QC9

Wednesday, March 3, 2021 4:29 PM

$2 + 4 = 6$ $f(x, y)$

$f(2, 4) = 6$

Inputs

⇒ Functional Completeness

- We can think $\wedge$ $\neg$ as function
  - $\wedge$ — function with two inputs (P, Q) $P \wedge Q$
  - $\neg$ — function with one input $\neg P$
- We can invent own function that takes
  - Number of inputs, $P, Q, R, \ldots$
  - with each having values of T, F

Boolean Function

- Consider $f(P, Q, R)$

	P	Q	R	f(P,Q,R)	
x_1	T	T	T	F	
x_2	T	T	F	F	
x_3	T	F	T	T	1
x_4	T	F	F	F	
x_5	F	T	T	F	
x_6	F	T	F	T	2
x_7	F	F	T	F	
x_8	F	F	F	T	3

P Q R

$f(x_1, x_2, \ldots, x_8)$

2^8 — possible functions

$\hat{x}$ $\hat{y}$ $\hat{z}$

- No matter how we choose these functions
  we can find equivalent expression
  that uses only function $\neg$ and $\wedge$ — functions
- Consider values T in the last column
  - 1st T occurs $P = T$, $Q = F$, $R = T$
    $\boxed{P \wedge \neg Q \wedge R} \Rightarrow \boxed{T}$
  - next T $P = F$ $Q = T$ $R = F$
    $\neg P \wedge Q \wedge \neg R$ T
  - Last $P = F$ $Q = F$ $R = F$
    $\neg P \wedge \neg Q \wedge \neg R \longrightarrow T$
- combining $\underbrace{\underbrace{(P \wedge \neg Q \wedge R)}_{C} \vee \underbrace{(\neg P \wedge Q \wedge \neg R)}_{D}} \vee (\neg P \wedge \neg Q \wedge \neg R) = f(P, Q, R)$
- final step is to replace $\vee$ $\boxed{C \vee D = \neg(\neg C \wedge \neg D)}$

$$\underbrace{\left[\neg\left(\neg(P \wedge \neg Q \wedge R) \wedge \neg(\neg P \wedge Q \wedge \neg R)\right)\right]}_{C} \vee \underbrace{(\neg P \wedge \neg Q \wedge \neg R)}_{D}$$

$$\neg\left(\neg\left[\neg\left(\neg(P \wedge \neg Q \wedge R) \wedge \neg(\neg P \wedge Q \wedge \neg R)\right)\right] \wedge \neg\left[\neg P \wedge \neg Q \wedge \neg R\right]\right) = f(P, Q, R)$$

$\{\neg, \wedge\} \rightarrow$ functionally complete. $\hat{\iota}, \hat{\jmath}, \hat{k}$

Nand not and and $\uparrow$ |

$P \uparrow Q = \neg(P \wedge Q)$

and.

P	Q	$P\uparrow Q$
T	T	F
T	F	T
F	T	T
F	F	T

P	Q	$P\wedge Q$
T	T	T
T	F	F
F	T	F
F	F	F

$\Rightarrow$ Nand is functionally complete

Just need to show that $\neg \rightarrow$ Nand, $\wedge \rightarrow$ Nand

Nand($P\uparrow P$) — same truth value

P	$P\wedge P$	$\neg(P\wedge P)$	$\neg P$
T	T	F	F
F	F	T	T

$P\uparrow P = \neg(P\wedge P) \equiv \neg P$

$\boxed{P\uparrow P = \neg P}$

$\neg \underbrace{(P\wedge Q)}_{C} = \neg C = C\uparrow C =$

$= \underbrace{(P\wedge Q)}_{\text{and}} \uparrow \underbrace{(P\wedge Q)}_{\text{and}} \equiv \underbrace{\neg(P\wedge Q)}_{\text{NANT}}$

not $\neg \longrightarrow$ Nand $\uparrow$.

$P\wedge Q = \neg\neg(P\wedge Q)$ $\quad \neg(P\wedge Q) = P\uparrow Q$ Nand

$P\wedge Q = \neg\underbrace{(P\uparrow Q)}_{D} \longrightarrow \boxed{\neg D} = \boxed{D\uparrow D}$.

$\rightarrow \neg D = D\uparrow D = \underbrace{(P\uparrow Q)}_{D}\uparrow\underbrace{(P\uparrow Q)}_{D}$

$\boxed{P\wedge Q = (P\uparrow Q)\uparrow(P\uparrow Q)}$

$C\wedge D = ((C\uparrow D)\uparrow(C\uparrow D))$

Nand $\rightarrow$ Functionally

$\neg A = A \uparrow A$

Nand

Complete

⇒ Henry Sheffer first introduced Nand in 1913

He used symbol | (instead of ↑) – Sheffer stroke

$P \uparrow Q$

$P \mid Q$

⇒ Charles Sanders Peirce also introduced the same logical function by late 19th century

F → 0

T → 1

operating Bits

⇒ Two Tables for $P \vee Q$

P	Q	$P \vee Q$
T	T	T
T	F	T
F	T	T
F	F	F

P	Q	$P \vee Q$
0	0	0
0	1	1
1	0	1
1	1	1