Functional Completeness

In "Boolean Algebra", we have seen many operations : {AND, OR, NOT, NAND, NOR, Exor, Exnor}

Using these operations, we can write boolean expression of any boolean function.

But Do we need all these operations to write expression of any boolean function??

What set of boolean operations is "sufficient" to write expression of every boolean function?

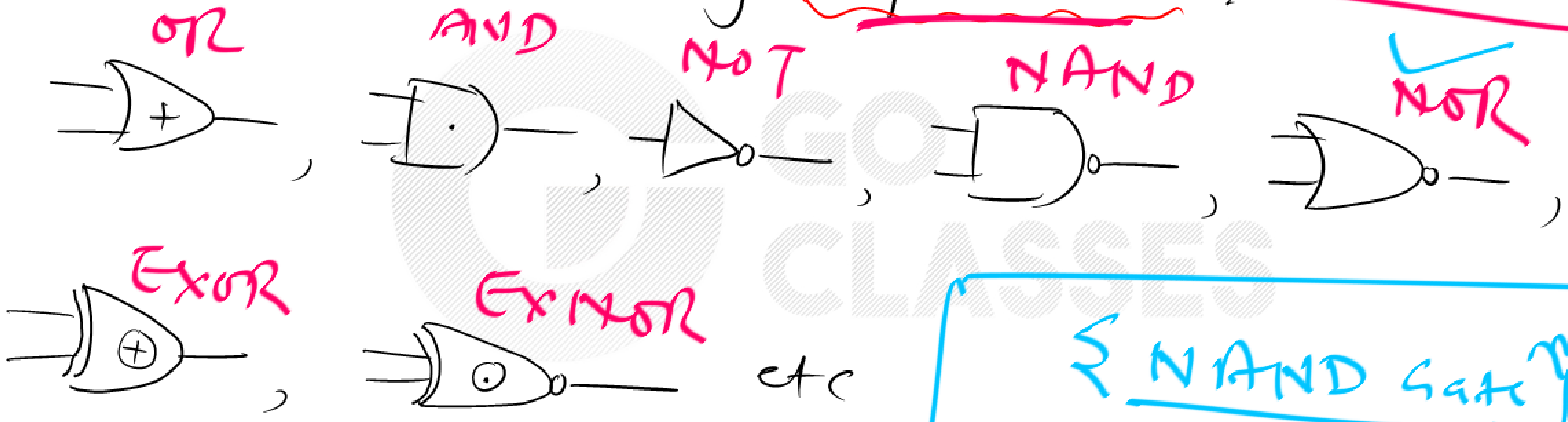
Such set is called "functional Complete" set of boolean expression.

Eg: {AND, OR, NOT} ; {NAND} ; {OR, NOT}

{AND, NOT} ; {NOR} etc.

In "Digital Circuits",

we have seen many logic Gates :



etc

{NAND Gate}

{NOR}

Do we need all these gates to implement any Digital circuit ?

Which set of logic Gates are enough to implement any Digital Circuit ?

Such set of gates is called "functional Complete" set of gates.

In Set Theory :

$(A \Delta B) - C$

Set operations : $\{ \cup, \cap, \bar{S}, A-B, A \Delta B \}$ etc

But $\{ \cup, \cap, \bar{S} \}$ is enough to express any Set theory expression.

$\{ \cup, \text{Complement} \}$ is also enough to express any set theory expression.

In Propositional logic:

logical Connectives: $\{\underbrace{\wedge, \vee, \neg, \oplus, \leftrightarrow, \rightarrow, \uparrow}, \downarrow\}$

functionally Complete sets:

$\{\wedge, \vee, \neg\}$ ✓

$\{\rightarrow, \neg\}$ ✓

$\{\downarrow\}$ ✓

$\{\wedge, \neg\}$ ✓

$\{\vee, \neg\}$ ✓

$\{\uparrow\}$ ✓

$\{\underbrace{\oplus, T, \wedge}\}$ ✓

Functional Completeness

- A set of operations is *functionally complete* if every Boolean function is equivalent to a Boolean expression that uses only operations in the set.

Functional Completeness

- A set of operations is *functionally complete* if every Boolean function is equivalent to a Boolean expression that uses only operations in the set.

- **{Addition, Multiplication, Complement}** is a functionally complete set:
  - For any input/output table, express the function with a sum of minterms expression
  - The sum of minterms expression uses only Addition, Multiplication, and Complement operations.

In Bool. Alg:

$$\{\underbrace{\text{AND, OR, NOT}}\}$$

a	b	$f = a \oplus b =$
0	0	0
0	1	1
1	0	1
1	1	0

$$f = \bar{a}b + a\bar{b}$$

OR

Not

AND

Minterm and Maxterm Expansions

Bool. fun $\longrightarrow$ Truth table $\longrightarrow$ Expression of f in terms of AND, OR, NOT.

$f(a,b,c)$

a	b	c	f
0	0	0	
0	0	1	
⋮	⋮	⋮	
1	1	1	

8 rows

$f = \sum m(f=1)$

$f = \prod M(f=0)$

f = summation of those minterms for which function value f = 1.

$$\underbrace{f = \sum m(f=1)}$$

$\underbrace{f(a,b,c)}$

minterms: $\underline{abc}$, $\underline{\bar{a}bc}$, $ab\bar{c}$, $\bar{a}b\bar{c}$, ...

min term : $\rightarrow$ {AND, Not} ✓

f = Summation of minterms

Summation $\Downarrow$ OR

minterms $\Downarrow$ {AND, Not}

$\rightarrow$ ✓ {OR, AND, Not} ✓

f = Product of maxterms (where f = 0)

AND

{OR, NOT}

$f = \prod M(f=0)$

{OR, NOT}

Maxterm: Sum term consisting of all Variables.

$f(a,b,c)$; $\bar{a} + \bar{b} + c$; $a+b+c$

- Minterm expansion (or called **standard sum of products**)

$$f = A'BC+AB'C'+AB'C+ABC'+ABC$$
$$= m_3 + m_4 + m_5 + m_6 + m_7 \quad \text{(m-notation)}$$
$$= \sum m(3, 4, 5, 6, 7)$$

- Maxterm expansion (or called **standard product of sums**)

$$f = (A+B+C)(A+B+C')(A+B'+C)$$
$$= M_0M_1M_2 \quad \text{(M-notation)}$$
$$= \prod M(0, 1, 2)$$

	ABC	f
0	000	0
1	001	0
2	010	0
3	011	1
4	100	1
5	101	1
6	110	1
7	111	1

$$f = \sum m(f=1) = \sum m(\underbrace{3, 4, 5, 6, 7})$$

$$f = \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$$

$$f = \prod M(f=0) = \prod M(0, 1, 2)$$

$$= (A+B+C)(A+B+\bar{C})(A+\bar{B}+C)$$

3 = 011
4 = 100
5 = 101
6 = 110
7 = 111

0 = 000
1 = 001
2 = 010

Any boolean function:

Can be Expressed using { AND, NOT, OR }

this set is fun. Complete.

ABC	f	f'
000	0	1
001	0	1
010	0	1
011	1	0
100	1	0
101	1	0
110	1	0
111	1	0

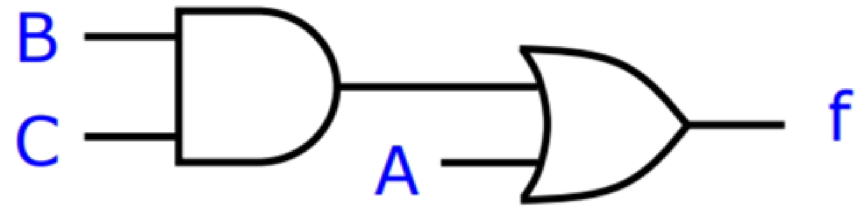
$$f = A'BC + AB'C' + AB'C + ABC' + ABC$$

(minterm expansion)

$$= A'BC + AB' + AB$$

$$= A'BC + A$$

$$= BC + A$$



ABC	f	f'
000	0	1
001	0	1
010	0	1
011	1	0
100	1	0
101	1	0
110	1	0
111	1	0

$f = (A+B+C)(A+B+C')(A+B'+C)$

(maxterm expansion)

$= (A+B)(A+B'+C)$

Alternative derivation by DeMorgan's law

$f' = A'B'C'+A'B'C+A'BC'$

$f = (A'B'C'+A'B'C+A'BC')'$

$= (A+B+C)(A+B+C')(A+B'+C)$

- The sets $\{\neg, \vee\}$ and $\{\neg, \wedge\}$ are functionally complete.

{ AND, OR, Not } ✓

{ AND, Not } ✓

{ OR, Not } ✓

{AND, Not} → fun. Comp.

$$a+b = \overline{\bar{a}\,\bar{b}}$$

$$a+b = \overline{\bar{a}\,\bar{b}}$$

OR = f(AND, NOT)

$$ab+c = \overline{\overline{ab}\;\bar{c}}$$

$$E+F = \overline{\bar{E}\,\bar{F}}$$

Eliminating Addition

An addition operation can be replaced by applying De Morgan's Law:

$$x + y = \overline{\overline{x + y}} = \overline{\bar{x}\bar{y}}$$

Can use this rule to eliminate any addition operation in a Boolean expression:

$a + b = \overline{\bar{a}\bar{b}}$

$(x + y)z = (\overline{\bar{x}\bar{y}}) \cdot z$

$f = (x + y)z = (\overline{\bar{x}\bar{y}})z$

$\underset{a}{xy} + \underset{b}{z} = \overline{\overline{xy}\,\bar{z}}$

$\{AND, OR, NOT\}$ ✓

function $g \Rightarrow f(AND, OR, NOT)$ ✓

$\Rightarrow f(AND, h(AND, NOT), NOT)$

$\{AND, NOT\}$ ✓

Eliminating Addition

An addition operation can be replaced by applying De Morgan’s Law:

$$x + y = \overline{\overline{x + y}} = \overline{\bar{x}\bar{y}}$$

The set **{Multiplication, Complement}** is functionally complete:

Input/Output table defining a Boolean functions → Equivalent Boolean expression that is a sum of minterms → Equivalent Boolean expression with no addition operations – only **{Mult, Comp}**

Eliminate addition operations using De Morgan’s Law function $f \Rightarrow$ Truth Table $\Rightarrow f = \underbrace{\sum m(f=1)}$

f = summation of minterms

$f = m_i + m_j + m_k + m_p = \{\text{AND, OR, NOT}\}$

$f = \overline{\left(\overline{m_i}\ \overline{m_j}\ \overline{m_k}\ \overline{m_p}\right)} = \{\text{AND, NOT}\}$

$$f(x,y) = \underbrace{\bar{x}y}_{E} + \underbrace{x\bar{y}}_{F}$$

$$E+F$$

$$= \overline{\bar{E}\,\bar{F}}$$

$$\overline{\overline{f(x,y)}} = \underbrace{\overline{\overline{\bar{x}y}\;\overline{x\bar{y}}}}$$

$\{AND, NOT\}$

Eliminating Addition

x	y	f(x,y)
0	0	0
0	1	1
1	0	1
1	1	0

$$f(x, y) = \underbrace{\bar{x}y}_{a} + \underbrace{x\bar{y}}_{b}$$

$$\overline{\overline{\bar{x}y} \cdot \overline{x\bar{y}}}$$

$$a + b = \overline{\bar{a}\,\bar{b}}$$

$$f = x(\overline{y} + \overline{z}) + \overline{x y}$$

$$= \underbrace{x(\overline{y z})}_{\alpha} + \underbrace{\overline{x y}}_{\beta}$$

$$= \overline{\left(\overline{x(\overline{y z})}\ \overline{\overline{x y}}\right)}$$

$$= \overline{\overline{x\, \overline{y z}}\ x y}$$

$$\alpha + \beta$$

$$= \overline{\overline{\alpha}\,\overline{\beta}}$$

$$\overline{y} + \overline{z}$$

$$= \overline{\overline{\overline{y}}\,\overline{\overline{z}}}$$

$$= \overline{y z}$$

Eliminating Addition Operations

Select the expression that is equivalent to:

$x\overline{yz} + \overline{xy}$ $\qquad \overline{\overline{x\overline{yz}}\ xy}$

$$f = x(\bar{y} + \bar{z}) + \overline{xy}$$

$x(\overline{\bar{\bar{y}}\bar{\bar{z}}})$

A) $\overline{x\,\bar{y}\,\bar{z}} \cdot xy$

C) $\overline{x\,\overline{yz}} \cdot xy$

B) $\overline{x\,\bar{y}\,\bar{z}} \cdot \overline{xy}$

D) $\overline{x\,\overline{yz}} \cdot \overline{xy}$

E) None of the above.

{AND, NOT}

{OR, NOT}

$f =$

$\alpha\beta = \overline{\overline{\alpha} + \overline{\beta}}$

$$\boxed{ab = \overline{\overline{a} + \overline{b}}}$$

De Morgan law

$a\overline{c}$

$= \overline{(\overline{a} + \overline{\overline{c}})}$

$= \overline{\overline{a} + c}$

More Functionally Complete Sets

Is the set {Addition, Complement} functionally complete?

Can eliminate Multiplication operations using the other version of De Morgan's Law: $xy = \overline{\overline{xy}} = \overline{\overline{x} + \overline{y}}$

Can use this rule to eliminate any addition operation in a Boolean expression:

$(x + y)z =$ $\overline{\overline{x+y} + \overline{z}}$

$xy + z =$ $\overline{\overline{x} + \overline{y}} + z$

$\{OR, NOT\}$ ✓

$\alpha\beta = \overline{\bar{\alpha} + \bar{\beta}}$

$f = \underbrace{(x+y)}_{\alpha} \cdot \underbrace{z}_{\beta} = \overline{\overline{x+y} + \bar{z}}$ ✓

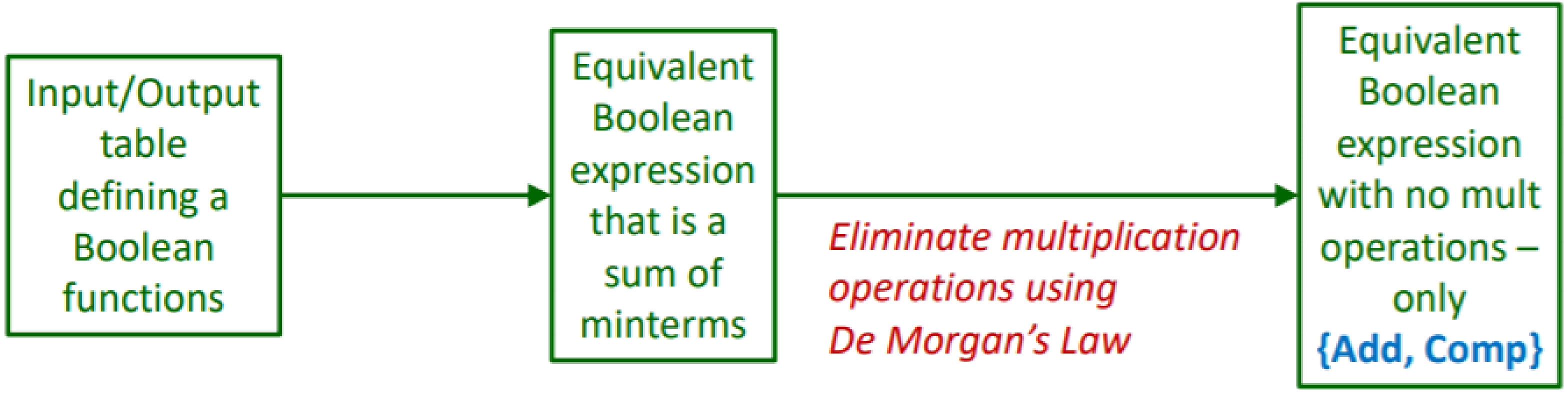
$f = x \cdot y + z$

$= \overline{(\bar{x} + \bar{y})} + z$ ✓

More Functionally Complete Sets

Is the set {Addition, Complement} functionally complete?

Can eliminate Multiplication operations using the other version of De Morgan's Law: $xy = \overline{\overline{xy}} = \overline{\overline{x} + \overline{y}}$



$$f = \bar{x}\cdot y + x\cdot\bar{y}$$

$$\alpha\beta = \overline{\bar{\alpha} + \bar{\beta}}$$

$$= \overline{(\bar{\bar{x}} + \bar{y})} + \overline{(\bar{x} + \bar{\bar{y}})}$$

$$f = \overline{(x + \bar{y})} + \overline{(\bar{x} + y)}$$

{OR, Not}

Eliminating Multiplication

x	y	f(x,y)
0	0	0
0	1	1
1	0	1
1	1	0

$$f(x, y) = \bar{x}y + x\bar{y}$$

$$\overline{x+\bar{y}} + \overline{\bar{x}+y}$$

Eliminating Multiplication Operations

Select the expression that is equivalent to:

$$\overline{\bar{x} + (\overline{\bar{y}+\bar{z}})} + \overline{\bar{x}+\bar{y}}$$

$$f = x(\bar{y} + \bar{z}) + \overline{xy}$$

A) $\overline{\bar{x} + (\overline{y+z})} + (\overline{\bar{x}+\bar{y}})$

E) None

B) $\overline{\bar{x} + (\overline{y+z})} + \bar{x} + \bar{y}$

C) $\overline{\bar{x} + (\overline{\bar{y}+\bar{z}})} + \bar{x} + \bar{y}$

D) $\overline{\bar{x} + \bar{y} + \bar{z}} + \bar{x} + \bar{y}$

$$f = x \cdot (\overline{y} + \overline{z}) + \overline{x \cdot y}$$

$$= \overline{\overline{x} + \overline{(\overline{y} + \overline{z})}} + \overline{\overline{\overline{x} + \overline{y}}}$$

$$f = \overline{(\overline{x} + \overline{\overline{y} + \overline{z}})} + \overline{x} + \overline{y}$$

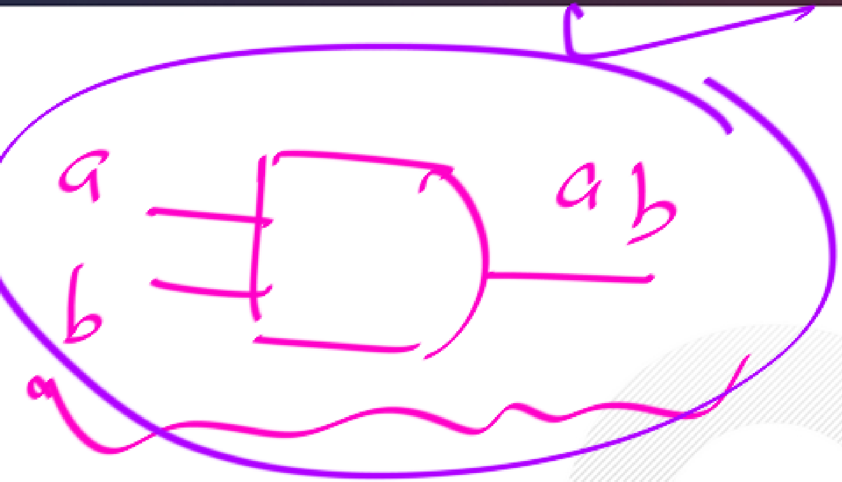
{OR, NOT} ✓

{AND, NOT} ✓

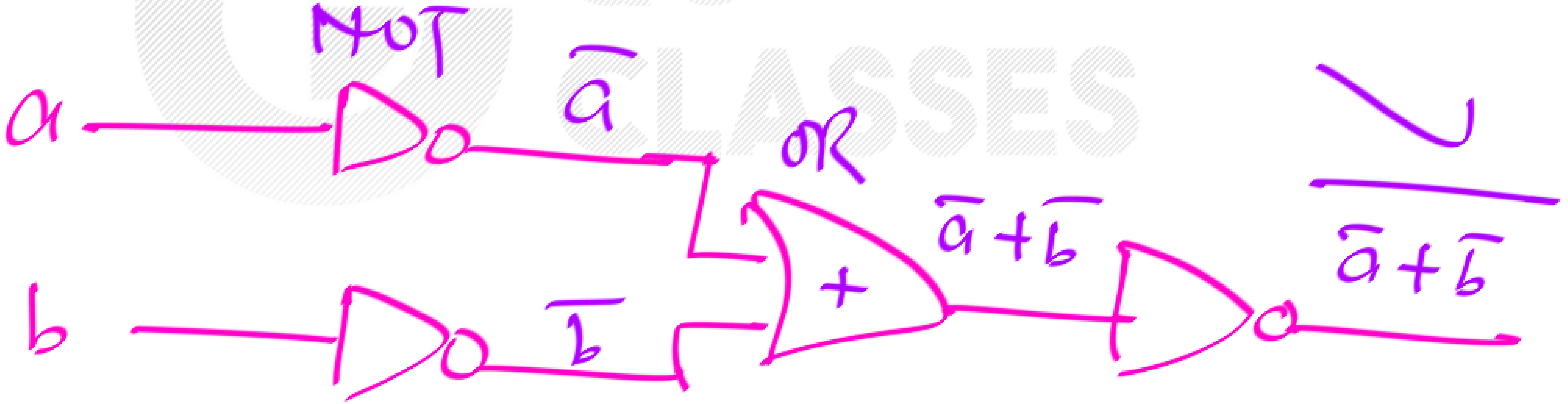
{AND, OR, NOT} ✓

{OR gate, Not gate}

fun. Comp set for Dig. Circuits.



$$ab = \overline{\bar{a} + \bar{b}}$$



{NAND gate}

NAND Gate is a Universal Gate:

To prove that any Boolean function can be implemented using only NAND gates, we will show that the AND, OR, and NOT operations can be performed using only these gates.

Universal Gates:

A universal gate is a gate which can implement any Boolean function without need to use any other gate type.

The NAND and NOR gates are universal gates.

In practice, this is advantageous since NAND and NOR gates are economical and easier to fabricate and are the basic gates used in all IC digital logic families.

NAND op^n : {AND, NOT}

a NAND b = $a \uparrow b = \overline{(ab)}$

NAND $\equiv \uparrow$

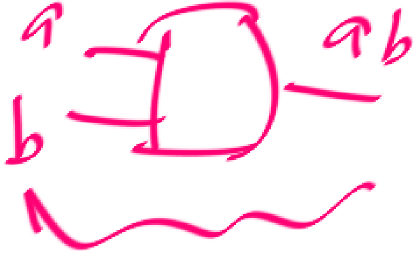
NAND = NOT of AND

AND: $ab = (a \uparrow b) \uparrow (a \uparrow b)$

$ab = \overline{(ab)} \uparrow \overline{(ab)}$

$= \overline{\overline{(ab)}\;\overline{(ab)}} = \overline{\overline{ab}} = ab$ ✓

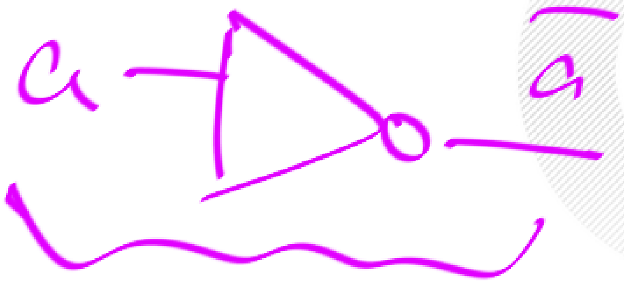
$$\boxed{\underline{\underline{ab}} = (a \uparrow b) \uparrow (a \uparrow b)}$$



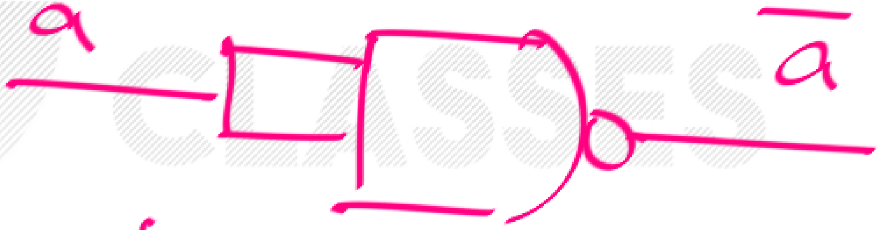
AND Gate

two NAND gates

$$\underbrace{\overline{a} = (a \uparrow a)} = \overline{a \cdot a} = \overline{a}$$



NOT Gate



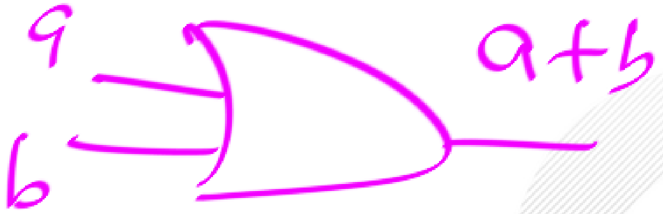
NAND gate

NAND gate $\Rightarrow$ { AND ; NOT }

NAND gate

fun. Comp.

OR GATE:



OR gate

$\overline{a} = a \uparrow a$

$\alpha\beta = (\alpha \uparrow \beta) \uparrow (\alpha \uparrow \beta)$

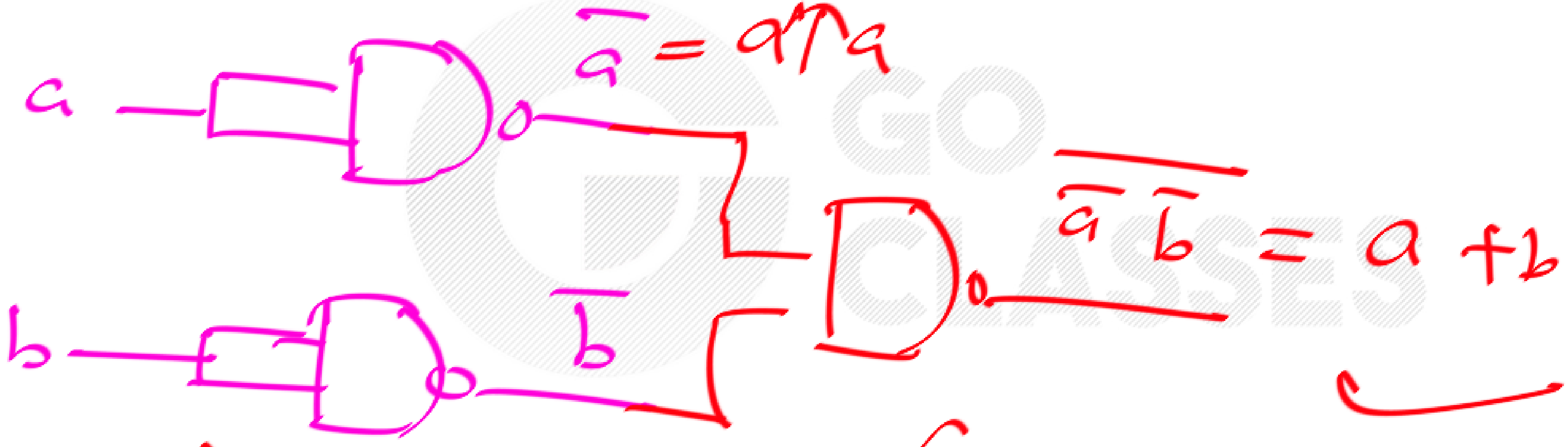
$a+b = \overline{\overline{a}\;\overline{b}}$

$= \overline{(a \uparrow a) \cdot (b \uparrow b)}$

$= [(a \uparrow a) \uparrow (b \uparrow b)] \uparrow [(a \uparrow a) \uparrow (b \uparrow b)]$

$a+b = \overline{\overline{a}\,\overline{b}}$

OR gate



3 NAND gates

$$\boxed{a+b = (a \uparrow a) \uparrow (b \uparrow b)}$$

$$= \bar{a} \uparrow \bar{b}$$

$$= \overline{\bar{a}\,\bar{b}} = a+b$$

{OR, Not} ✓

Dis ckt

{NAND} ✓

Express $\overline{x + y}$ using only NAND operations.

- First eliminate addition: a+b = $\overline{\bar{a}\bar{b}}$
- The eliminate multiplication: ab = (a↑b) ↑ (a↑b)
- Then eliminate compliment: $\bar{a}$ = a↑a

$$\overline{x+y} = \underbrace{\bar{x}}_{a}\,\underbrace{\bar{y}}_{b} = (\bar{x} \uparrow \bar{y}) \uparrow (\bar{x} \uparrow \bar{y})$$

$$= ((x \uparrow x) \uparrow (y \uparrow y)) \uparrow ((x \uparrow x) \uparrow (y \uparrow y))$$

NOR Gate is a Universal Gate:

To prove that any Boolean function can be implemented using only NOR gates, we will show that the AND, OR, and NOT operations can be performed using only these gates.

NOR gate

NAND gate

a NOR b = $a \downarrow b = \overline{a+b}$

NOR = NOT of OR

$a \downarrow b = \overline{a+b}$

{NOR} fun. Comp.

NOT opn:

$$\overline{a} = a \downarrow a = \overline{a}$$ ✓

$$\overline{\alpha} = \alpha \downarrow \alpha = \overline{\alpha + \alpha} = \overline{\alpha}$$ ✓

OR opⁿ:

$$\boxed{\alpha+\beta = (\alpha\downarrow\beta)\downarrow(\alpha\downarrow\beta)}$$

$$\underline{\underline{a+b}} = (a\downarrow b)\downarrow(a\downarrow b)$$

$$(\overline{a+b})\downarrow(\overline{a+b})$$

$$= \overline{\overline{a+b}+\overline{a+b}} = \overline{\overline{a+b}} = a+b$$

AND opⁿ:

$ab = \overline{\overline{a}+\overline{b}}$

$$\boxed{ab = (a \downarrow a) \downarrow (b \downarrow b)}$$

$\overline{a} \downarrow \overline{b} = \overline{\overline{a}+\overline{b}} = ab$

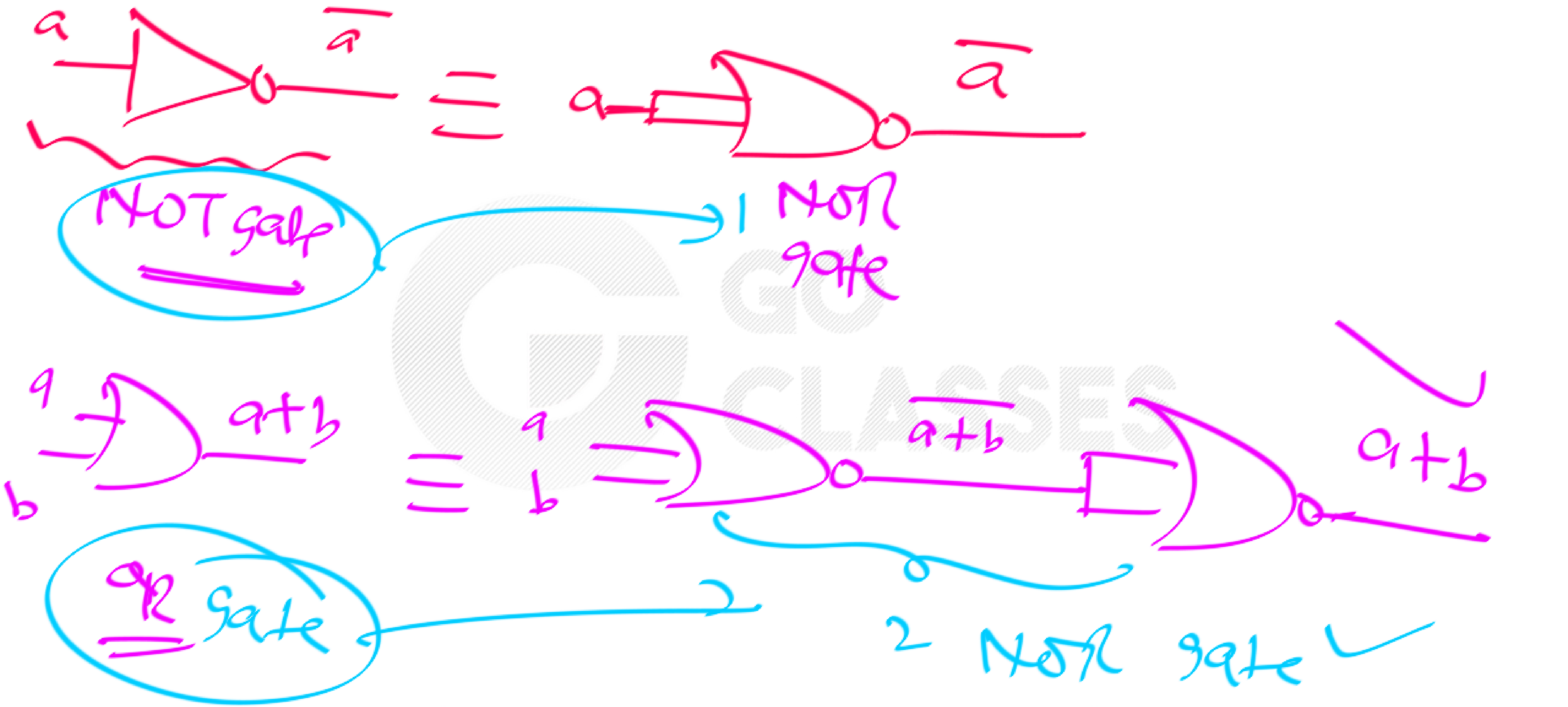
$$ab = (a \downarrow a) \downarrow (b \downarrow b)$$

$$a+b = (a \downarrow b) \downarrow (a \downarrow b)$$

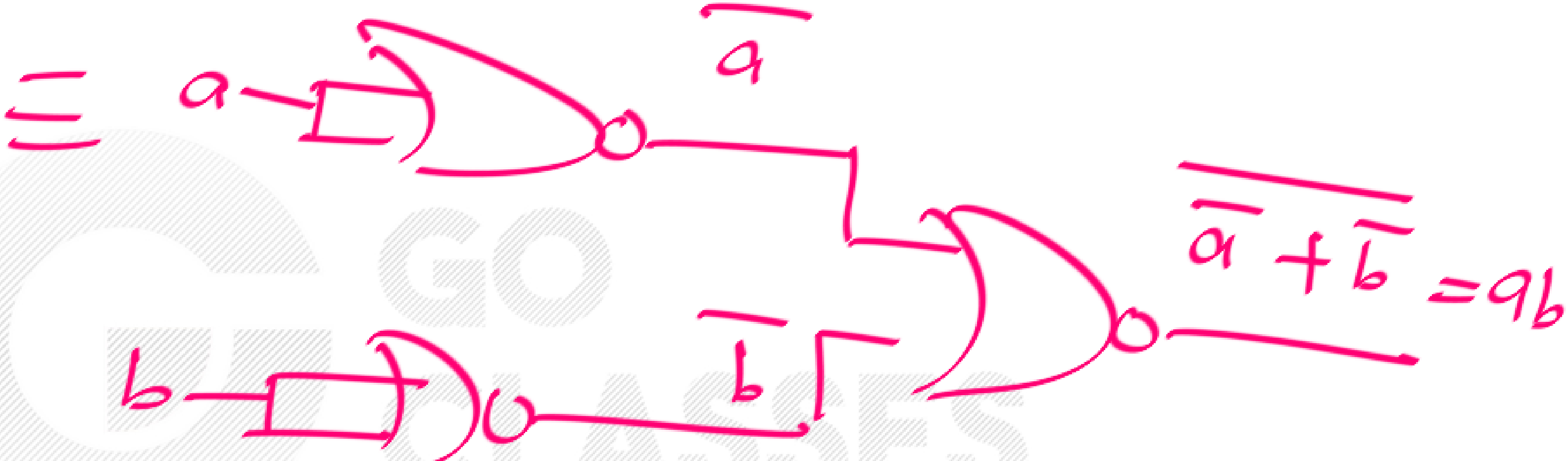
$$\bar{a} = a \downarrow a$$

$\{NOR\} \Rightarrow$ fun. cpl.

NOR gate = universal gate


a
$\bar{a}$
$\equiv$
a
$\bar{a}$
NOT gate
1 NOR gate
a
b
a+b
$\equiv$
a
b
$\overline{a+b}$
a+b
OR gate
2 NOR gate

AND gate



AND

3 NOR gate

Proving Functional Completeness

Which set of equations prove that $\{\downarrow\}$ is functionally complete?

A) $x+y = (x\downarrow y)\downarrow(x\downarrow y)$

$\bar{x} = x\downarrow x$

B) $x+y = (x\downarrow y)\downarrow(x\downarrow y)$

$\overline{x\downarrow y} = x+y$

C) $x\downarrow y = \overline{x+y}$

$\overline{x\downarrow y} = x+y$

D) $x+y = (x\downarrow y)\downarrow(x\downarrow y)$

$xy = (x\downarrow x)\downarrow(y\downarrow y)$

{NOR opn} → {OR, NOT}

or

{AND, NOT}

{AND, OR} set is NOT fun. Comp.

Given set of boolean functions :

$S = \{ F_1, F_2, F_3, \dots, F_n \}$

Is S fun. Complete ?

Eg: $S = \{ \text{AND}, \text{OR} \} = \{ a+b, ab \}$

Guidelines to check if a set of boolean functions is functionally Complete or not?

① minimize the expressions if possible.

② If "for all" functions in the set, $f(0)=0$ then set is Not functionally Complete.

3. If "for all" functions, $f(1) = 1$ then set is Not functionally Complete.

4. If every function is self Dual then set is Not functionally Complete.

5. If all above guidelines do not apply then Try to create "Not", "AND" "OR" operation.

$S = \{F_1, F_2, F_3\}$ Is S fun Comp?

min $\downarrow$ min $\downarrow$ min $\downarrow$

$S = \{G_1, G_2, G_3\}$

(2) If $G_1(0) = 0$; $G_2(0) = 0$; $G_3(0) = 0$

then S is NOT func. Comp.

(3) If $G_1(1) = 1$; $G_2(1) = 1$; $G_3(1) = 1$

then S is NOT fun. Comp.

(4) If $G_1 = G_1^d$; $G_2 = G_2^d$; $G_3 = G_3^d$

then S is Not func. Comp.

Note: Guideline 2, 3, 4 are only one way.

Try to create NOT opⁿ;

AND AND opⁿ OR OR opⁿ

$S = \{ \text{AND}, \text{OR} \} = \{ a+b, ab \} \Rightarrow$ Not fun. Comp.

Guideline 2:

$0+0=0$
$0 \cdot 0 = 0$

$\forall f \in S \; (f(0) = 0)$ then S Not fun Comp.

Guidelines; set S is fun. Comp or not

① minimization ✓ $S = \{f_1, f_2, f_3\}$

② $\forall f \in S \; (f(0) = 0)$ then S

is NOT func. Comp.

Is the set **{Addition, Multiplication}** functionally complete?

No…can’t express $\overline{x}$ using only Addition or Multiplication:

$$x + x = x$$

$$x \cdot x = x$$

③ $\forall_{f \in S} (f(1) = 1)$ then S is Not fun. Comp.

④ $\forall_{f \in S} (f^d = f)$ then S is Not fun. Comp.

⑤ Try to create Not opⁿ; AND opⁿ; OR opⁿ.

None of Multiplication, Addition, or Complement is functionally complete by itself.

$S = \{AND\} = \{ab\} \rightarrow$ Not func. Comp.

Guideline 2:

$f = ab$

$f(0) = 0 \cdot 0 = 0$

$\forall_{f \in S} \; (f(0) = 0)$ so S is Not fun. Comp.

$S = \{NOT\} \rightarrow$ Not fun. Comp.

$S = \{ f(a) = \bar{a} \}$

$f(0) = \bar{0} = 1 \neq 0$

$f(1) = \bar{1} = 0 \neq 1$

So C. 2, 3 Do not Apply.

$S = \{ \text{NOT} \} = \{ f(a) = \bar{a} \}$

$f = \bar{a}$

$f^d = \bar{a}$

"NOT op" is self Dual.

S is Not fun. Comp.

$S = \{ XOR, OR \} = \{ a \oplus b, a + b \} \Rightarrow$ Not fun. Comp.

Guideline 2 : $f_1(a, b) = a \oplus b$

$f_1(0, 0) = 0 \oplus 0 = 0$

$f_2(a, b) = a + b$

$f_2(0, 0) = 0 + 0 = 0$

$S = \{ XOR, AND \} = \{ a \oplus b, ab \}$

$0 \oplus 0 = 0$
$0 \cdot 0 = 0$

Not fun. Comp.

Using Guideline?

$S = \{ \overline{1}, a, b, \overline{a}, \overline{b} \}$

Guideline 2, 3 Do not Apply.

$\overline{a}$

$\left.\begin{array}{l} \overline{0} \neq 0 \\ \overline{1} \neq 1 \end{array}\right\}$

Guideline 4

Applies

$f = a$; $f^d = a$

$f = b$; $f^d = b$

$f = \overline{a}$; $f^d = \overline{a}$

$f = \overline{b}$; $f^d = \overline{b}$

"Not" is self Dual.

$\forall_{f \in S} (f = f^d)$

So S is Not fun. Comp.

$S = \{ \text{XNOR}, \text{OR} \} = \{ a \odot b, a + b \}$

$\left\{ \begin{array}{l} 1 \odot 1 = 1 \\ 1 + 1 = 1 \end{array} \right\}$ Guideline 3 Applies

So $S \Rightarrow$ Not fc.

$S = \{ \text{XNOR}, \text{AND} \} = \{ a \odot b, ab \}$

$$\left\{ \begin{array}{l} 1 \odot 1 = 1 \\ 1 \cdot 1 = 1 \end{array} \right\} \quad \underbrace{\forall_{f \in S} \left(f(1) = 1 \right)}$$

So S Not fc.

$a \rightarrow b \equiv a' + b$

Prove that the set {→, ¬} is functionally complete

$\{\rightarrow, \neg\}$

a	b	$a \to b$	$a' + b$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	1	1

b

$$\underbrace{a \to b \equiv a' + b}$$

$\{\rightarrow, \text{complement}\} \rightarrow$ fun. Comp.

$\{\rightarrow, \neg\}$

$\{\rightarrow, \text{Not}\}$

$\alpha \rightarrow \beta = \alpha' + \beta$

$a + b = a' \rightarrow b$

$= a + b$

$\{\rightarrow, \text{Not}\}$ fun. Comp.

$S = \{\rightarrow, \neg\} = \{a \rightarrow b, \bar{a}\}$

(1) minimize each function

(2) $\forall_{f \in S} \left(\underline{f(0) = 0} \right)$ then S is Not func Comp.

$\underbrace{0 \rightarrow 0} = 1 \neq 0$; $\bar{0} \neq 0$

$\{ a \rightarrow b, \overline{a} \}$

(3) $1 \rightarrow 1 = 1$

$\overline{1} = 0$

(4) $f = \overline{a}$ is self Dual

but $g = a \rightarrow b$ is Not self.

$f = a \rightarrow b = a' + b$

$f_d = \underbrace{a' b} \neq f$

implication function is Not self Dual.

5. $\underbrace{\{\rightarrow, \bar{a}\}}$ $\{\rightarrow, \text{NOT}\}$

OR: $a+b = a' \rightarrow b$

a	b	f
0	0	0
0	1	1
1	0	1
1	1	0

$f = a \oplus b = a \cdot \bar{b} + \bar{a} \cdot b$

{OR, AND, NOT}

$f = \overline{\overline{\bar{a} + b} + \overline{a + \bar{b}}}$

{OR, NOT}

$=$

$\boxed{\alpha\beta = \overline{\bar{\alpha} + \bar{\beta}}}$

$\boxed{x + y = \\ x' \to y}$

$$f = \overline{\bar{a}+b} + \overline{a+\bar{b}}$$

$$\boxed{x+y = x' \to y}$$

$$= \overline{\overline{(a \to b)} + \overline{(b \to a)}}$$

$$\boxed{f = (a \to b) \to \overline{(b \to a)}}$$

$$\{\to, \text{NOT}\}$$

Prove that the set $\{\rightarrow, \neg\}$ is functionally complete

The implication $\rightarrow$ is defined by

$$a \rightarrow b \equiv \neg a \vee b.$$

This means, that

$$\neg a \rightarrow b \equiv a \vee b$$

and thus, you can express logical or using $\rightarrow$ and $\neg$. Furthermore you know de Morgans rules and you have

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

Thus you can express all logical operators using $\rightarrow$ and $\neg$. The set of these is known to be functionally complete.

$$S = \{\rightarrow, \lor\}_0 = \{\bar{a} \rightarrow b, \bar{a} + b\}_0$$

$$\left.\begin{matrix} 1 \rightarrow 1 = 1 \\ 1 + 1 = 1 \end{matrix}\right\}_0 \quad \forall_{f \in S} \, (f(1) = 1)$$

So S is Not FC.

$\{\rightarrow, \text{AND}\}$ is not FC.

$S = \{a \rightarrow b, ab\}$

$$\left.\begin{array}{l} 1 \rightarrow 1 = 1 \\ 1 \cdot 1 = 1 \end{array}\right\}$$

So S is not FC.

$\{\rightarrow, \text{NOT}\}$ fc ✓

$\{\rightarrow, \vee\}$ ✗ fc

$\{\rightarrow, \wedge\}$ ✗ fc

Not functionally complete sets :

{ → , OR }

{Exor, OR}

{Exor, AND}

{ExNor, OR}

$S = \{\rightarrow, 1\}$ is Not fun Comp.

$\{a \rightarrow b, 1\}$

$\{\rightarrow, 0\}$ FC ✓

$\{\rightarrow, 1\}$ ✗ FC

$1 \rightarrow 1 = 1$

1

Guideline 3 Applies

So S is Not FC.

Prove that {F, →} is functionally complete

$S = \{ \rightarrow, 0 \}$

constant 0

$S = \{ a \rightarrow b,\ 0 \}$

① minimize every function

② $\{0 \rightarrow 0\} = 1$ Guideline no. 2

Does not Apply.

$\{a \rightarrow b, 0\} = \{f_1 = a \rightarrow b, f_2 = 0\}$

$1 \rightarrow 1 = 1$

$f_2 \neq 1$

$\{f_2 = 0, f_2 = 1\}$

$$f = a \rightarrow b = a' + b$$

$$f^d = \underbrace{a'b \neq a' + b}$$

$S = \{ a \to b, 0 \}$

$\bar{a}$

$\bar{a}$: $\boxed{a \to 0} = \bar{a}$

$a+b$: $\boxed{(a \to 0) \to b} = a+b$

$(a \to 0) \to b = (a') \to b$

$= a + b$

$\boxed{\alpha \to \beta = \alpha' + \beta}$

$\{\to, 0\}$ is fun. comp.

4.17.1 Functional Completeness: GATE CSE 1989 | Question: 4-iii top

Show that {NOR} is a functionally complete set of Boolean operations.

4.17.4 Functional Completeness: GATE IT 2008 | Question: 1 top

https://gateoverflow.in/3222

A set of Boolean connectives is functionally complete if all Boolean functions can be synthesized using those. Which of the following sets of connectives is NOT functionally complete?

A. EX-NOR
B. implication, negation
C. OR, negation
D. NAND

$A = \{ 0 \} \rightarrow$ Not FC

linear function

$1 \odot 1 = 1$ ✓

$\{\rightarrow, \text{NOT}\}$ $\rightarrow$ $\{\text{NOT}, \text{OR}\}$

FC

$a+b$: $a' \rightarrow b$

We say that boolean function f is linear if one of the following two statements holds for f:

- For every 1-value of f, the number of 1's in the corresponding input is *odd*, and for every 0-value of f, the number of 1's in the corresponding input is *even*.

or

- For every 1-value of f, the number of 1's in the corresponding input is *even*, and for every 0-value of f, the number of 1's in the corresponding input is *odd*.

If one of these statements holds for f, we say that f is linear[1]. We denote the class of linear boolean functions with L and write $f \in L$.

$S = \{F_1, F_2, F_3, F_4\}$

Q: S is fun. Comp. or not?

1. minimize f_i.

2. $\forall_{f \in S} \left(f(0) = 0 \right)$ then S is not FC.

③ $\forall_{f \in S}$ ($f(1) = 1$) then S Not FC.

④ $\forall_{f \in S}$ (f is self Dual) then S is Not FC.

⑤ If "every function" is S is linear then S is Not FC.

If $\forall_{f \in S}$ ($\underline{f \text{ is linear}}$) then S is NOT FC.

linear function:

input	f
Even 1's	1
odd 1's	0
Even 1's	1
odd 1's	0
Even 1's	1
odd 1's	0
Even 1's	1

f linear

OR

f = 1 ⇒ input has odd 1's

AND

f = 0 ⇒ input has even 1's

f linear

Not: $f = \bar{a}$

a	$f = \bar{a}$
0	1
1	0

$f = 1 \Rightarrow$ Even 1's

AND

$f = 0 \Rightarrow$ odd 1's

So f is linear.

OR

$f = a + b$

a	b	f
0	0	0
0	1	1
1	0	1
1	1	1

$f = 1 \not\Leftarrow$ even 1's

$f = 1 \not\Leftarrow$ odd 1's

$f = a + b$ is Not linear.

f is linear:

Input		OR	Input
f = 1 ⟹ even 1's			f = 1 ⟹ odd 1's
AND	f lin.		AND
f = 0 ⟹ odd 1's			f = 0 ⟹ even 1's

f lin.

AND : $f = ab \Rightarrow$ Not linear

input

a	b	ab = f
0	0	0
0	1	0
1	0	0
1	1	1

$f = 1 \Rightarrow$ even 1's

so

$f = 0 \Rightarrow$ odd 1's ✗

XOR $f = a \oplus b$

a	b	$f = a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

input

$f=1 \Rightarrow$ odd 1's

so

$f=0 \Rightarrow$ even 1's

So $f = a \oplus b$ is linear

XNOR $f = a \odot b$

a	b	$f = a \odot b$
0	0	1
0	1	0
1	0	0
1	1	1

input

$f = 1 \Rightarrow$ even 1's

So

$f = 0 \Rightarrow$ odd 1's

So $f = a \odot b$ is linear.

linear : NOT, $\oplus$, $\odot$

Not linear : $+$, and, $\uparrow$, $\downarrow$, $\rightarrow$

NAND $a \uparrow b = f = \overline{ab}$

a	b	f
0	0	1
0	1	1
1	0	1
1	1	0

Not linear.

$f = 1$ ⟹ input

Sometimes even is, Sometimes odd is

NOR ⇒ Not linear ✓

implication: $f = a \to b = a' + b$ → Not linear

a	b	$a \to b = f$
0	0	1 ✓
0	1	1 ✓
1	0	0
1	1	1 ✓

f=1 → input → evenis ~~oddis~~

$S = \{ f_1, f_2, f_3, f_4 \}$

If All f_i are linear ;

then S is NOT FC.

Eg: S = { NOT, EXOR, EXNOR }

$S = \{ NOT, \oplus, \odot \}$

S is FC or not?

$\forall_{f \in S}$ (f is linear) ; so S is NOT FC.

Eg: $S = \{\oplus, \text{AND}\} \rightarrow$ NOT FC

"linearity of function" Doesn't Apply.

Mistake: AND is Not linear so S is FC.

$S = \{\oplus, \text{AND}\} \rightarrow$ NOT FC

$S = (a \oplus b, ab)$

$0 \oplus 0 = 0$

$0 \cdot 0 = 0$

So S is NOT FC.

$S = \{f_i \mid i \geq 1\}$

S is NOT FC if

1. All f_i self Dual.

or

2. All f_i are linear.

~~3~~ $f_i(0) = 0 \; ; \; \forall i$

or

~~4~~ $f_i(1) = 1 \; ; \; \forall i$

minimize every f_i; AND Try to create NOT opn; OR opn/AND

Now, let us check whether our three functions are linear.

x	$\neg x$
0	1
1	0

x_1	x_2	$x_1 \wedge x_2$
0	0	0
0	1	0
1	0	0
1	1	1

x_1	x_2	$x_1 \vee x_2$
0	0	0
0	1	1
1	0	1
1	1	1

In the case of $\neg$, for every 1-value of this function, the number of 1's among the input variables is always even. Actually, there is only one 1-value of $\neg$, and the number of 1's in the corresponding input is 0, which is an even number. Similarly, for every 0-value of $\neg$ (and there is only one such value), the number of 1's in the input is odd (we have the only input variable, and it is 1 in the corresponding row). Thus, $\neg$ is linear.

Using the same reasoning, we can show that neither $\wedge$ nor $\vee$ is linear. $\wedge$ is not linear because on 0 outputs, the number of 1's in the corresponding inputs is sometimes even and sometimes odd. Thus, neither of two conditions from our definition can hold for $\wedge$. Similarly, $\vee$ is not linear because on 1-outputs, the number of 1's in the corresponding inputs is sometimes even and sometimes odd.

Thus, $\neg \in L$, but $\wedge \notin L$ and $\vee \notin L$.

$\{\neg, \oplus\}$ is not complete;

$S = \{ \text{NOT}, a \oplus b \}$ so S is NOT FC.

NOT → linear

$a \oplus b$ → linear

$\{\wedge, \oplus\}$ is not complete either, because if all the inputs are false, then the output is always false too.

$S = \{\wedge, \oplus\} = \{ab, a \oplus b\}$

Not linear

linear

$\{0 \cdot 0 = 0$
$0 \oplus 0 = 0\}$

linearity Guideline cannot Apply.

$(\rightarrow, \oplus) \Longrightarrow (\text{NOT}, \rightarrow, \oplus) \Rightarrow \text{FC}$

FC

$(\rightarrow, \text{NOT})$ itself is FC.

$a+b$: $a' \rightarrow b$

However $\{\rightarrow, \oplus\}$ is complete, because $\{\neg, \rightarrow\}$ is known to be, and $(A \rightarrow A) \oplus A$ is equivalent to $\neg A$.

$$\boxed{(A \rightarrow A) \oplus A = \overline{A}}$$

Not op

$A \rightarrow A = 1$; $\boxed{1 \oplus A = \overline{A}}$

$S = \{\rightarrow, \oplus\} = \{a \rightarrow b, a \oplus b\}$

① $0 \rightarrow 0 \neq 0$ — G2 Not Applicable.

② $1 \oplus 1 \neq 1$ — G3 Not " .

③ $a \rightarrow b = f \Rightarrow$ Not self Dual

$a \oplus b = g \Rightarrow$ " " "

} G4 Not Applicable.

(4) linear fun $\Rightarrow$ $a \oplus b$ —

Non-linear $\Rightarrow$ $a \rightarrow b$ —

linearity Guideline Not Applicable.