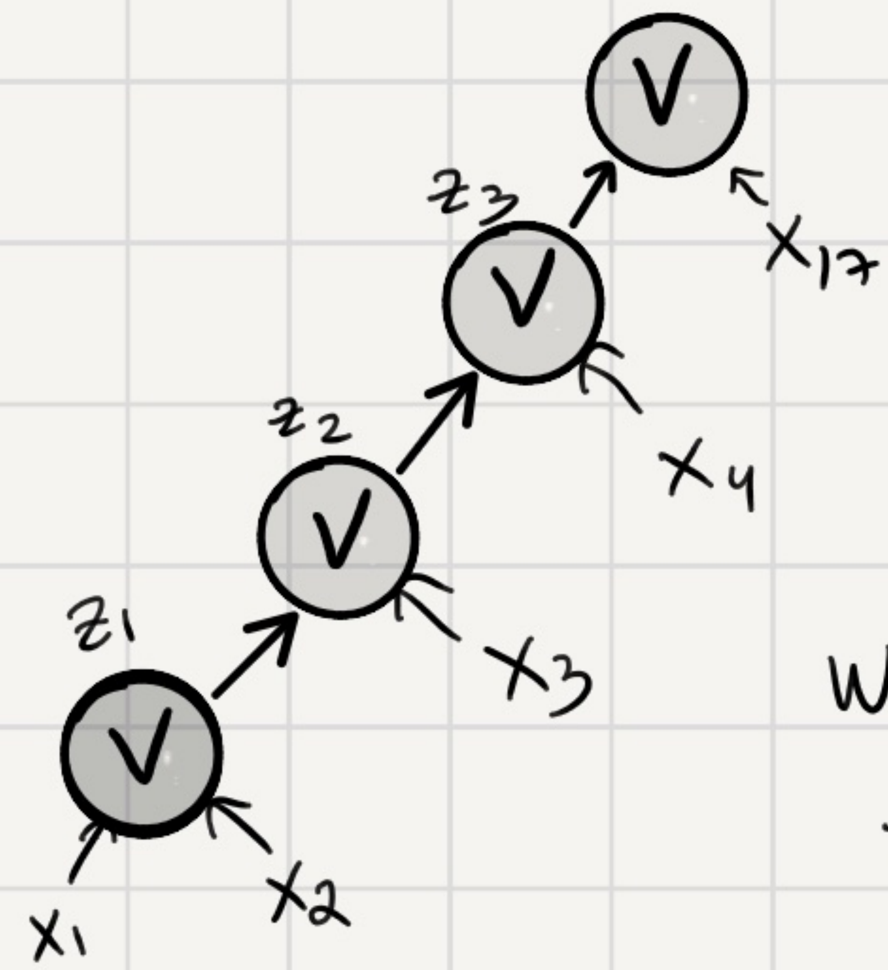
SAT $\rightarrow$ 3SAT (i.e. reducing general SAT on general CNFs to SAT on 3CNFs)

Example: Given

$$\underbrace{(x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_{17})}_{\text{clause } C_1} \wedge \underbrace{(\bar{x}_1 \vee x_5 \vee \bar{x}_{17} \vee x_{23})}_{\text{clause } C_2} \wedge \cdots \wedge \underbrace{(\quad)}_{\text{clause } C_m}$$

We convert every clause C_i to a 3CNF φ_i:

we write a simple circuit that computes C_i and then convert the circuit to a 3CNF formula.



> **Note:** During lecture, there were inaccuracies. They are fixed in these notes.

We introduce "helper" vars z_1, z_2, z_3 to capture the values of intermediate gates.

$$(z_3 \vee x_{17}) \wedge (z_3 = z_2 \vee x_4) \wedge (z_2 = z_1 \vee x_3) \wedge (z_1 = x_1 \vee x_2)$$

$z_3 = z_2 \vee x_4 \;\equiv\; (z_2 \vee x_4 \vee \bar{z}_3) \wedge (\bar{z}_2 \vee z_3) \wedge (\bar{x}_4 \vee z_3)$

$z_2 = z_1 \vee x_3 \;\equiv\; (z_1 \vee x_3 \vee \bar{z}_2) \wedge (\bar{z}_1 \vee z_2) \wedge (\bar{x}_3 \vee z_2)$

$z_1 = x_1 \vee x_2 \;\equiv\; (x_1 \vee x_2 \vee \bar{z}_1) \wedge (\bar{x}_1 \vee z_1) \wedge (\bar{x}_2 \vee z_1)$

This gives a 3CNF $\varphi_1(x, z)$

Given the values of x we have

$$C_1(x) = 1 \iff \exists z : \varphi_1(x, z) = 1.$$

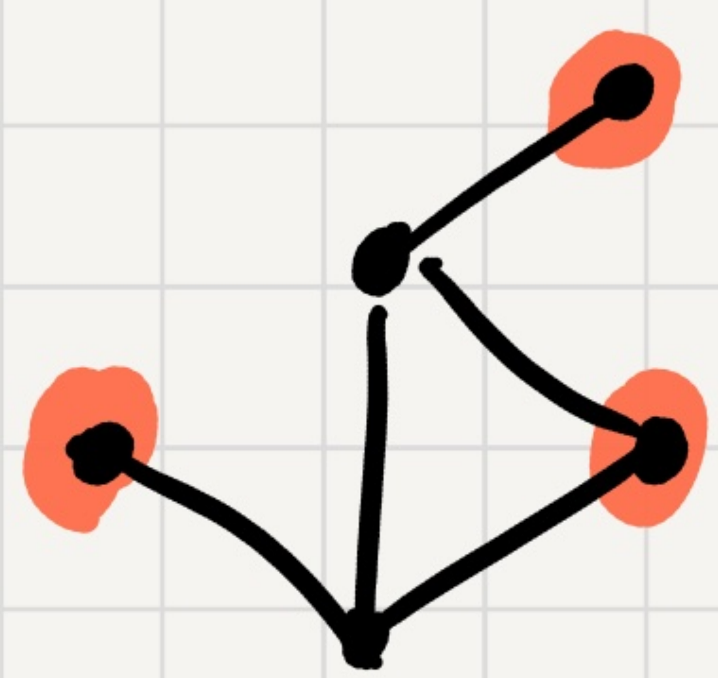
> **Note:** The 3CNF φ_1 is a bit different than the one described in the book but its still correspond to a valid reduction.
>
> - The 3CNF in the book is:
>
> $$(z_3 \vee x_{17}) \wedge (z_2 \vee x_4 \vee \bar{z}_3) \wedge (z_1 \vee x_3 \vee \bar{z}_2) \wedge (x_1 \vee x_2 \vee \bar{z}_1)$$
>
> $(z_2 \vee x_4 \vee \bar{z}_3) \equiv$ "$z_3 \le z_2 \vee x_4$", $(z_1 \vee x_3 \vee \bar{z}_2) \equiv$ "$z_2 \le z_1 \vee x_3$", $(x_1 \vee x_2 \vee \bar{z}_1) \equiv$ "$z_1 \le x_1 \vee x_2$"

3SAT → IS

IS: Given graph $G=(V,E)$ & integer k.
Is there an independent set of size k in G?

Defn: A set of vertices $S \subseteq V$ is an independent set if no two vertices in S have an edge between them.

Example:



The red vertices form an independent set of size 3.

Thm: IS is NP-complete.

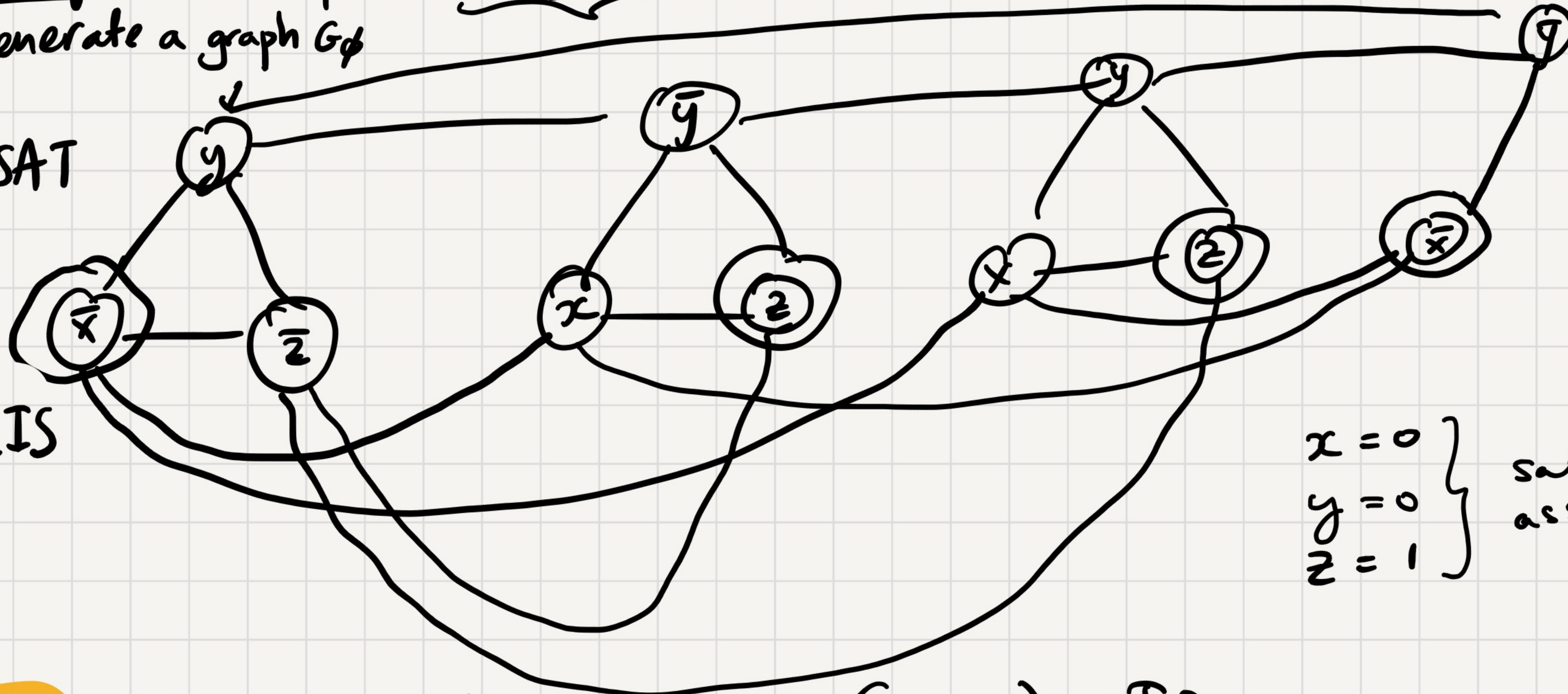
Proof: 1. IS $\in$ NP. Relation would be $((G,k), S) \in R$ iff $S \subseteq V$, $|S| = k$ & S is an independent set.

2. IS is NP-Hard ... Next slide

2. IS is NP-Hard: 3SAT $\longrightarrow$ IS.

Example: Given $\phi = \underbrace{(\bar{x} \vee y \vee \bar{z})} \wedge (x \vee \bar{y} \vee z) \wedge (x \vee y \vee z) \wedge (\bar{x} \vee \bar{y})$.

we generate a graph G_ϕ



$\phi \in 3SAT$

$\Updownarrow$

$(G_\phi, 4) \in IS$

$$\left.\begin{array}{l} x = 0 \\ y = 0 \\ z = 1 \end{array}\right\} \text{satisfying assignment}$$

Claim: ϕ is satisfiable $\Leftrightarrow$ $(G_\phi, 4) \in IS$

$\uparrow$

number of clauses

Claim: ϕ is satisfiable $\iff$ $(G_\phi, 4) \in IS$

$(\Rightarrow)$: ϕ is satisfiable $\Rightarrow$ fix an assignment $\vec{a} \in \{0,1\}^n$ that satisfies ϕ. In every clause, there exists at least one literal that $\vec{a}$ satisfies.

Pick exactly one literal in each clause that $\vec{a}$ satisfies.

$\updownarrow$

vertex in G_ϕ.

This set of $m = 4$ vertices is an independent set.

$(\Leftarrow)$: Let S be an independent set of size $m = 4$ in G_ϕ.

By the structure of G_ϕ, S has at most 1 vertex per "cloud" (triangle). $\Rightarrow$ We picked exactly one per cloud.

For each vertex $v \in S$, v represent some literal. Assign the relevant variable to satisfy the literal.

$\Rightarrow$ This gives a partial assignment that satisfies ϕ.

Def'n: A set $S \subseteq V$ of vertices in an undirected graph $G=(V,E)$ is called a "clique" if all pairs in S are connected by an edge.

A set $S \subseteq V$ is called a vertex cover if S touches all edges.

Clique: Given (G,k) is there a clique of size k in G?

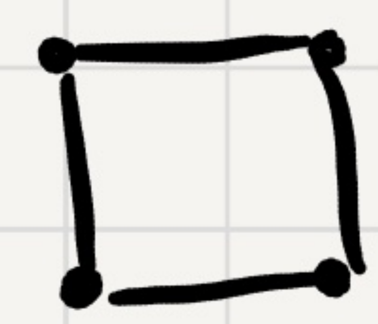
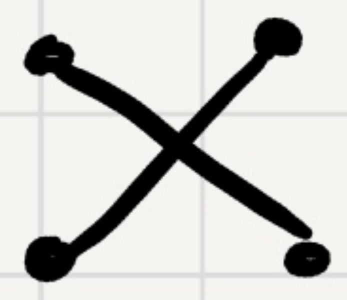
VC: Given (G,k) is there a vertex cover of size k in G?

Simple Reductions from IS to Clique & VC.

IS → Clique

$$(G,k) \to (\bar{G},k)$$

$G = (V,E)$, $\bar{G} = (V,\bar{E})$ where $\bar{E} = \{(u,v) : u,v \in V,\ (u,v) \notin E\}$

G  $\bar{G}$ 

IS → VC:

$$(G,k) \to (G, |V|-k).$$

claim: $S \subseteq V$ is an independent set

$\Updownarrow$

$V \setminus S$ is a vertex cover.