~P	=	Not-P
$(\mathbf{P} \cdot \mathbf{Q})$	=	Both P and Q
$(\mathbf{P} \lor \mathbf{Q})$	=	Either P or Q
$(\mathbf{P} \supset \mathbf{Q})$	=	If P then Q
$(P \equiv Q)$	=	P if and only if Q

Propositional logic

- 1. Any capital letter is a wff.
- 2. The result of prefixing any wff with "~" is a wff.
- 3. The result of joining any two wffs by "•" or "∨" or "⊃" or "≡" and enclosing the result in parentheses is a wff.

Two translation rules

Put "(" wherever you see "both," "either," or "if."

Either not A or B = $(\sim A \lor B)$ Not either A or B = $\sim (A \lor B)$

Group together parts on either side of a comma.

If A, then B and C = $(A \supset (B \cdot C))$ If A then B, and C = $((A \supset B) \cdot C)$

LogiCola C (EM & ET)

Pages 113–14

$$\sim P = \text{Not-P}$$

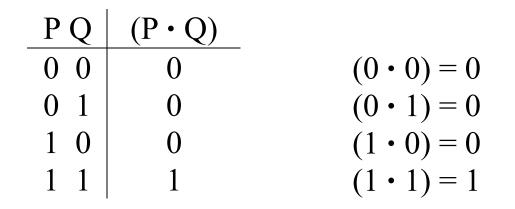
(P \cdot Q) = Both P and Q
(P \cdot Q) = Either P or Q
(P \cdot Q) = If P then Q
(P \cdot Q) = P if and only if Q

Propositional translations

Put "(" wherever you see "both," "either," or "if." Group together parts on either side of a comma.

Pages 112–14

"I went to Paris and I went to Quebec."

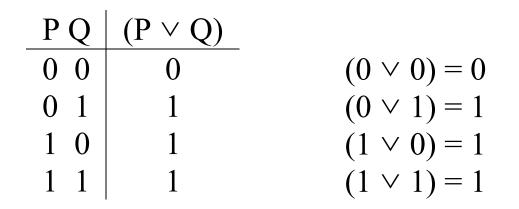


" $(P \cdot Q)$ " claims that *both* parts are true. " $(P \cdot Q)$ " is a *conjunction*; P and Q are its *conjuncts*.

LogiCola D (TE & FE)

Page 114–18

"I went to Paris or I went to Quebec."



"($P \lor Q$)" claims that *at least one* part is true. "($P \lor Q$)" is a *disjunction*; P and Q are its *disjuncts*.

LogiCola D (TE & FE)

Page 114–18

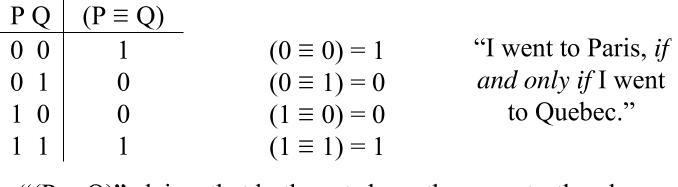
"If I went to Paris, then I went to Quebec."

ΡQ	$(\mathbf{P} \supset \mathbf{Q})$	
0 0	1	$(0 \supset 0) = 1$
0 1	1	$(0 \supset 1) = 1$
1 0	0	$(1 \supset 0) = 0$
1 1	1	$(1 \supset 1) = 1$

"($P \supseteq Q$)" says we *don't* have the first part true and the second false. "($P \supseteq Q$)" is a *conditional*; P is the *antecedent* and Q the *consequent*.

Falsity implies anything.	$(0 \supset) = 1$
Anything implies truth.	$(\supset 1) = 1$
Truth doesn't imply falsity.	$(1\supset 0)=0$

Page 114–18



"($P \equiv Q$)" claims that both parts have the *same* truth value. "($P \equiv Q$)" is a *biconditional*.

P
$$\sim P$$
"I didn't go01 $\sim 0 = 1$ "I didn't go10 $\sim 1 = 0$ to Paris."

"~P" has the *opposite* value of "P." "~P" is a *negation*.

Pages 114-18

Basic Truth Equivalences

AND	OR	IF-THEN	IFF	NOT
$(0 \cdot 0) = 0$ $(0 \cdot 1) = 0$ $(1 \cdot 0) = 0$ $(1 \cdot 1) = 1$	$(0 \lor 0) = 0$ $(0 \lor 1) = 1$ $(1 \lor 0) = 1$ $(1 \lor 1) = 1$	$(0 \supset 0) = 1$ $(0 \supset 1) = 1$ $(1 \supset 0) = 0$ $(1 \supset 1) = 1$	$(0 \equiv 0) = 1$ $(0 \equiv 1) = 0$ $(1 \equiv 0) = 0$ $(1 \equiv 1) = 1$	$\begin{array}{l} \sim 0 = 1 \\ \sim 1 = 0 \end{array}$
both parts	at least one	we don't have <i>first</i>	both parts have	reverse the

are true

part is true

true & second false *same* truth value

truth value

Falsity implies anything. Anything implies truth. Truth doesn't imply falsity.

Truth Evaluations

Assume that A=1 and X=0.

What is the truth value of " \sim (A · X)"?

- Plug in "1" and "0" for the $\sim (A \cdot X)$ letters. $\sim (1 \cdot 0)$
- Simplify from the inside out, until you get "1" or "0."

Visualize "~
$$(1 \cdot 0)$$
" as "~ $(1 \cdot 0)$."

Pages 118-19

~0

Unknown Evaluations

Assume that T=1 and F=0 and U=? (unknown). What is the truth value of "(U • F)"?

"($? \cdot 0$)" has to be 0 because:

An AND with one part 0 is 0. It comes out 0 both ways: $(1 \cdot 0) = 0$ and $(0 \cdot 0) = 0$.

LogiCola D (UE, UM, & UH)

Pages 119–20

Complex Truth Tables: do one for " $(P \equiv \sim Q)$ "

PQ	$(P \equiv \sim Q)$	
0 0	0	$(0 \equiv \sim 0) = (0 \equiv 1) = 0$
0 1	1	$(0 \equiv \sim 1) = (0 \equiv 0) = 1$
1 0	1	$(1 \equiv \sim 0) = (1 \equiv 1) = 1$
1 1	0	$(1 \equiv \sim 1) = (1 \equiv 0) = 0$

- Write the formula: " $(P \equiv \sim Q)$ "
- On the left, write each letter in the formula: "P" and "Q"
- Below this, write each truth combination; n letters give 2ⁿ truth combinations.
- Figure out the value of the formula on each combination.

VALID = No possible case has premises all true and conclusion false.



1INVALID = Some possi-1ble case has premises all $\therefore 0$ true and conclusion false.

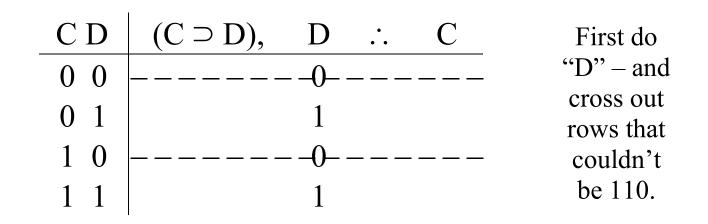
Truth-table test: Construct a truth table showing the truth value of the premises and conclusion for all possible cases. The argument is **VALID** if and only if no possible case has premises all true and conclusion false.

C D	$(C \supseteq D),$	D	•	С		
0 0	1	0		0		
0 1	1	1		0	\leftarrow	Invalid
1 0	0	0		1		
1 1	1	1		1		

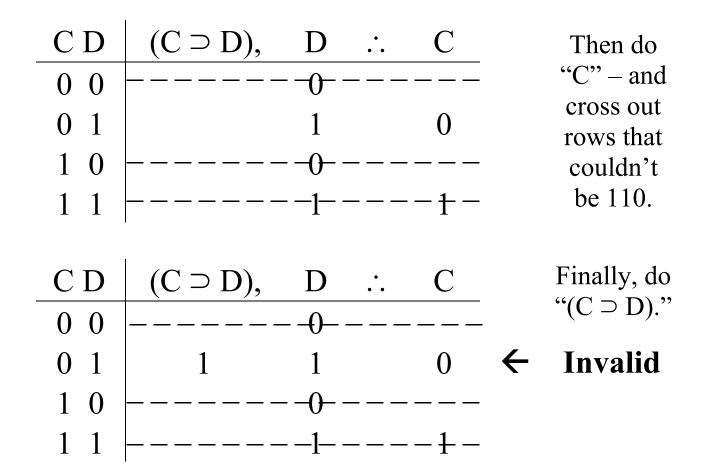
LogiCola D (AE & AM)

Pages 122-24

On the short-cut truth-table test, evaluate easier wffs first and cross out rows that couldn't have true premises and a false conclusion.



Pages 122-24



LogiCola D (AE & AM)

Pages 122–24

VALID = No possible case has premises all true and conclusion false.



1INVALID = Some possi-1ble case has premises all $\therefore 0$ true and conclusion false.

Truth-assignment test: Set each premise to 1 and the conclusion to 0. Figure out the truth value of as many letters as possible. The argument is **VALID** if and only if no possible way to assign 1 and 0 to the letters will keep the premises all 1 and conclusion 0.

$$(L \lor R) = 1 \qquad (L^0 \lor R^0) = 1 \qquad (L^0 \lor R^0) \neq 1$$

$$\sim L = 1 \qquad \Rightarrow \qquad \sim L^0 = 1 \qquad \Rightarrow \qquad \sim L^0 = 1$$

$$\cdot R^0 = 0 \qquad \therefore R^0 = 0$$

LogiCola E (S & E)

Pages 126–28

Harder Translations

Translate "but" ("yet," "however," "although," and so on) as "and."

> Translate "unless" as "or."

Translate "just if" and "iff" (a logician word) as "if and only if." Northwestern played, but it won = (N · W)

You'll die *unless* you give me your money $= (D \lor M)$

I'll take the job *just if* you pay me a million = $(J \equiv M)$

LogiCola C (HM & HT)

Pages 132-33

The part after "if" ("provided that," "assuming that," and so on) is the if-part (the antecedent, the part before the horseshoe).

Gensler is happy if Michigan wins.

= *If* Michigan wins, Gensler is happy.

= (M \supset H)

The part after "only if" is the then-part (the consequent, the part after the horseshoe). (Or write " \supset " for "only if.")

You're alive *only if* you have oxygen. = *Only if* you have oxygen are you alive. = $(A \supset O) = (\sim O \supset \sim A)$

LogiCola C (HM & HT)

Pages 132-33

Oxygen is *sufficient for* life = $(O \supset L)$ Oxygen is *necessary for* life = $(\sim O \supset \sim L)$ Oxygen is *necessary and sufficient for* life = $(O \equiv L)$

LogiCola C (HM & HT)

Harder Translations

- BUT = YET = HOWEVER = ALTHOUGH = AND.
- UNLESS = OR.
- JUST IF = IFF = IF AND ONLY IF.
- The part after "if" ("provided that," "assuming that," and so on) goes *before* the horseshoe.
- The part after "only if" goes *after* the horseshoe.
- Oxygen is *sufficient for* life = $(O \supset L)$
- Oxygen is *necessary for* life = $(\sim O \supset \sim L)$
- Oxygen is *necessary and sufficient for* life = $(O \equiv L)$

LogiCola C (HM & HT)

Pages 132–33

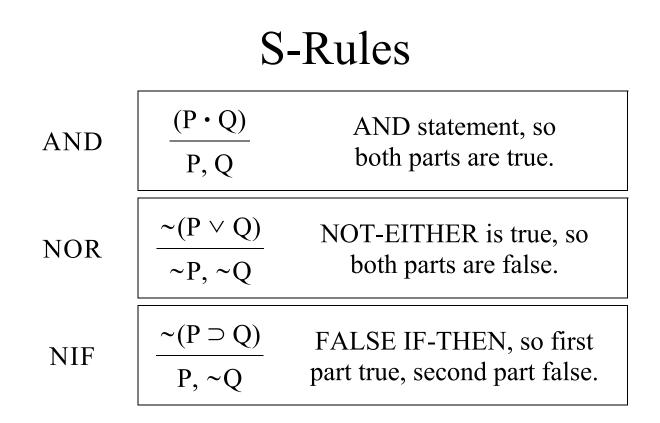
Idiomatic arguments

1. Identify the conclusion.

These often indicate premises: Because, for, since, after all ... I assume that, as we know ... For these reasons ... These often indicate conclusions:

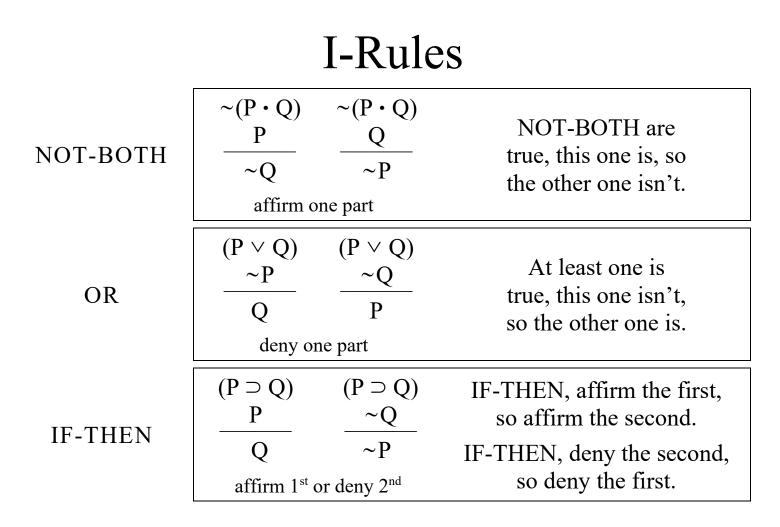
Hence, thus, so, therefore ...

- It must be, it can't be ... This proves (or shows) that ...
- 2. Translate into logic, using wffs. Add implicit premises if needed.
- 3. Test for validity.



We can simplify AND, NOR, and NIF.

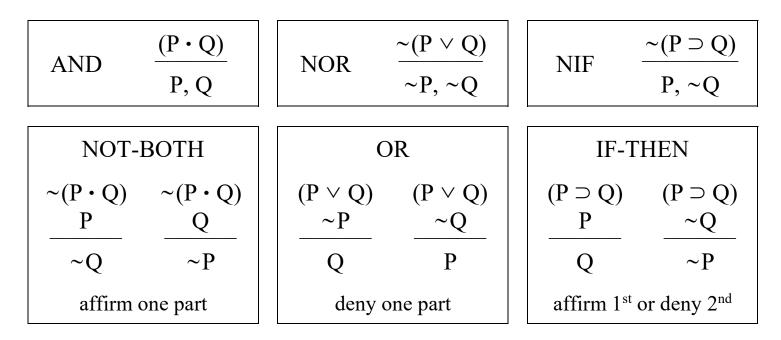
Pages 136–39



LogiCola F (IE & IH)

Pages 139-42

S- and I-Rules



Extended S- and I-rule inferences

 $\frac{\sim ((\mathbf{A} \cdot \mathbf{B}) \supset \sim \mathbf{C})}{(\mathbf{A} \cdot \mathbf{B}), \mathbf{C}}$

FALSE IF-THEN, so first part true, second part false.



 $\frac{((\mathbf{A} \cdot \mathbf{B}) \supset \sim \mathbf{C})}{\sim (\mathbf{A} \cdot \mathbf{B})}$ nil

IF-THEN: need first part true or second part false $\frac{(\mathbf{(A \cdot B)} \supset \mathbf{\sim C})}{\mathbf{\sim (A \cdot B)}}$ nil

Logic gates and computers

$$\begin{array}{c|c} A & \overrightarrow{} \\ B & \overrightarrow{} \end{array} \quad AND-GATE \quad \overrightarrow{} \quad (A \cdot B) \end{array}$$

AB	$(\mathbf{A} \boldsymbol{\cdot} \mathbf{B})$
0 0	0
0 1	0
1 0	0
1 1	1

An AND-GATE gives an output voltage if and only if both inputs have an input voltage.