Decision Problems

- **We will consider only problem with Yes-No answers.**

Example Traveling Salesman Problem (TSP)

1. **Given a nonnegative integer edge weighted graph, what is the minimum length cycle that visits each node exactly once ?**
2. **Given a nonnegative integer edge weighted graph and an integer K, is there a cycle that visits each node exactly once, with weight at most K?**

CNF

- A Boolean formula is called conjunctive normal form (CNF) if, for the input Boolean variable $x_1, x_2, \ldots, x_n$, it has the following form

$(x_1 + \overline{x_1} + x_2)$ $(x_3 + \overline{x_2} + x_1)$ $(x_2 + x_1)$ (x_3)

$$x_{out} = \underbrace{(x_1 \vee \neg x_1 \vee x_2)}_{\text{clause}} \wedge (x_3 \vee \neg x_2 \vee x_1) \wedge (x_2 \vee x_1) \wedge (x_3)$$

Output Boolean variable

literals

i.e. CNF = product-of-sums

CNF

- 2-CNF: each clause has two distinct literals.

$$x_{out} = (x_1 \vee \neg x_1) \wedge (x_3 \vee x_2) \wedge (x_2 \vee x_1)$$

$(x_1 + \overline{x_1}) \quad (x_3 + x_2) \quad (x_2 + x_1)$

- 3-CNF: each clause has three distinct literals.

$$x_{out} = (x_1 \vee \neg x_1 \vee x_2) \wedge (x_3 \vee \neg x_2 \vee x_1) \wedge (x_2 \vee x_1 \vee x_3)$$

Some Problems: P vs NP-complete

- **2SAT:** Given a Boolean formula in **2-CNF**, is there a satisfying truth assignment to the input variables?
- **3SAT:** Given a Boolean formula in **3-CNF** is there a satisfying truth assignment to the input variables?
- SAT: Is a given n-variable Boolean formula in CNF satisfiable? Here, "satisfiable" means "can be made 'true'"

Example: $(x+y)(\bar{x}+z)$ is satisfiable

$(x+z)(\bar{x})(\bar{z})$ is not satisfiable

P

Definition:

A problem A is in **P** if for every input x the solution $A(x)$ can be computed in polynomial time $O(\text{poly}(\text{length}(x)))$.

NP

Formal Definition

A problem A is **NP** *if there exist a polynomial p and a polynomial-time algorithm $V()$ such that x is a YES-input for problem A if and only if there exists a solution y, with $length(y) \leq p(length(x))$ such that $V(x, y)$ outputs YES.*

NP = Non-deterministic Polynomial-time

Intuitively, a problem is in **NP** if it can be formulated as the problem of whether there is a solution

- They are *small*. In each case the solution would never have to be longer than a polynomial in the length of the input.
- They are *easily checkable*. In each case there is a polynomial algorithm which takes as inputs (the input of the problem) and (the alleged solution), and checks whether the solution is a valid one for this input. In the case of **3SAT**, the algorithm would just check that the truth assignment indeed satisfies all clauses. In the case of *Hamilton cycle* whether the given closed path indeed visits every node once. And so on.
- Every "yes" input to the problem has at least one solution (possibly many), and each "no" input has none.

- For every yes-input *x* to *A*, there is a polynomial size piece of ***evidence_x*** which can be checked in polynomial time that indeed *x* is a yes-input. This `evidence' (sometimes called a **`certificate'** or **`witness'** of **`proof'**) may be very hard to come up with, but is easy to check.

$P \subseteq NP$

THEOREM: $P \subseteq NP$

Proof: This is obvious. If you can solve a problem efficiently, there exists short evidence as to what the solution is (both for YES and NO inputs!). Simply, **use the execution trace of your solving algorithm as the evidence** that the answer is YES or NO depending on what the case may be.

3-COLOR

3-COLOR problem: Given a graph $G=(V,E)$, is there a function $c: V \longrightarrow \{1,2,3\}$ s.t. $c(x) \neq c(y)$ for all edge $xy \in E$?

Fact: 3-COLOR problem is a decision problem.

Fact: 3-COLOR problem is in **NP**.

pf: Let G be a YES-input for the 3-COLOR problem.

G is 3-colorable $\Longrightarrow \exists\ c: V \longrightarrow \{1,2,3\}$

$c(x) \neq c(y)$ for any edge $\overline{xy}$ of G

Such a coloring c can serve as a certificate which can be checked in polynomial time that indeed G is 3-colorable.

For each edge $xy \in E(G)$
  do if $c(x)=c(y)$ **then** STOP and G is not 3-colorable.
Print "G is 3-colorable"

QED

Reductions

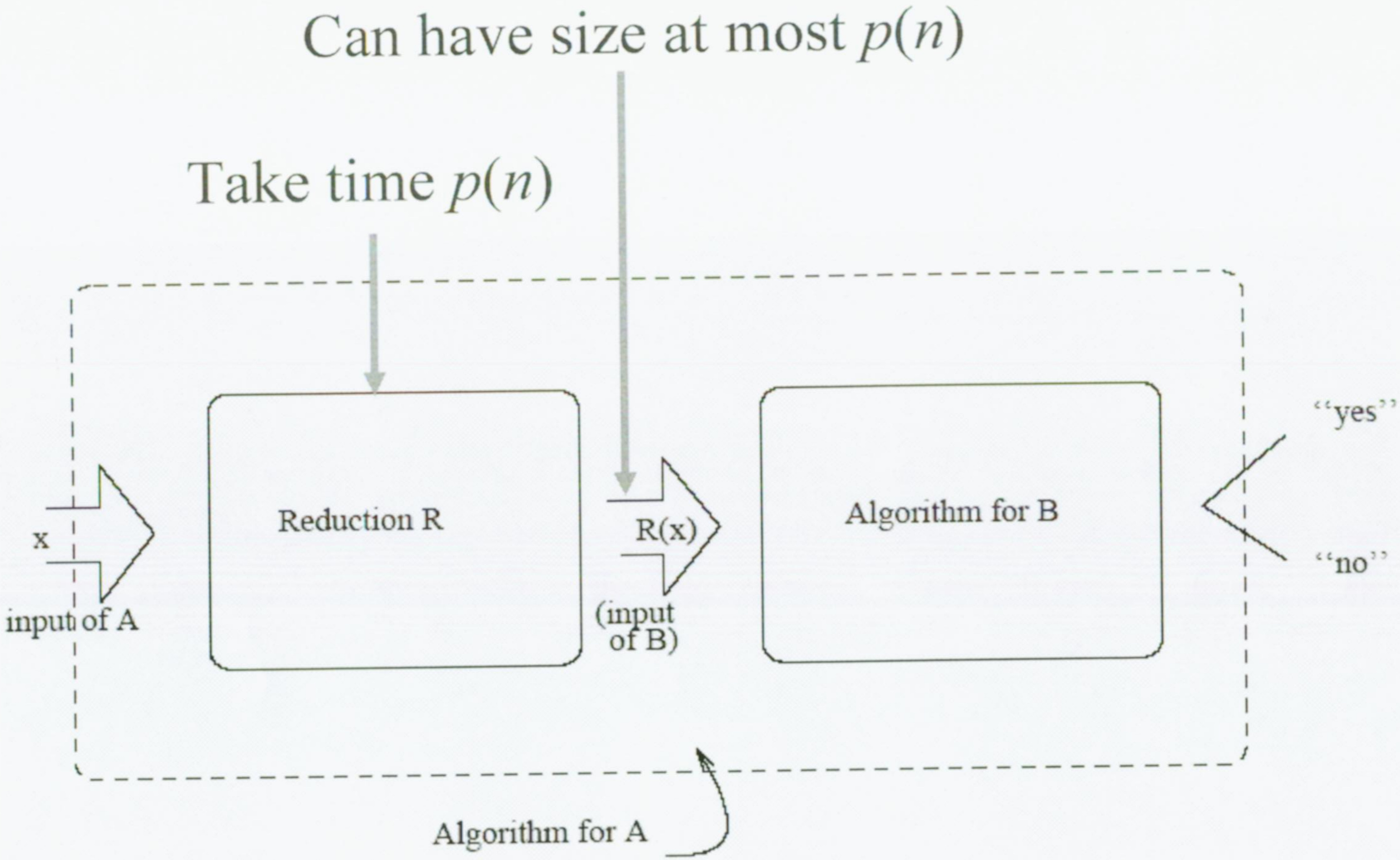
- A *reduction* from decision problems A to B is a polynomial-time algorithm R such that for every input x to problem A we have

$$A(x) = B(R(x))$$

x is a "yes" input of A $\Leftrightarrow$ $R(x)$ is a "yes" input of B

- We write

$$A \leq B$$



$$\{A \leq B\} + \{B \text{ has a poly. time algor.}\}$$

$$\Rightarrow \{A \text{ has a poly. time algor.}\}$$

R3

Reductions

$\{A \leq B\}$

$\Rightarrow$ If we known A is hard,

then B is hard.

NP-hard

Definition

A problem B is **NP-hard** if for every problem A in NP, $A \leq B$.

NP-hard = Non-deterministic Polynomial-time hard

NP-complete

Definition

A problem B is **NP - complete** if it is **NP - hard** and it is contained in **NP**.

Lemma

If B is **NP-complete**, then B **is in P** if and only if **P**=**NP**.

Proving more NP-completeness result

Lemma $A \leq B$ & $B \leq C \Rightarrow A \leq C$

Lemma Let $C \in$ NP-complete and $A \in NP$.
If $C \leq A$ then $A \in$ NP-complete

The first NP-Complete Problem

Cook's Theorem (1971)

SAT is NP-Complete.

- Stephen Arthur Cook, "The complexity of Theorem Proving Procedures"
- Cook received Turning Award in 1982.

Millennium Prize Problem: P = NP ?

SAT $\leq$ 3SAT

Thm 3SAT is NP-complete.

Pf: It suffices to show that SAT $\leq$ 3SAT. (sketch!)

Let $(\cdots\cdots)(x_1 + x_2 + \cdots\cdots + x_t) \cdots\cdots (\cdots\cdots)$ be an input to SAT.

$$(x) \cong (x + a + b)(x + \bar{a} + b)(x + a + \bar{b})(x + \bar{a} + \bar{b})$$

$$(x + y) \cong (x + y + c)(x + y + \bar{c})$$

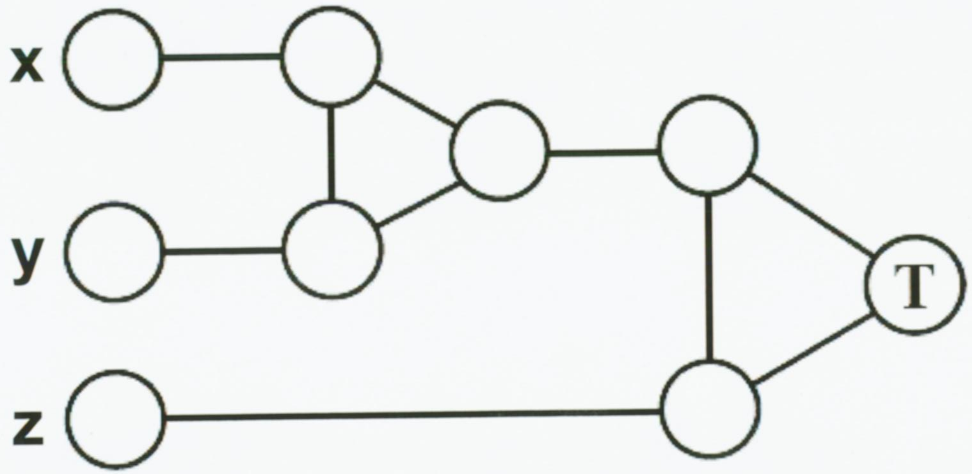
$$(x_1 + x_2 + x_3 + \cdots + x_6) \cong (x_1 + x_2 + \alpha)(\bar{\alpha} + x_3 + \beta)(\bar{\beta} + x_4 + \gamma)(\bar{\gamma} + x_5 + x_6)$$

Note that LHS is satisfiable iff RHS is satisfiable.

QED

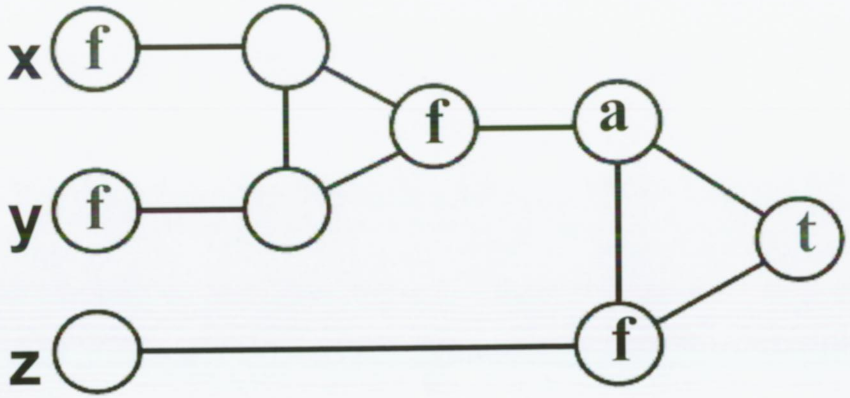
Observations

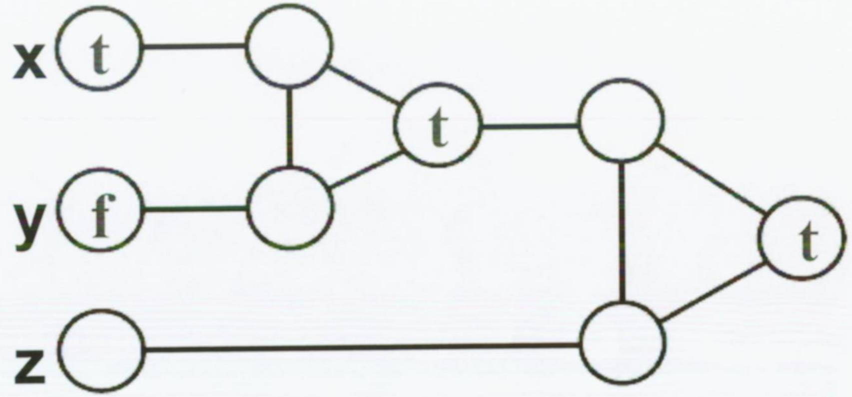
Claim: Using three colors **{t, f, a}** to color a graph. Suppose we are given the following graph in which vertices ***x***, ***y*** and ***z*** not all receiving the same color. Then we can properly **3-color** the other vertices so that vertex T receives the color **t.**

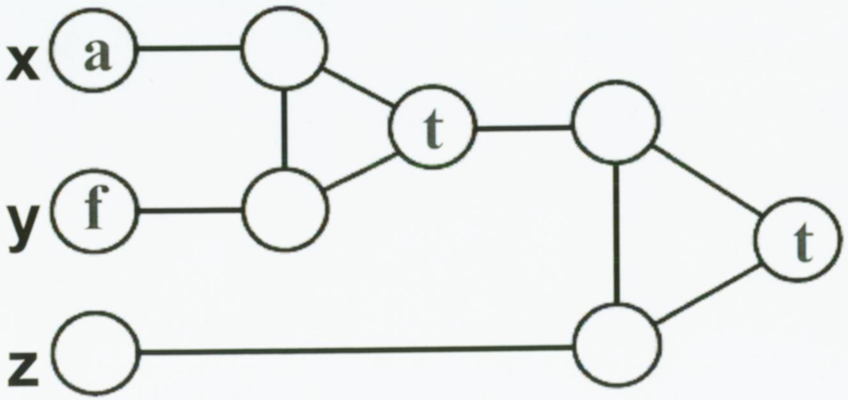


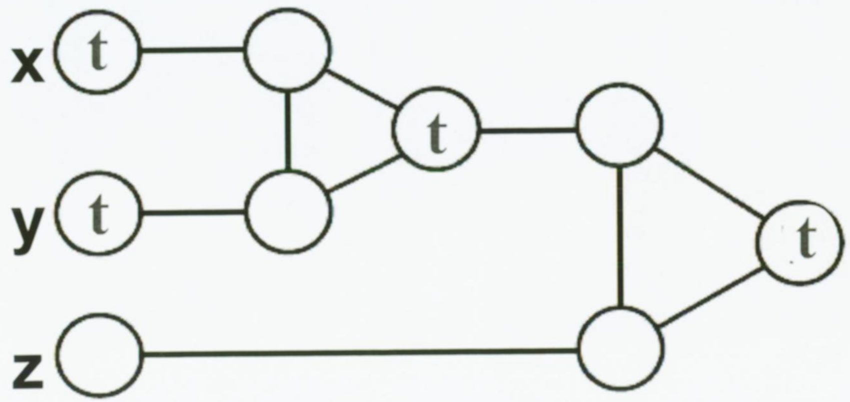
02

proof:



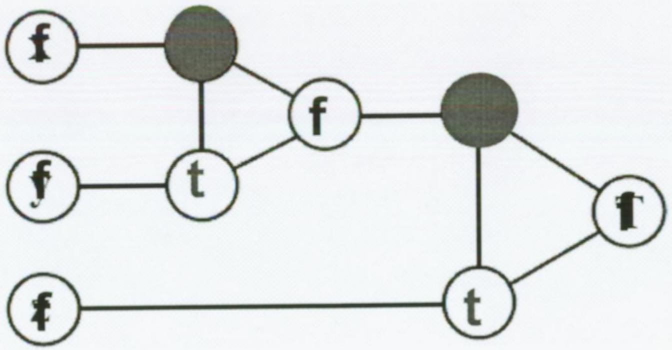






QED

Remark: Suppose vertices x, y and z received the same color. When we 3-color the other vertices of the graph we will arrive at the following situation. That vertex T always receive the same color as $\{x,y,z\}$.



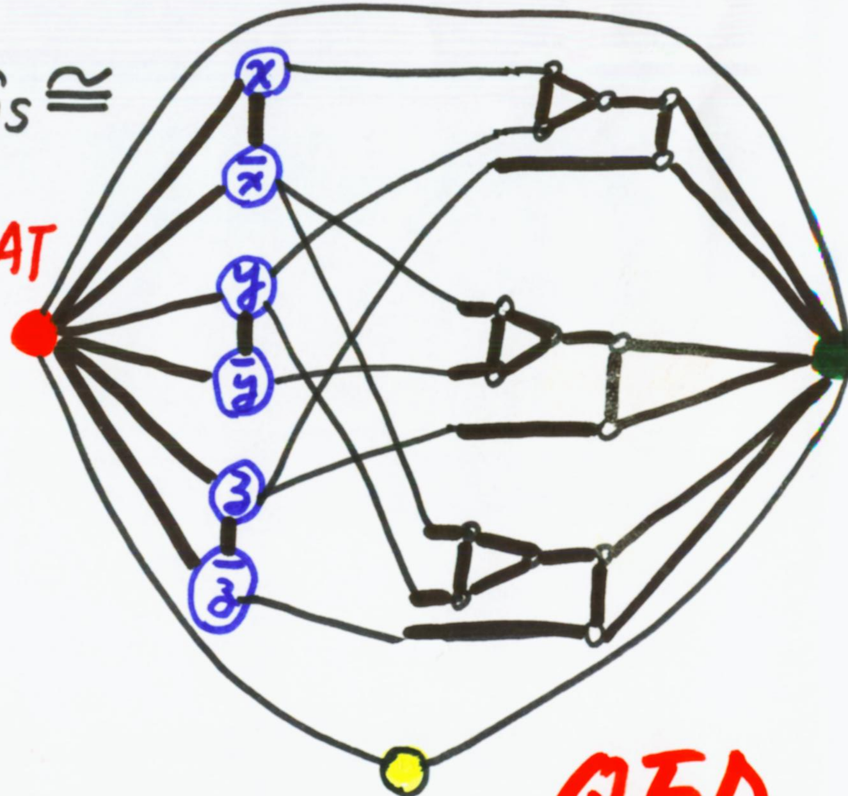
3-COLOR

Thm 3-COLOR problem is NP-complete.

pf: To show 3SAT $\leq$ 3-COLOR. (sketch)

let $S = (x+y+z)(\bar{x}+\bar{y}+z)(\bar{x}+y+\bar{z})$ be an input of 3SAT.

let G_S be an input of 3-COLOR, where $G_S \cong$



We claim that S is a "yes" input of 3SAT if and only if G_S is a "yes" input of 3-COLOR. why?

QED