LECTURE 21

APRIL 7

→ RUDRATA CYCLE is NP-complete
(HAMILTON) .

→ CIRCUIT SAT is NP-complete
(Mother of all NP-completeness results)

Every problem in NP $\leq_p$ Circuit Sat

→ CIRCUIT SAT $\leq_p$ 3SAT

RUDRATA CYCLE (directed)

INPUT: A directed graph $G = (V, E)$

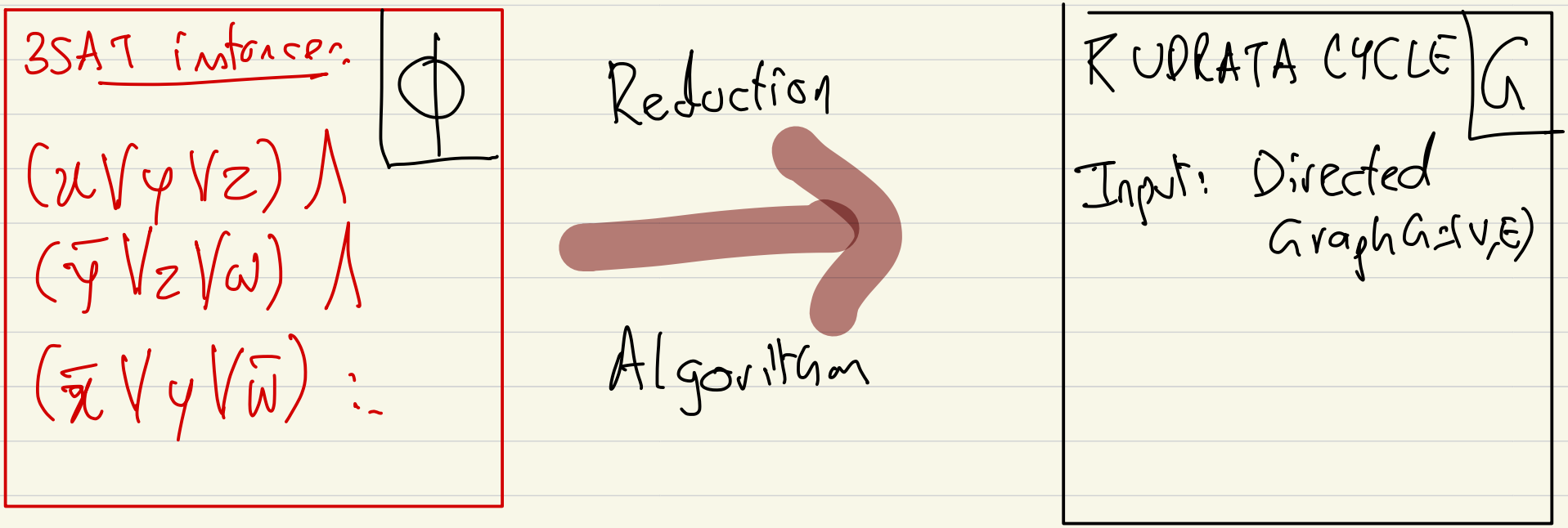
SOLUTION: A cycle passing through all nodes exactly once

RUDRATA CYCLE $\in$ NP

GOAL: RUDRATA CYCLE is NP-complete

∴ Thm: 3SAT $\leq_P$ RUDRATA CYCLE

3SAT $\leq_P$ RUDRATA CYCLE



$\exists$ a satisfying assignment to ϕ $\Longrightarrow$ $\exists$ a Rudrata cycle in G

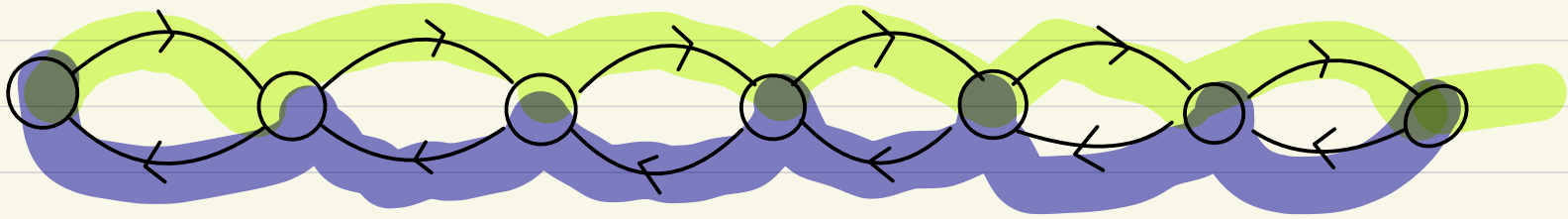
$\exists$ a RUDRATA cycle in G $\Longrightarrow$ $\exists$ a satisfying assignment in ϕ

3SAT $\longrightarrow$ RUDRATA CYCLE

$x \in \{0, 1\}$

$x = 0 \hookrightarrow$ Right to left

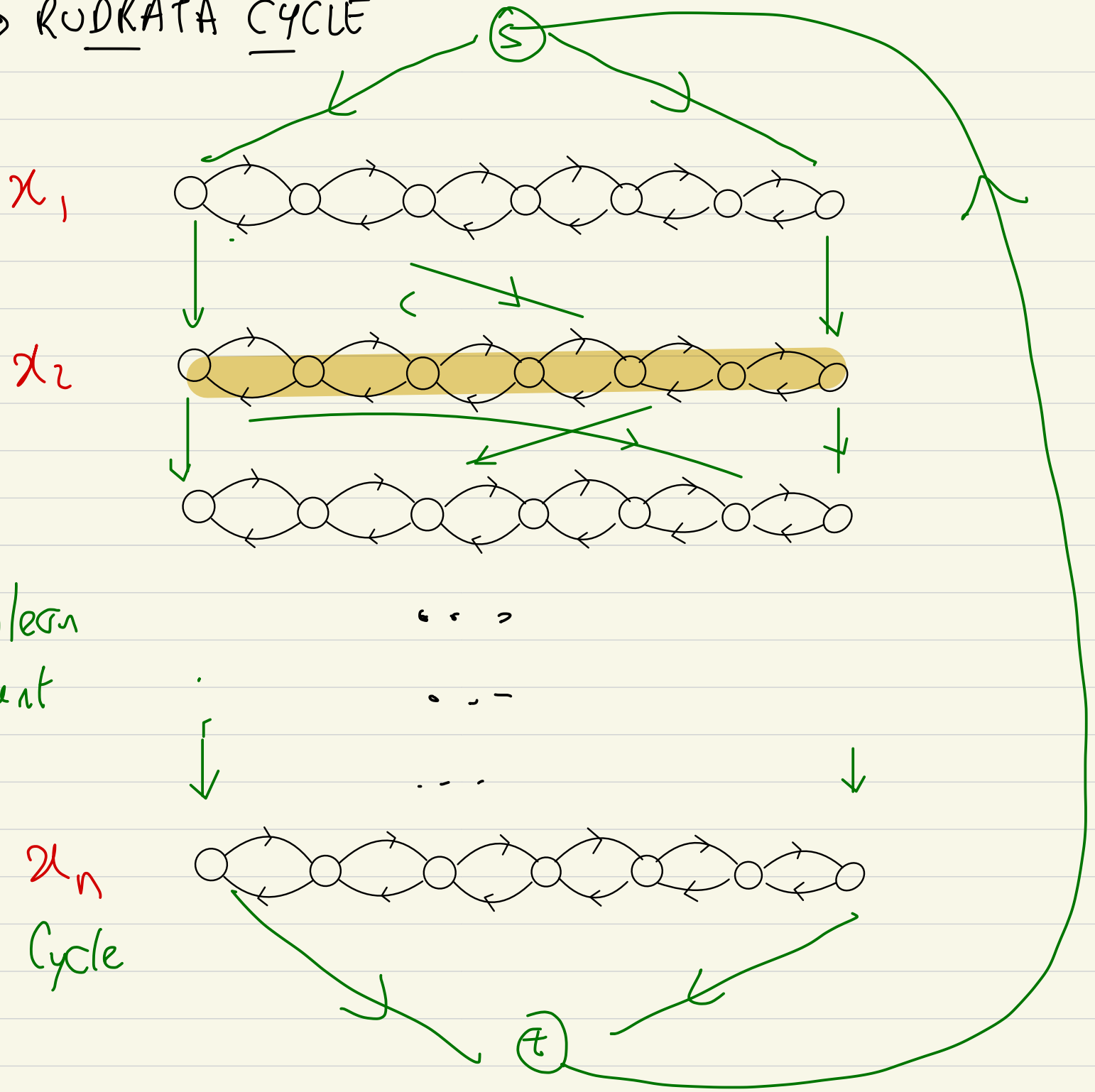
$x = 1 \hookrightarrow$ left to right



x

(Traverse the graph without skipping / repeating any vertex)

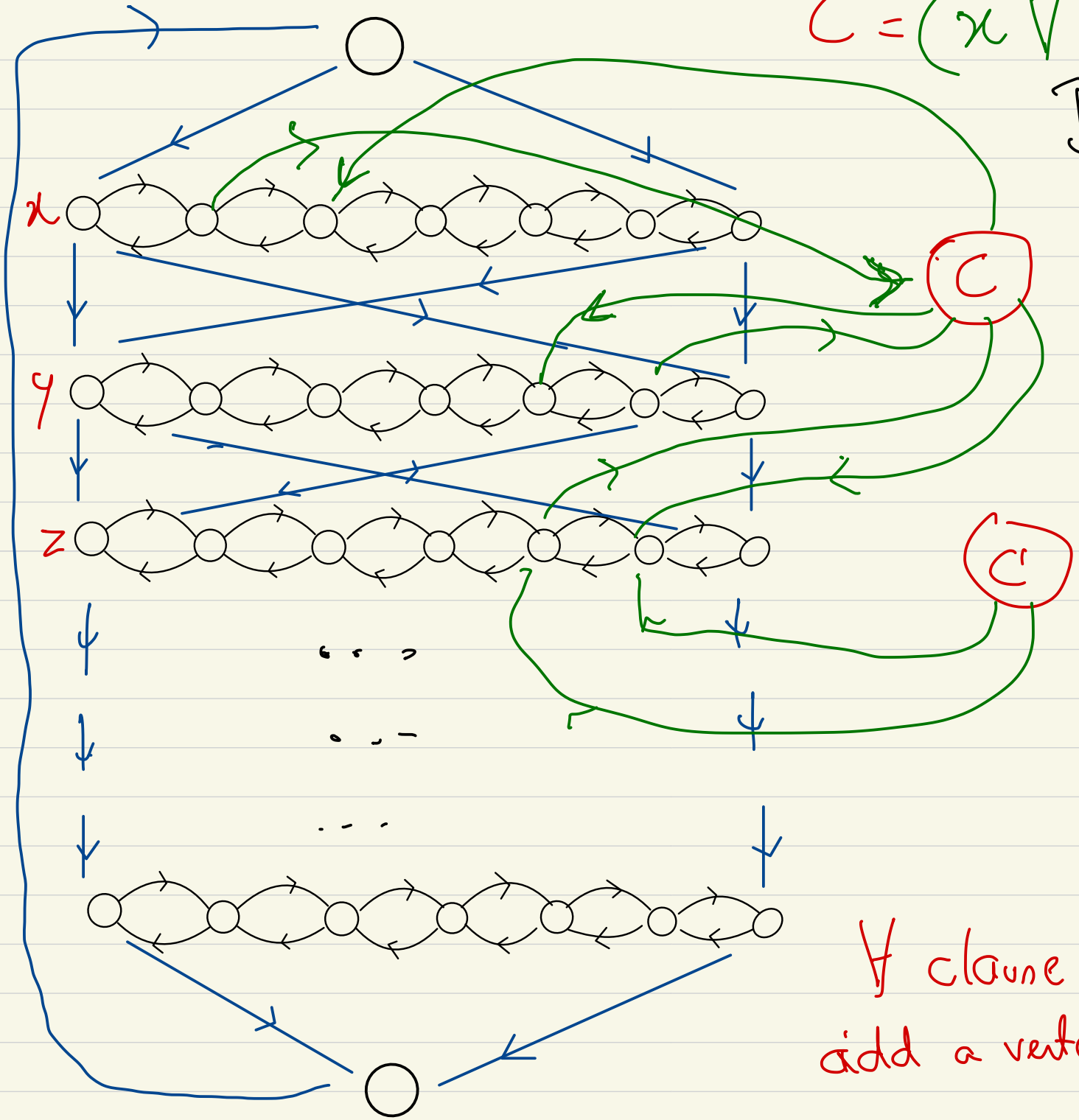
3SAT → RUDRATA CYCLE



Every boolean assignment

⇕

Rudrata Cycle



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

$x \Leftarrow$ L to R

OR

$y \Leftarrow$ R to L

or

$z \Leftarrow$ L to R

Cycle covers C

$\forall$ clause in 3SAT add a vertex, and connect it on Hamiltonian above

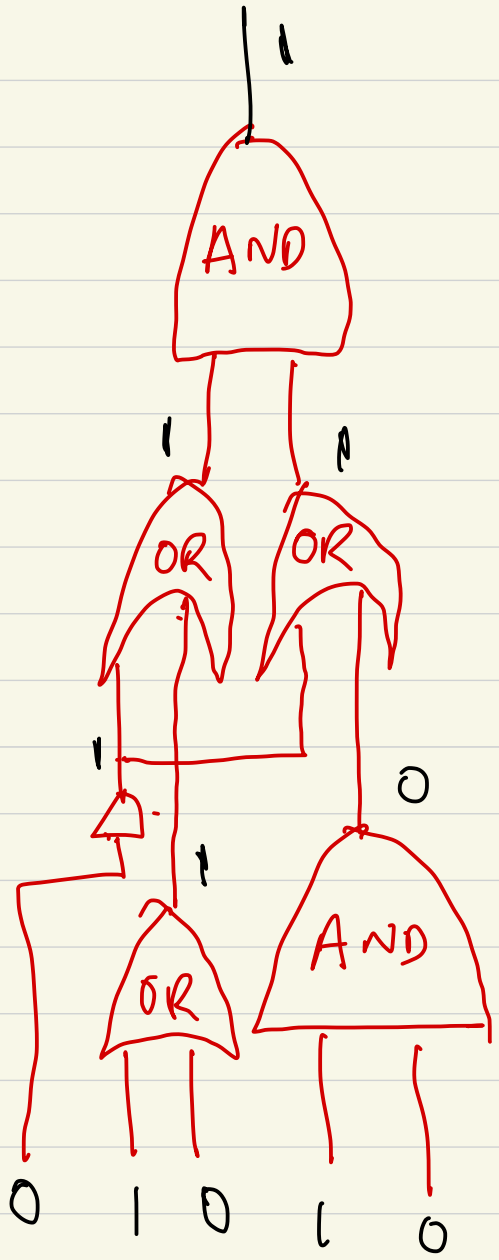
CIRCUIT SAT

INPUT: 1) Circuit with AND/OR/NOT gates

2) n inputs

SOLUTION: An boolean assignment so that output = 1.

Circuit SAT $\in$ NP



Circuit SAT is NP-complete:

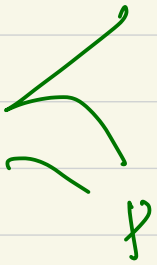
Every problem in NP $\leq_P$ Circuit SAT

Example: Factorization $\leq_P$?

FACTORIZATION

INPUT: An n bit number N

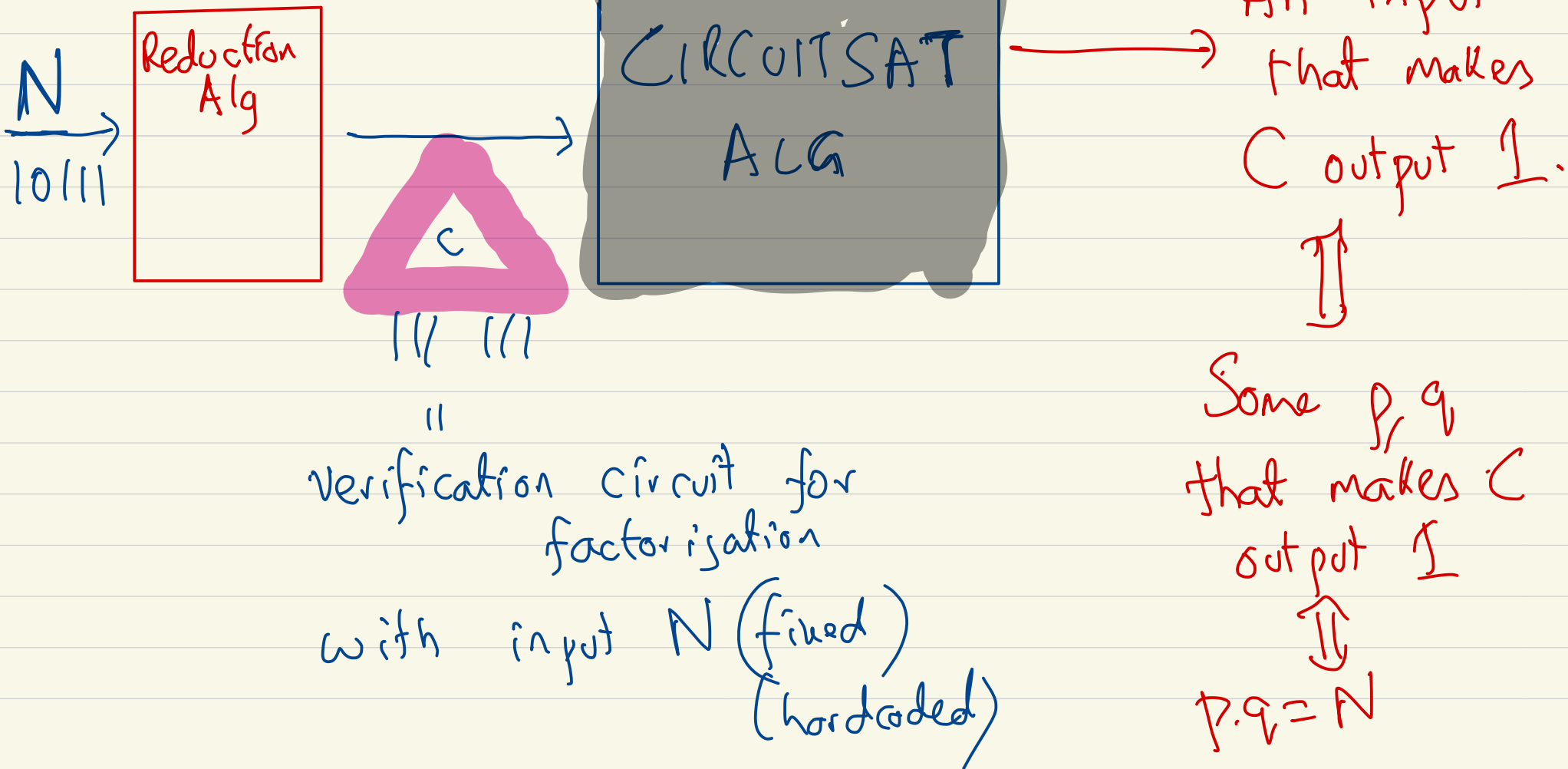
SOL: p, q $p, q > 1$ and $p \cdot q = N$



CIRCUIT SAT

INPUT: Circuit C

SOL: x s.t $C(x) = 1$



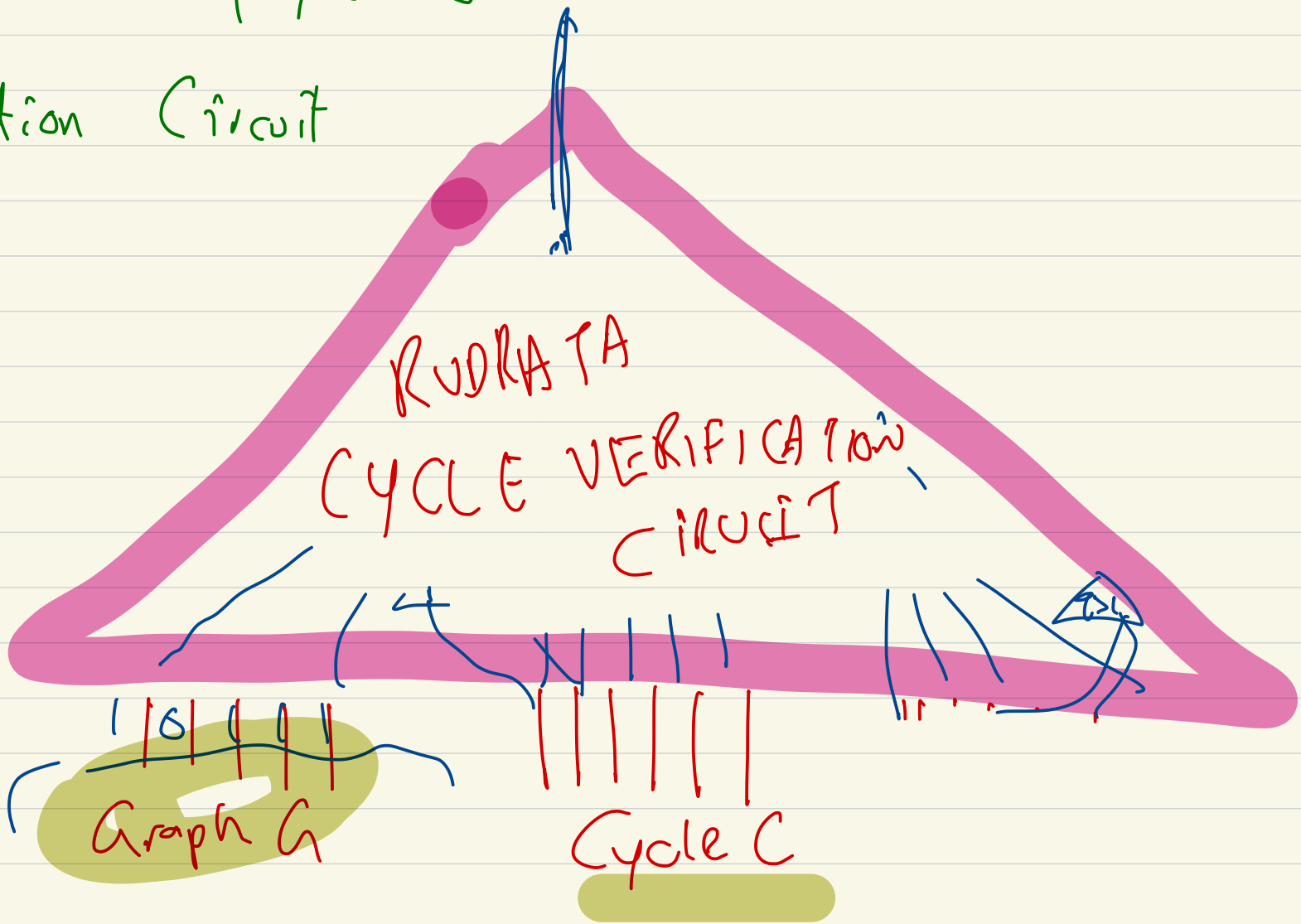
Polynomial

∃ a verification algorithm for Factorization

⇕

polytime Circuit

Verification Circuit



Same proof applies to every problem in NP

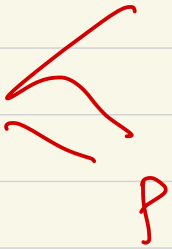
$\Rightarrow$ Every problem in NP $\leq_P$ Circuit SAT.

$\Rightarrow$ Circuit SAT is NP-complete.

CIRCUIT SAT

INPUT: A circuit C

SOL: An assignment x, s.t $C(x) = 1$

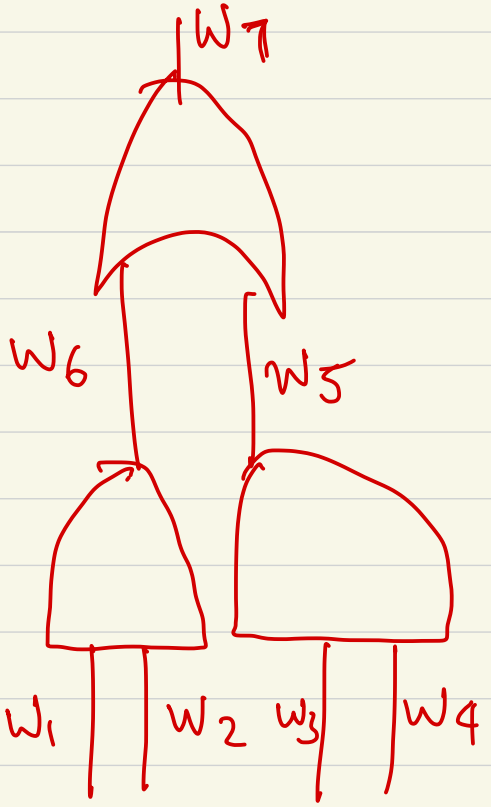


3SAT

INPUT: A 3SAT formula

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

SOL: A satisfying assignment



1) $\forall$ each wire w_i, introduce a variable in 3SAT

$\{w_1, \ldots w_7\}$

2) $\forall$ each gate introduce clauses

Example: we want

$w_6 = w_1 \wedge w_2$

$\Updownarrow$

Simulate this using clauses

Obs: Every constraint on 3 variables can be simulated via clauses of form $(x \vee y \vee z)$ ....

Example: Suppose we want a constraint

$x = y \wedge z$,

Then $x = y \wedge z$

$$\left\{ \begin{array}{l} (x \vee \bar{y} \vee \bar{z}) \\ (\bar{x} \vee y \vee \bar{z}) \\ (\bar{x} \vee \bar{y} \vee z), \\ (\bar{x} \vee y \vee z) \end{array} \right\}$$

4 clauses simulate $x = y \wedge z$.

To express a constraint using 3SAT clauses

Example: $x = y \wedge z$

↓

Forbidden assignments

0	1	1	⟶	$(x \vee \bar{y} \vee \bar{z})$
1	0	1	⟶	$(\bar{x} \vee y \vee \bar{z})$
1	1	0	⟶	$(\bar{x} \vee \bar{y} \vee z)$
1	0	0	⟶	$(\bar{x} \vee y \vee z)$