CS124 Lecture 19

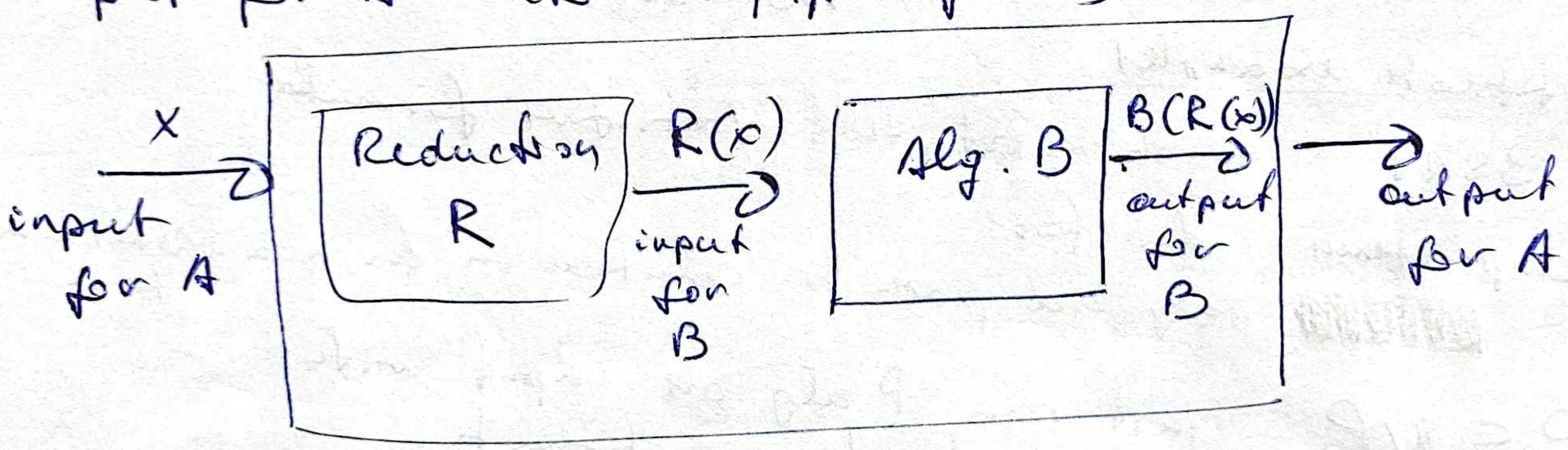
Polynomial time problems P

P = set of problems with a yes/no answer s.t.
∃ algorithm A, integer k s.t.
A runs in $O(n^k)$ time on inputs of size n

Luckily, most algs have small polynomials.

Reduction

Poly-time Reduction from A to B
is an Alg. R which turns
input for A into an input for B



Poly-time Alg for B
Poly-time reduction
⇒ Poly-time Alg for A

$A \leq_R B$

$T(x)$ = time $x \to R(x)$

$S(y)$ = time for Alg. B on y

$T(x) + S(|R(x)|) =$ poly n $(|x|)$

$A \leq_R B$, $B \leq_R C \Rightarrow A \leq_R C$
composition of reductions, transitivity

$X \to R_1(X) \to R_2(R_1(X))$

⟶ direction of reduction

$A \leq_R B$

① B easy ⇒ A easy, use reduction to solve prob.

② A hard ⇒ B hard, use reduction identify hard problems

NP (nondeterministic polynomial time)

"checked" in polynomial time

Problems where — if the answer is yes
there is a "short" certificate, s.t.
↑ poly-size
given the soln. / short certificate
the soln. can be verified in poly time

∃ a checking algorithm that
takes as input : problem input, short certificate
and returns : valid, if answer is yes to problem
and certificate is valid
no otherwise

certificate examples
- 3SAT : truth asst that satisfies formula
- Composites : factor
- [redacted] poly-size path to solution in an non-deterministic tree

$P \subseteq NP$ just run P alg. on input with empty string certificate

$P \stackrel{?}{=} NP$

Hardest Problems in NP

NP-Complete : ← NP-complete

① in NP

② (NP-Hardness) all other problems in NP reduce to it
may not be in NP ⇒

defn: $A \leq_R B$ $B \in NP$ ⇒ $A \in NP$

Once a problem is NP-complete

X, new NP-complete problems

show $X \leq_R Y$

↑ $\in NP$

Cook-Levin theorem (NP-Complete)

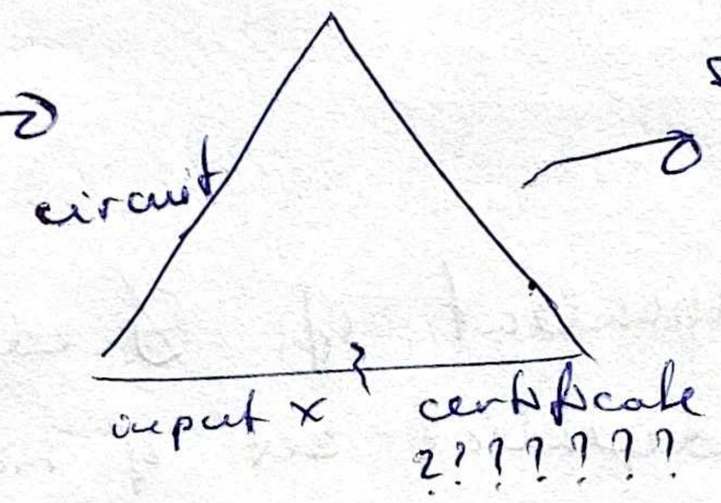
Circuit SAT:

given a Boolean circuit, values of some inputs, can the rest of the inputs be set s.t. it evaluates to true

Iff a problem is in NP, it can be reduced to circuit SAT.

[input, certificate] Alg. A

Alg A = Polysized boolean circuit

Alg A, input → 

→ solution to circuit SAT solves for certificate that makes x evaluate to true

⇒ any NP problem can be reduced to circuit SAT

↓

if certificate found → true
if no certificate found → false

3SAT is NP-Complete (circuit SAT $\leq_R$ 3SAT)

Reduction circuit SAT to 3SAT formula, IFF Poly-time transformation

① input gates	if T	include (x)
	if F, unspec. input	include ($\bar{x}$)
		nothing
② $x = y \vee z$		include $(\bar{y} \vee x) \wedge (\bar{z} \vee x) \wedge (\bar{x} \vee y \vee z)$
$x = y \wedge z$		include $(\bar{x} \vee y) \wedge (\bar{x} \vee z) \wedge (\bar{y} \vee \bar{z} \vee x)$
$x =$ NOT y		incl. $(\bar{x} \vee \bar{y}) \wedge (x \vee y)$
③ output gate x		(x)

3SAT $\leq_R$ ILP

T ↪ 1
F ↪ 0

$0 \leq X \leq 1$

for all variables in 3SAT clauses

$(X \vee \bar{Y} \vee Z) \wedge (a \vee b \vee c)$

↓ IFF

$X + (1-Y) + Z \geq 1$

↓ IFF

$a + b + c \geq 1$

3SAT $\leq_R$ Independent Set

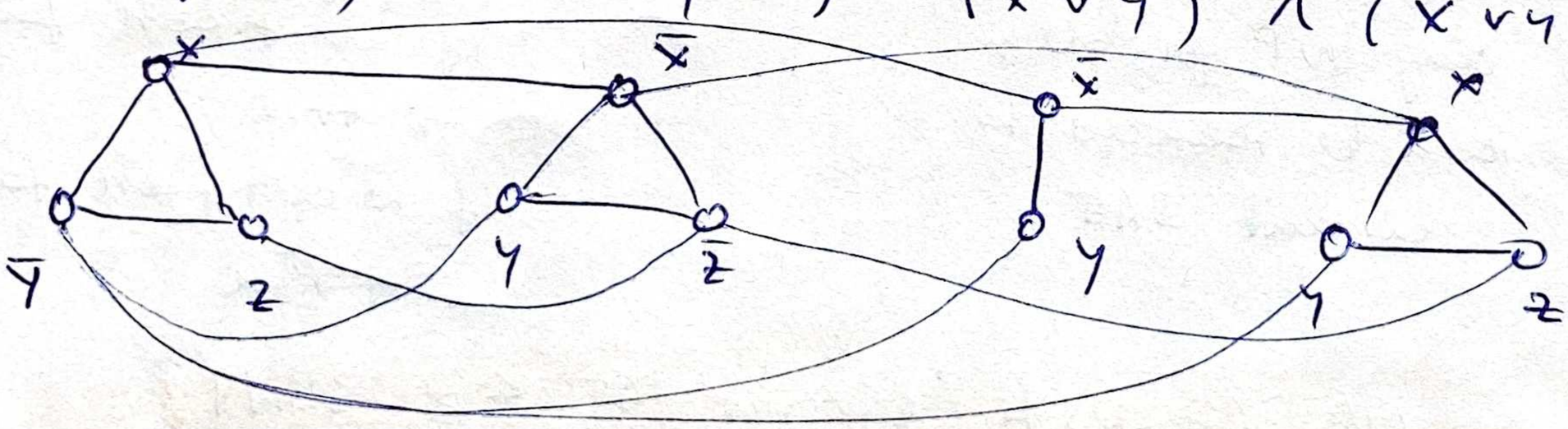
Independent Set

$G = (V, E)$

$I \subseteq V$ is independent if $\nexists$ edge (u, v), s.t. $u \in I$, $v \in I$

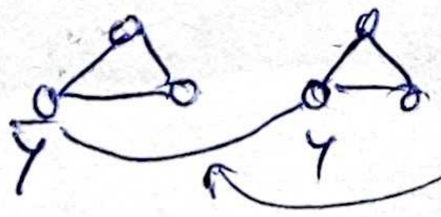
Is there an independent set of size $\geq k$

$(X \vee \bar{Y} \vee Z) \wedge (\bar{X} \vee Y \vee \bar{Z}) \wedge (\bar{X} \vee Y) \wedge (X \vee Y \vee Z)$



formula satisfiable $\Leftrightarrow$ $\exists$ indep. set of size $\geq$ # of clauses

← force to pick one vertex from each clause
↳ true vertex



cannot pick $\bar{Y}$ and Y, both cannot be true

poly-time contradiction ✓

Independent set ≤P Vertex Cover

$S \subseteq V$ s.t. all edges are incident to at least one vertex in S

"vertices covering all edges"

Is there a VC ≤ [redacted] size k

S is VC <=> V−S is indep. set

Independent set ≤P Clique

take G^c

independent sets in G are cliques in G^c