Alt. Def of NP

nondeterministic polynomial

NP: set of decision problems Q for which there is a poly-time randomized algorithm A s.t. $Q(x) = YES$ then $P(A(x) = YES) > 0$ $(x \in Q)$ and if $Q(x) = NO$ then $P(A(x) = YES) = 0$

NPR:EB

↓ test random bit is a single step

Definitions are EQUIVALENT $Q \in NP \iff NPR:EB$

HP $\in$ NPR:EB

HP(G)

p ← random permutation of G.V
return isValidPath(G, p)

HP $\in$ NP

HP-VERIFY(G, p)

if p contains all v's exactly once then
  return isValidPath(G, p)
else
  return NO

$NP \subseteq NPR:EB$ if $Q \in NP$ then there is a poly-time verification alg Q-VERIFY(x, y)
then let Q-RANDOMIZED be 1) randomly generate y
2) return Q-VERIFY(x, y)

$NPR:EB \subseteq NP$ if $Q \in NPR:EB$ then there is a Q-RANDOM(x) for Q that runs in poly-time

then Q-VERIFY(x, y) is 1) simulate Q-RANDOM(x) using y as random bits

CIRCUIT-SAT is NP-complete (Cook-Levin Thm)

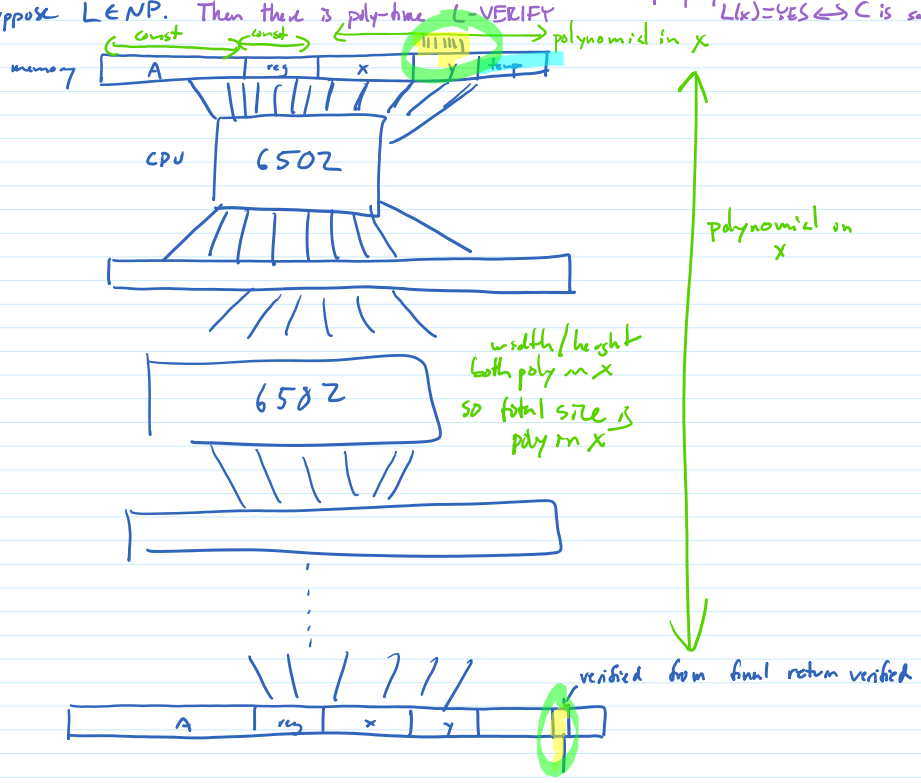
→ given a combinational circuit, is there a setting of 0/1 on inputs to make output 1

really SAT is NP-c but similar idea

1) CIRCUIT-SAT $\in$ NP

2) $\forall L \in NP$, $L \leq_p$ CIRCUIT-SAT [goal: given input x to L, construct circuit C in poly time s.t. L(x)=YES $\leftrightarrow$ C is satisfiable]

Suppose $L \in NP$. Then there is poly-time L-VERIFY



CIRCUIT-SAT → SAT ⟷ 3-SAT → IS → VC

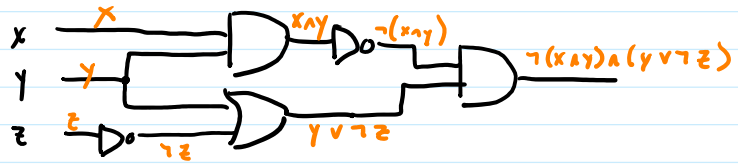
SAT is NP-complete

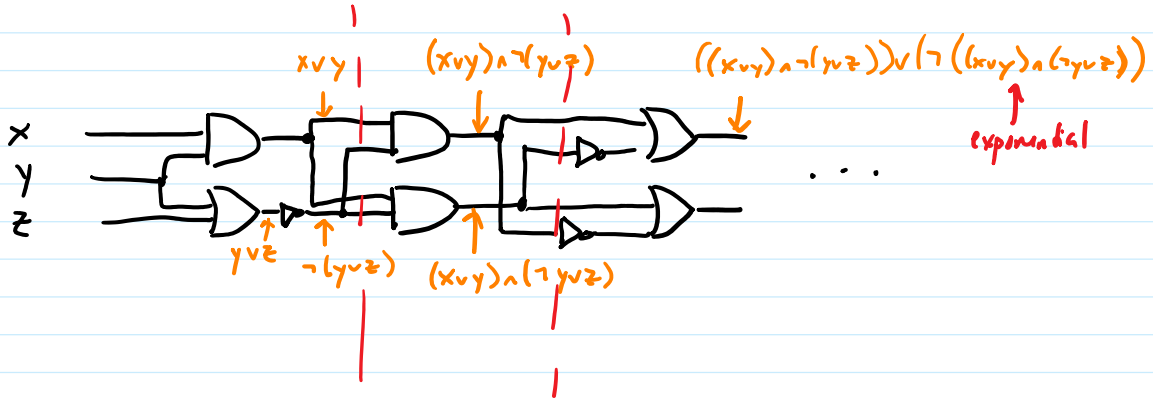
Proof: 1) SAT ∈ NP [evidence is satisfying assignment]

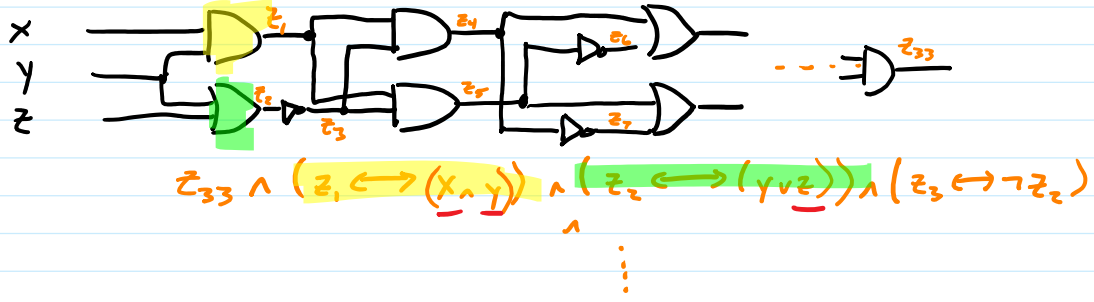
2) CIRCUIT-SAT $\in_P$ SAT in poly-time

[goal: given circuit C, find φ s.t. size(φ) is polynomial in size(C) and C is satisfiable if and only if φ is]

CIRCUIT-SAT(C)
1) transform C to φ
2) output SAT(φ)







SAT and 3-SAT

3-SAT $\in$ NP (before)

SAT $\leq_p$ 3-SAT [Goal: given φ, create 3-CNF φ' in poly-time s.t. $\varphi \equiv \varphi'$]

$(x \wedge y) \wedge \sim(x \vee y)$ $\quad z_1 \wedge (z_1 \leftrightarrow x \wedge y) \wedge (z_2 \leftrightarrow (x \vee y)) \wedge \cdots$

z_1 z_4 z_3 z_2

x	y	z_1	$z_1 \leftrightarrow x \wedge y$	$\neg(z_1 \leftrightarrow x \wedge y)$
T	T	T	T	F
T	T	F	F	T
T	F	T	F	T
T	F	F	T	F
F	T	T	F	T
F	T	F	T	F
F	F	T	F	T
F	F	F	T	F

$\neg(z_1 \leftrightarrow x \wedge y) \equiv (x \wedge y \wedge \neg z_1) \vee (x \wedge \neg y \wedge z_1) \vee \cdots$

$\neg\neg(z_1 \leftrightarrow x \wedge y)$

$\equiv z_1 \leftrightarrow x \wedge y \equiv (\neg x \vee \neg y \vee z_1) \wedge \cdots$

$z_1 \leftrightarrow x \wedge y \equiv (x \wedge y \wedge z_1) \vee (x \wedge \neg y \wedge \neg z_1) \vee (\neg x \wedge y \wedge \neg z_1) \vee (\neg x \wedge \neg y \wedge \neg z_1)$

$\neg(z_1 \leftrightarrow x \wedge y) \equiv (\neg x \vee \neg y \vee \neg z_1) \wedge (\neg x \vee y \vee z_1) \wedge \cdots$