Translating into propositional logic o

# **Propositional Logic**

#### **Robert Y. Lewis**

CS 0220 2024

January 29, 2024

Translating into propositional logic

#### Overview

- 1 Propositions from Propositions (3.1)
  - NOT, AND, and OR (3.1.1)
  - IMPLIES (3.1.2)
  - If and Only If (3.1.3)
- 2 Translating into propositional logic

Translating into propositional logic o

### What is propositional logic?

Today, we care about the *shape* of propositions, not their content. We have:

- *Atoms*: propositions that we can't decompose any further
- Connectives: operators that take in propositions and produce new propositions

We build compound propositions out of smaller ones. "It is snowing, and I am cold."

"Boolean logic": every proposition is either true or false. The truth of a compound proposition depends on the truth of its components.

NOT, AND, and OR (3.1.1)

# Notation for propositional logic

Our connectives:

- **NOT:** ¬*P*
- **AND:**  $P \land Q$
- **OR:** *P* ∨ *Q*
- IMPLIES (IF-THEN):  $P \rightarrow Q$
- $\blacksquare \ \textbf{IFF:} P \leftrightarrow Q$

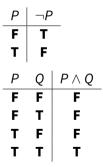
$$egin{aligned} & P \wedge 
eg Q \ & P \wedge (
eg Q) \ & (A \wedge B) o (C \lor D) \end{aligned}$$

Translating into propositional logic O

Translating into propositional logic O

### Truth tables

Truth tables help convey how to interpret logical statements.



Like a multiplication table, but flattened (so easier to apply to more inputs.)

NOT, AND, and OR (3.1.1)

#### OR

Ρ	Q	$P \lor Q$
F	F	F
F	Т	Т
т	F	Т
т	т	т

In English, often rendered as "*P* or *Q* or both". Inclusive or.

Ρ	Q	P XOR Q
F	F	F
F	Т	Т
т	F	Т
Т	Т	F

Exclusive or, too.

Translating into propositional logic  $_{\rm O}$ 

# Implication

Perhaps counterintuitive.

 $\begin{array}{c|c|c} X & Y & X \rightarrow Y \\ \hline F & F & T \\ F & T & T \\ T & F & F \\ T & T & T \end{array}$ 

This is known as the *material conditional*. Compare:

If it is raining outside, then I have my umbrella. (material)

If it were raining outside, then I would have my umbrella. (counterfactual)

Math deals with material conditionals, so this class does too.

Translating into propositional logic o

Implication

 $\begin{array}{c|ccc} X & Y & X \rightarrow Y \\ \hline F & F & T \\ F & T & T \\ T & F & F \\ T & T & T \end{array}$ 

**Claim**: If *p* is a prime number greater than 2, then *p* is odd.

"Should" be true for every number *p*.

Definition: In " $X \rightarrow Y$ ", X is called the *hypothesis* and Y the *conclusion*.

Can you think of an equivalent way to write this using  $\wedge,\vee,\neg$  ?

Translating into propositional logic o

Implication

Χ	Y	$X \to Y$
F	F	Т
F	т	Т
Т	F	F
Т	Т	Т

X	Ŷ	$\neg X$	$\neg X \lor Y$
F	F	Т	Т
F	т	Т	Т
Т	F	F	F
т	Т	F	т

Translating into propositional logic

Truth table for iff

 $\begin{array}{c|cc} X & Y & X \leftrightarrow Y \\ \hline F & F & T \\ F & T & F \\ T & F & F \\ T & T & T \end{array}$ 

Example: A number is divisible by three *if and only if* the sum of its digits is divisible by three.

Definitions have this form.

Translating into propositional logic

Translating into propositional logic O

#### Truth tables for compound formulas

To build a truth table for a compound formula:

- Find the atoms, and make those columns.
- Find the compound subformulals, and make those columns.
- Fill in all combinations of truth values for the atoms.
- Propogate right, column by column.

An example in Lean

## Natural language translations

Translating English sentences into propositional logic seems straightforward: choose your atomic propositions. Then "and" becomes  $\land$ , "or" becomes  $\lor$ , ...

But it's not always quite so easy. Define:

- *T* := "Rob likes t rexes"
- S := "Rob likes stegosauruses"
- *F* := "Rob's dinosaur zoo is full"
- *W* := "Rob wants more dinosaurs"

"Rob likes t rexes and stegosauruses."  $T \wedge S$ 

"Rob wants more dinosaurs, but his zoo is full."  $W \wedge F$ 

"If Rob does not want more plants, then he doesn't like t rexes or stegosauruses."  $\neg W \rightarrow \neg (T \lor S)$  or maybe  $\neg W \rightarrow (\neg T \land \neg S)$