

⇒ Functional Completeness

- We can think \wedge \neg as functions

\wedge - function with two inputs (P,Q) $P \wedge Q$

\neg - function with one input $\neg P$

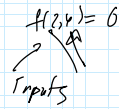
- We can invent our own function that takes

- Number of inputs, P, Q, R, ...

- with each having values of T, F

→ Boolean Function

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



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

P	Q	R	f(P,Q,R)
T	T	T	F
T	T	F	F
T	F	T	T
T	F	F	F
F	T	T	F
F	T	F	F
F	F	T	F
F	F	F	F

2^3 possible functions

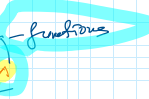
$f(x_1, x_2, \dots, x_n)$

$\begin{pmatrix} \neg \\ \wedge \\ \vee \end{pmatrix}$

- No matter how we choose these functions

We can find equivalent expression

that uses only function \neg and \wedge



- Consider values T in the last column

- 1st T occurs $P \equiv T, Q \equiv F, R \equiv T$

$P \wedge \neg Q \wedge R \Rightarrow T$

- next T $P \equiv F, Q \equiv T, R \equiv F$

$\neg P \wedge Q \wedge \neg R \Rightarrow T$

- last $P \equiv F, Q \equiv F, R \equiv F$

$\neg P \wedge \neg Q \wedge \neg R \Rightarrow T$

- combining $(P \wedge \neg Q \wedge R) \vee (\neg P \wedge Q \wedge \neg R) \vee (\neg P \wedge \neg Q \wedge \neg R) = f(P,Q,R)$

- final step is to replace \vee $C \vee D = \neg(\neg C \wedge \neg D)$

$\neg(\neg(\neg(P \wedge \neg Q \wedge R) \wedge \neg(\neg P \wedge Q \wedge \neg R)) \vee \neg(\neg P \wedge \neg Q \wedge \neg R))$

$\neg(\neg(\neg(P \wedge \neg Q \wedge R) \wedge \neg(\neg P \wedge Q \wedge \neg R)) \wedge \neg(\neg P \wedge \neg Q \wedge \neg R)) = f(P,Q,R)$

$\{\neg, \wedge\} \rightarrow$ functionally complete. \neg, \wedge, \uparrow

Nand not and and \uparrow |

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

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 and $\wedge \rightarrow$ Nand.

same truth value

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

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

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

$\neg \rightarrow$ Nand \uparrow

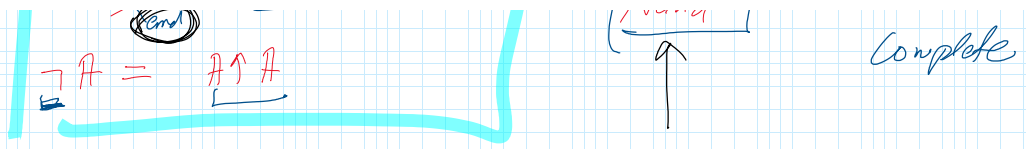
$P \wedge Q = \neg(\neg(P \wedge Q))$

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

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

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

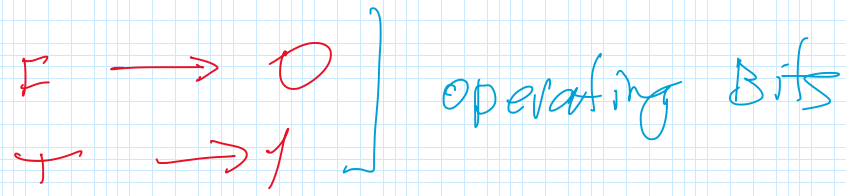
$C \wedge D = \neg(\neg C \uparrow \neg D)$ Nand \rightarrow Functionally



⇒ Henry Sheffer first introduced Nand in 1913
 He used symbol \uparrow (instead of \neg) - sheffer stroke

⇒ Charles Sanders Peirce also introduced the same
 Logical Function by late 19th century

$\neg \neg Q$
 $\neg \neg \neg Q$
 \vdots



⇒ Two Tables for PVQ

P	Q	PVQ
T	T	T
T	F	F
F	T	F
F	F	T

P	Q	PVQ
0	0	0
0	1	1
1	0	1
1	1	1

