

Propositional Logic: Fundamental Elements for Computer Scientists

0. Motivation for Computer Scientists

1. Propositions and Propositional Variables

2. Operators \neg , \wedge , \vee , \Rightarrow

3. Truth Tables

4. Compound Propositions and Functions

5. Completeness of a Set of Operators

6. Tautologies and their significance in Proofs

0. Motivation for Computer Scientists

Practice: to define, verify, check, etc real systems.

Theory: Foundation of proof techniques,
(and all other rational arguments).

1A. Propositions

Definition: Propositions are unambiguous declarative statements which are either True (T) or False (F).

Examples:

“Washington, DC is the capital of the U.S.” is a true proposition.

$5 > 3$ is a true proposition.

$3 > 5$ is a false proposition.

$x > 3$ is not a proposition.

Its truth value depends on x .

“What time is it?” is not a proposition. It is not declarative.

“Athens is in Georgia” is not a proposition.

It is ambiguous. Athens, GA or Athens, Greece ?

1A. Propositions (continued)

Propositions can be more complex and interesting.

Here are some **further examples:**

“For every integer n , there exists an integer m , such that m is strictly greater than n ” is also a true proposition.

It is more complex because it involves statements like “for every” and “there exists”.

But otherwise it says something pretty obvious.

1A. Propositions (continued)

And **yet another example:**

“For every prime number p , there exists a prime number q , such that q is strictly greater than p ” is a true proposition.

This is a very interesting proposition, because it is actually true. It's truth implies that there are infinitely many prime numbers. This is a primordial mathematical fact.

(“Primes are to numbers what notes are to music.”)

It's truth is also of major practical significance in cryptography, where increasingly larger prime numbers are needed to encrypt and decrypt messages.

1B. Propositional Variables

Definition: Propositional variables are variables whose value is either T (true) or F (false).

Notation:

Just as we mostly use k, l, m, n to denote integer variables, or x, y, z to denote real variables, we mostly use p, q , (and to a lesser extent r, s, t) to denote propositional variables.

2. Operators \neg , \wedge , \vee , \Rightarrow

Motivation: Eventually, beyond single one statement propositions, we want to express somewhat more interesting statements.

Operators are the building blocks allowing us to express complex propositions, when we start from very simple propositions.

Examples:

$$p \vee q$$

$$\neg(p \vee q) \Rightarrow (\neg p \wedge \neg q)$$

$$[(p_1 \Rightarrow p_2) \wedge (p_2 \Rightarrow p_3)] \Rightarrow (p_1 \Rightarrow p_3)$$

2. Operators \neg , \wedge , \vee , \Rightarrow (continued)

*Ok, ok, this is too formal right now...
just move on to the next slide
and it will be completely clear.
Then come back to this slide
and see what this definition is talking about.*

Definition: Operators are functions from a finite set of propositional variables to the set $\{\top, \text{F}\}$

3. Truth Tables

\neg , \wedge , \vee , \Rightarrow

p	$\neg p$
T	F

Truth Table
for NEGATION

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Truth Table
for logical AND

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Truth Table
for logical OR

3. Truth Tables \neg , \wedge , \vee , \Rightarrow (continued)

p	q	$p \Rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

Truth Table
for logical
IMPLICATION

Note: Do not confuse this logical implication with “if” ... “then” ... commands in programming languages.

4. Compound Propositions and Functions

*We usually
denote this as
 $p \Leftrightarrow q$*

Example: $(p \Rightarrow q) \wedge (q \Leftarrow p)$

p	q	$p \Rightarrow q$	$q \Rightarrow p$	$(p \Rightarrow q) \wedge (q \Leftarrow p)$
T	T	T	T	T
T	F	F	T	F
F	T	T	F	F
F	F	T	T	T

5. Completeness of a Set of Operators

The set of operators \neg , \wedge , \vee are very important because they form a complete set: any compound proposition or function can be expressed in terms of \neg , \wedge , \vee

For example, $p \Rightarrow q$

is exactly the same as $(p \wedge q) \vee (\neg p \wedge q) \vee (\neg p \wedge \neg q)$

For further example, $p \Leftrightarrow q$

is exactly the same as $(p \wedge q) \vee (\neg p \wedge \neg q)$

In general, we can use the \neg , \wedge , \vee operators to express the combinations of truth values of the variables that make the general function true.

6. Tautologies and their significance in Proofs

Definition: A tautology is a function which is always true, ie, it is true for all combinations of truth values of the variables.

Tautologies and Proofs: There is small set of tautologies that expresses many fundamental proof methods.

6. Tautologies and their significance in Proofs (continued)

Here are some basic tautologies.

De Morgan's Laws:

$$\neg(p \wedge q) \Leftrightarrow [\neg p \vee \neg q]$$

$$\neg(p \vee q) \Leftrightarrow [\neg p \wedge \neg q]$$

6. Tautologies and their significance in Proofs (continued)

Modus Ponens, aka Direct Proof

it means reduction to a more general principal:

$$[(p \Rightarrow q) \wedge p] \Rightarrow q$$

6. Tautologies and their significance in Proofs (continued)

Modus Tollens, aka Proof by Contradiction:

$$[(p \Rightarrow q) \wedge \neg q] \Rightarrow \neg p$$

6. Tautologies and their significance in Proofs (continued)

Contraposition:

$$(p \Rightarrow q) \Leftrightarrow (\neg q \Rightarrow \neg p)$$

6. Tautologies and their significance in Proofs (continued)

Hypothetical Syllogism:

$$[(p_1 \Rightarrow p_2) \wedge (p_2 \Rightarrow p_3)] \Rightarrow (p_1 \Rightarrow p_3)$$