▷ In this way, any clause in k literals can be converted to an equivalent 3-CNF (i.e. conjunction of clauses, each having at most 3 literals).

▷ Thus, 3SAT := {φ | φ is a satisfiable 3CNF formula} is as "hard" as SAT.

– What about 2SAT ?

Reduction

– Defn: • We call a language A poly-time Karp reducible to a language B if ∃ poly-time TM M s.t.

$$x \in A \iff M(x) \in B.$$

• We denote this as $A \leq_p B$.

• We say B is NP-hard if

$$\forall A \in NP,\ A \leq_p B.$$

• Further, B is <u>NP-complete</u> if
B is NP-hard & $B \in NP$.

<u>Theorem</u> (Cook, Levin 1971): 3SAT is NP-complete.

<u>Proof:</u>

• We have proved that $\forall L \in NP$, $L \leq_p$ 3SAT.

• Also, 3SAT $\in NP$. □

– Some easy properties of reductions:

<u>Proposition</u>: (i) $A \leq_p B \leq_p C \Rightarrow A \leq_p C$.

(ii) A is NP-hard & $A \in P \iff P = NP$.

(iii) A is NP-hard & $A \leq_p B \Rightarrow B$ is NP-hard.

– In this sense, the NP-complete problems are the <u>hardest</u> in NP!

– Other egs. of NP-complete problems?

- TSP, Subsum, IntProg, etc. are all NP-complete.

- <u>Several hundreds</u> of NP-complete problems are known!

- We already saw that testing the existence of a boolean feasible point in an integer program is in NP.

▷ IntProg is NP-complete.

<u>Proof:</u>

- Let ϕ be a 3CNF formula in $x_1, \dots, x_n$.
- We will convert each clause in ϕ to a linear inequality in $x_1, \dots, x_n$.
- Eg. convert the clause $(x_1 \vee \bar{x_2} \vee x_3)$ to:

$$\begin{cases} x_1 + (1 - x_2) + x_3 \geq 1 \\ 0 \leq x_1 \leq 1, \ 0 \leq x_2 \leq 1, \ 0 \leq x_3 \leq 1. \end{cases}$$

- Clearly, $3SAT \leq_p$ IntProg. □

- This conversion of a boolean CNF to an algebraic polynomial is called arithmetization.

- Another way to arithmetize CNF:

Proposition: $\text{Quad}Eqn_2 := \{S \mid S$ is a system of quadratic equations modulo 2 & S has a root $\}$ is NP-complete.

Proof:

- Since, given a point $x \in \mathbb{F}_2^n$ it is easy to verify whether it is a root of S, we have $\text{Quad}Eqn \in NP$.

- For any 3CNF ϕ we now convert each clause to a quadratic system (mod 2).

- Eg. the clause $(x_1 \vee \bar{x}_2 \vee x_3)$ becomes:

$$\begin{cases} (1-x_1)z = 0 \pmod 2 \\ z = x_2(1-x_3) \pmod 2 \end{cases}$$

- The clause is true iff the quadratic system has a root.

- Similarly, ϕ is satisfiable iff the corresponding quadratic system has a root.

$\Rightarrow$ $3SAT \leq_p Quad Eqn_2$. $\square$

- Exercise: How about $Quad Eqn_1$?