Thus far …

1. Propositions, truth tables, laws of propositional logic, rules of inference
2. Checking validity of logical arguments
3. Quantified predicates

Today

Given a truth table, construct a corresponding proposition.

There are (infinitely) many equivalent answers.

Example

x	y	z	P
T	T	T	T
T	T	F	T
T	F	T	F
T	F	F	F
F	T	T	T
F	T	F	T
F	F	T	F
F	F	F	T

Example

	x	y	z	P
Row 1	T	T	T	T
Row 2	T	T	F	T
	T	F	T	F
	T	F	F	F
Row 3	F	T	T	T
Row 4	F	T	F	T
	F	F	T	F
Row 5	F	F	F	T

Row 1: $x \wedge y \wedge z$

Row 2: $x \wedge y \wedge \neg z$

Row 3: $\neg x \wedge y \wedge z$

Row 4: $\neg x \wedge y \wedge \neg z$

Row 5: $\neg x \wedge \neg y \wedge \neg z$

$$P \equiv (x \wedge y \wedge z) \vee (x \wedge y \wedge \neg z) \vee (\neg x \wedge y \wedge z) \vee (\neg x \wedge y \wedge \neg z) \vee (\neg x \wedge \neg y \wedge \neg z)$$

Disjunctive Normal Form

For every truth table with variables $x_1, x_2, \ldots, x_n$ there is a corresponding equivalent proposition of the form:

$$\bigvee_{1}^{m} \bigwedge_{1}^{n} l_i$$

Where $l_i = x_j \ or \ \neg x_j$.

Every proposition can be expressed using only the connectives ¬, ∧, ∨

¬, ∧, ∨ is a *functionally complete* set of connectives

Functional Completeness

$x \vee y \equiv \neg\neg(x \vee y) \equiv \neg(\neg x \wedge \neg y)$

We can replace every occurrence of $\vee$ with $\neg$, $\wedge$.

So every proposition can be expressed using only $\neg$, $\wedge$.

Similarly, since $x \wedge y \equiv \neg\neg(x \wedge y) \equiv \neg(\neg x \vee \neg y)$

we can replace every occurrence of $\wedge$ with $\neg$, $\vee$.

So every proposition can be expressed using only $\neg$, $\vee$.

Is there one connective that is functionally complete?

Common binary connectives

x	y	$x \vee y$	$x \wedge y$	$x \Rightarrow y$	$XOR(x,y)$	$NOR(x,y)$	$NAND(x,y)$
T	T	T	T	T	F	F	F
T	F	T	F	F	T	F	T
F	T	T	F	T	T	F	T
F	F	F	F	T	F	T	T

There are 16 binary connectives. This table includes 6 of them.

$$XOR(x,y) = x \oplus y \equiv (\neg x \wedge y) \vee (x \wedge \neg y)$$
$$NOR(x,y) = x \downarrow y \equiv \neg (x \vee y)$$
$$NAND(x,y) = x \uparrow y \equiv \neg (x \wedge y)$$

NAND is functionally complete!

$$NAND(x,y) = x \uparrow y \equiv \neg(x \wedge y)$$

If we can emulate $\neg x$ and $x \vee y$ using only $\uparrow$ then $\uparrow$ is functionally complete.

$\neg x \equiv x \uparrow x$

$x \vee y \equiv \neg\neg(x \vee y)$
$\equiv \neg(x \uparrow y)$
$\equiv (x \uparrow y) \uparrow (x \uparrow y)$

NOR is functionally complete!

$$NOR(x,y) = x \downarrow y \ \equiv \neg(x \vee y)$$

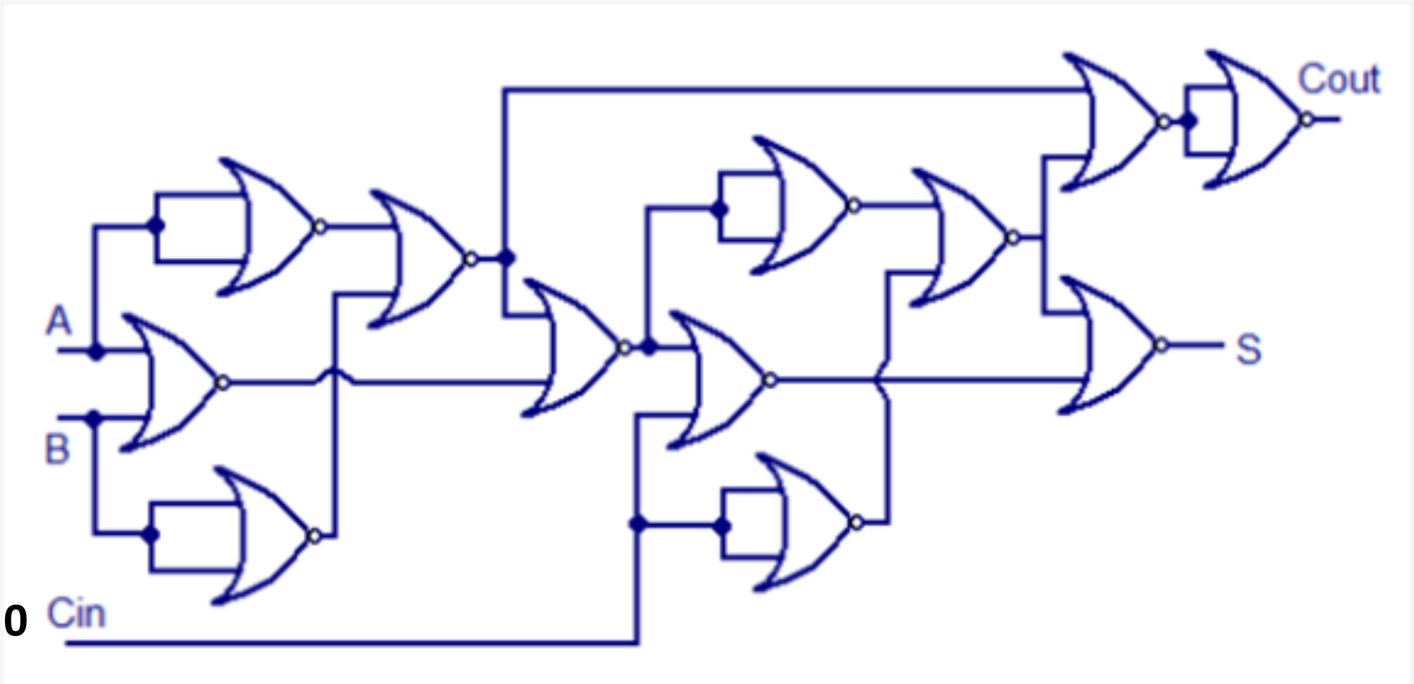
If we can emulate $\neg x$ and $x \wedge y$ using only $\downarrow$ then $\downarrow$ is functionally complete.

$$\neg x \equiv x \downarrow x$$

$$x \wedge y \equiv \neg\neg(x \wedge y)$$
$$\equiv \neg(\neg x \vee \neg y)$$
$$\equiv \neg((x \downarrow x) \vee (y \downarrow y))$$
$$\equiv (x \downarrow x) \downarrow (y \downarrow y)$$

1-bit adder using NOR gates

$A{\downarrow}0$
$B{\downarrow}0$
1 $S{\downarrow}0$



2-bit adder using NOR gates

↓1 A↓0
↓1 B↓0
2 S↓1 S↓0