LOGIC REDUCTION

$(R \vee W) \wedge \overline{(R \wedge W)} = (\bar{R} \wedge W) \vee (R \wedge \bar{W})$ (xor)

$(R \vee W) \wedge (\bar{R} \vee \bar{W})$

or-form

and-form

TRUTH TABLE

tautology } true

falasy / contradiction } false

LOGICAL EQUIVALENCE

$R \rightarrow W \equiv \neg R \vee W$

$\neg\neg R \equiv R$

$(R \vee W) \wedge S \equiv (R \wedge S) \vee (W \wedge S)$

$(R \wedge W) \vee S \equiv (R \vee S) \wedge (W \vee S)$

} distributive laws.

$\neg(A \vee B) \equiv \neg A \wedge \neg B$

$\neg(A \wedge B) \equiv \neg A \vee \neg B$

} demorgan laws.

L.HS [illegible] (R

$LHS = (R \vee W) \wedge \overline{(R \wedge W)}$

$= (R \wedge \overline{(R \wedge W)}) \vee (W \wedge \overline{(R \wedge W)})$ (distributive law)

$= (R \wedge (\bar{R} \vee \bar{W})) \vee (W \wedge (\bar{R} \vee \bar{W}))$

$= (R \wedge \bar{W}) \vee (W \wedge \bar{R})$

$RHS = (\bar{R} \wedge W) \vee (R \wedge \bar{W})$

$= (R \wedge \bar{W}) \vee (\bar{R} \wedge W)$

$= (R \wedge \bar{W}) \vee (W \wedge \bar{R})$ ✓

$\equiv$

COMPLETE SET OF CONNECTIVES, OPERATORS

- or, not
- and, not

× • or, and (w

not a complete set

$a \wedge b = \neg\neg(a \wedge b) = \neg(\neg a \vee \neg b)$

$a \vee b = \neg\neg(a \vee b) = \neg(\neg a \wedge \neg b)$

show it can be generated.

- nor (not-or) is complete ($\downarrow$)
- nand (not-and) is complete ($\uparrow$)

$$\neg X = X \downarrow X$$

$$X \vee Y = (X \downarrow Y) \downarrow (X \downarrow Y)$$

$$X \wedge Y = \neg\neg(X \wedge Y) = \neg(\neg X \vee \neg Y)$$
$$= \neg((X \downarrow X) \vee (Y \downarrow Y))$$
$$= ((X \downarrow X) \downarrow (Y \downarrow Y)$$

REPRESENTATIONS

- DNF disjunctive normal form
- CNF conjunctive normal form

DNF

$(\bar{R} \wedge W) \vee (\bar{R} \wedge \bar{W}) \vee (R \wedge W)$

connected with ORs.

CNF

$(\bar{R} \vee W) \wedge (\bar{R} \vee \bar{W}) \vee (R \wedge W)$

connected with ANDs

APPLICATION OF LOGIC - CIRCUIT DESIGN

HALF ADDER

Coday

|

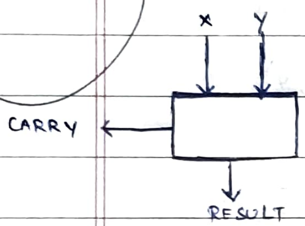
logic

|

P

```
  0 1 1        0 1 1
  1 0 0        1 1 0
  -----      -------
  1 1 1      1 0 0 1
```

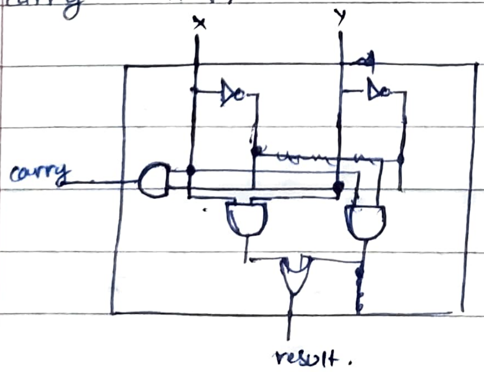
→ half adder.



X	Y	result	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

result $= X \oplus Y = (\neg X \wedge Y) \vee (X \wedge \neg Y)$

carry $= X \wedge Y$



FULL ADDER ?

SUBTRACTION →

2's complement with carry set to 1

APPLICATION

① algorithms satisfiability

② resolution gives a way to prove (PROLOG)

REDUCTIONS

$A \leq B$ A can be reduced to B

A = harder problem

Satisfiability problem -> CNF

$$(x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee w \vee v) \wedge (u \vee \bar{z} \vee \bar{y})$$

5 variables 3 clauses CNF

for which assignment the expression is true

e.g. X = T U = T

not a unique solution

want to do exhaustively -> truth tables

general satisfiability problem $\leq$ 3 sat problem

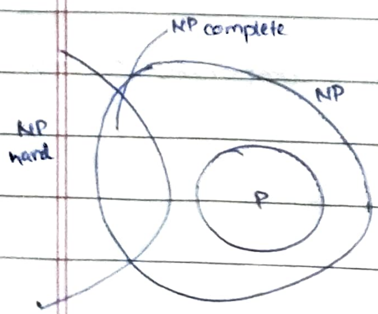
no. limit to length of clause length of clause = 3

no. of clauses

$(2^n \times m)$?

n = no. of variables

m = no. of clauses.



P - polynomial time problems

NP - non-deterministic polynomial time problems

NP -> checking polynomial time

guessing much more time

NP-complete problems

all problems (hard) can be converted to polynomial problems.

NP- complete example

SAT $\leq$ 3-SAT

(i) $(x) \leftrightarrow (x \vee a \vee b) \wedge (x \vee \bar{a} \vee b)$
$\wedge (x \vee a \vee \bar{b}) \wedge (x \vee \bar{a} \vee \bar{b})$

(ii) $(y \vee z) \leftrightarrow (y \vee z \vee c) \wedge (y \vee z \vee \bar{c})$

(iii) $(u \vee v \vee w \vee x \vee y \vee z)$

link variable

$\updownarrow$

$(u \vee v \vee d) \wedge (\bar{d} \vee w \vee e) \wedge (\bar{e} \vee x \vee f)$
$\wedge (\bar{f} \vee y \vee z)$

SAT $\leq$ 3SAT

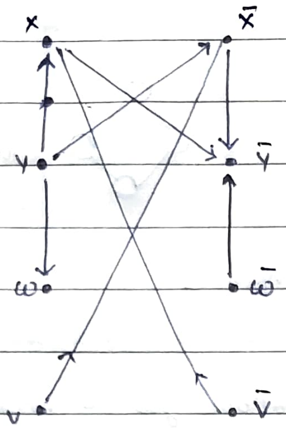
2SAT : known easy problem
(finding strongly connected components in graph)

DFS (O (# of edges))

$(x \vee \bar{y}) \wedge (v \vee x) \wedge (w \vee \bar{y}) \wedge (\bar{x} \vee \bar{y})$

✓ $x = T, \; y = F$

graph : components - solution.



not of one variable connect to another variable with directed arrow

$X \vee \bar{Y}$

$G\ \bar{X} \rightarrow \bar{Y}$, $Y \rightarrow X$

$V \vee X$

$G\ \bar{V} \rightarrow X$, $\bar{X} \rightarrow V$

arrows → implications

reduction of logical satisfiability problem to a graph problem

has both x and $\bar{x}$ (or others) as part of

rule: (not) of one variable and connect to the other variable.

2 SAT is not satisfiable if any strongly connected component has both X and $\bar{X}$.