Victor Amelkin

Propositional Logic

Rules of inference begin on Page 7

Mathematical logic is all about formally operating on statements that can be true or false. Multiple logics – propositional, first-, and second-order – differ in their expressiveness; propositional logic is the least expressive, but it is still good for many applications.

def A proposition is a declarative statement that can be either true or false.

def Propositional variables represent propositions.

Exercise 1: Which of the following are propositions?

1) "I have a pen." ✓
2) "'To be or not to be?' – is a question." ✓
3) "A proposition is either true or false." ✓
4) "True." ×
5) "True is not False." ✓
6) "True is not True." ✓

7*) "This statement is false." ×

$P \equiv$ "This statement is false" $\equiv$

$\equiv$ "P is false".

(i) Suppose P is true; hence,
$P \equiv$ "P is false" is true
$\Rightarrow$ P is false – a contradiction.

(ii) Suppose P is false; hence,
$P \equiv$ "P is false" is false
$\Rightarrow$ P is true – a contradiction.

$\Rightarrow$ P cannot be true or false.
$\Rightarrow$ P is not a proposition.

def A compound proposition is a proposition built from simpler propositions using logical connectives, such as $\neg, \vee, \wedge$. The simplest proposition we can have is just a propositional variable – such propositions are "indivisible" and, hence, are referred to as atomic or simple. (Correction: an even simpler proposition is just "True" or "False".)

Most popular logical connectives:

- $\neg x$ – negation ("not")
- $x \vee y$ – disjunction ("or")
- $x \wedge y$ – conjunction ("and")
- $x \rightarrow y$ – implication ("if-then")
- $x \leftrightarrow y$ – bidirectional implication ("iff" ≡ "if and only if")
- $x \oplus y$ – exclusive or ("xor")

(Sometimes, xor is denoted with $\veebar$. However, $\oplus$ is rather widespread.)

Examples:

1) "I have a pen" - simple/atomic proposition

2) "It is raining" - another atomic proposition

3) P - it is a proposition assuming that P is a propositional variable.

(If we agree that propositional variables can only represent atomic propositions (e.g., $P \equiv$ "I have a pen"), then P is atomic. If we agree that P can stand for any proposition (e.g., $P \equiv$ "2+2=4" $\wedge$ "3×3=9"), then P may be also seen as (potentially) compound.)

4) $\neg P \vee$ "I have a pen." - compound proposition.

5) $(P \oplus Q) \leftrightarrow (\neg P \wedge Q \vee P \wedge \neg Q)$ - another compound proposition.

(Sometimes propositions are called "formulas" or "logical formulas".)

<u>def</u> A <u>truth table</u> is a table describing values ("truth values") of a (usually, compound) proposition for every possible assignment of values to its (propositional) variables. (These assignments are called "truth assignments".)

<u>Examples:</u> (<u>Note</u>: The only possible <u>truth values</u> are <u>true</u> and <u>false</u>. Sometimes, we write F or 0 for false and T or 1 for truth - <u>just to save space</u>.)

1) $\neg x$:

x	$\neg x$
0	1
1	0

truth assignments

truth values for the proposition $\neg x$

Thus, $\neg$True is False.

2) $x \vee y$:

x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

Thus, (True $\vee$ False) is True.

truth assignments

truth values of $x \vee y$ under all possible truth assignments

3)

x	y	$x \wedge y$
0	0	0
0	1	0
1	0	0
1	1	1

4)

x	y	$x \rightarrow y$
0	0	1
0	1	1
1	0	0
1	1	1

5)

x	y	$x \leftrightarrow y$
0	0	1
0	1	0
1	0	0
1	1	1

6)

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

7)

x	y	z	$(x \wedge y) \rightarrow z$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Notice that if a proposition is large, it may be convenient to also describe its "sub-propositions" in the same truth table.

8)

x	y	z	$x \rightarrow y$ ("sub-propositions")	$y \wedge z$ ("sub-propositions")	$(x \rightarrow y) \oplus (y \wedge z)$ (target proposition)
0	0	0	1	0	1
0	0	1	1	0	1
0	1	0	1	0	1
0	1	1	1	1	0
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	1	0	1
1	1	1	1	1	0

def: Two propositions are *equivalent* if on the same truth assignments (or "inputs") they have the same truth values (or "outputs"). Equivalence is denoted with "$\equiv$": $P \equiv Q$ – P and Q are equivalent.

Examples:

> **Note:** In the lecture notes, equivalence of P and Q is denoted with $P \Leftrightarrow Q$.

1) $x \equiv \neg(\neg x)$ – not not x is clearly just x.
2) $x \vee y \equiv y \vee x$ – for disjunction, order does not matter ("commutativity").
3) $x \oplus y \equiv \neg x \wedge y \vee x \wedge \neg y$ – xor is just defined to behave this way.
4) $((x \wedge y) \vee y) \leftrightarrow \neg((x \oplus x) \vee \neg y) \equiv \text{True}$ – harder to tell why, but these are also equivalent.

Exercise 2: How many unique propositions in n propositional variables are there?

By "unique" we mean that, having counted a proposition, we are not going to also count the propositions equivalent to it. Hence, every proposition is not defined by how it looks, but it is defined by its "behavior", that is, by the values it has on different inputs (or, alternatively, under different truth assignments). In other words, each proposition is determined by its truth table. Thus, we just need to count the number of truth tables in n variables.

Since n is fixed, the left part of the truth table is defined. What is not defined are the truth values of the target proposition. Each unique combination of truth values of the proposition defines a unique truth table.

x_1	x_2	x_3	...	x_{n-1}	x_n	P
0	0	0	...	0	0	?
0	0	0	...	0	1	?
0	0	0	...	1	0	?
0	0	0	...	1	1	?
		...	...			⋮
1	1	1	...	1	0	?
1	1	1	...	1	1	?

2^n

Since each row corresponds to a unique binary string of length n, then there are 2^n rows.

Thus the number of truth tables in n variables equal the number of binary strings of length 2^n, which is 2^{2^n}.

Note: If we counted not "behaviorally unique", but "syntactically unique" propositions, then the resulting number would be ∞ instead of 2^{2^n}. Here is how you can generate some of these propositions ($n=1$):

$P_1 = x$, $P_2 = x \vee x$, $P_3 = x \vee x \vee x$, ...

$P_1 \equiv P_2 \equiv P_3 \equiv \dots$, but syntactically they are different.

How to establish equivalence of propositions?

$P \stackrel{?}{\equiv} Q$

Way #1: Write down the truth tables for both P and Q and check whether these tables are the same.

Exercise 3: P is a proposition in n variables; Q is a proposition in m variables; and $n \neq m$. Can it happen that $P \equiv Q$?

Exercise 4: Both P and Q are propositions in $\emptyset$ (zero) variables. Can they be equivalent?

Exercise 5: $\overbrace{x \leftrightarrow y}^{P} \stackrel{?}{\equiv} \overbrace{(x \to y) \wedge (y \to x)}^{Q}$

x	y	$x \leftrightarrow y$	$x \to y$	$y \to x$	$(x\to y)\wedge(y\to x)$
0	0	1	1	1	1
0	1	0	1	0	0
1	0	0	0	1	0
1	1	1	1	1	1
		P			Q

On the same assignments, P and Q have the same truth values. Thus, $P \equiv Q$, by definition.

Way #2: Using logical equivalences, try to "convert" P into Q (or Q into P).

The idea is to figure out what is equivalent to what in simple cases, and then to use these equivalences to "rewrite" more complex propositions.

Most popular logical equivalences:

$\neg(\neg x) \equiv x$ – double negation

$x \wedge T \equiv x$ (T – true), $x \vee F \equiv x$ – identity

$x \vee T \equiv T$, $x \wedge F \equiv F$ – domination

$x \wedge x \equiv x$ – idempotency

$x \vee \neg x \equiv T$, $x \wedge \neg x \equiv F$ – inverse laws

$x \vee y \equiv y \vee x$, $x \wedge y \equiv y \wedge x$ – commutativity

$x \wedge (y \wedge z) \equiv (x \wedge y) \wedge z$, $x \vee (y \vee z) \equiv (x \vee y) \vee z$ – associativity

$x \wedge (y \vee z) \equiv (x \wedge y) \vee (x \wedge z)$, $x \vee (y \wedge z) \equiv (x \vee y) \wedge (x \vee z)$ – distributive laws

$x \vee x \wedge y \equiv x$, $x \wedge (x \vee y) \equiv x$ – absorption

$\neg(x \vee y) \equiv \neg x \wedge \neg y$, $\neg(x \wedge y) \equiv \neg x \vee \neg y$ – De Morgan's laws

Each of these equivalences can be proven using Way #1. Some can be proven using other equivalences from this list.

Exercise 6: Prove that $x \wedge (x \vee y) \equiv x$ without using truth tables.

$\underline{x \wedge (x \vee y)} \equiv$ (identity) $\equiv$
$\equiv (x \vee F) \wedge (x \vee y) \equiv$ (distributivity) $\equiv$
$\equiv x \vee (F \wedge y) \equiv$ (domination) $\equiv$
$\equiv x \vee F \equiv$ (identity) $\equiv$
$\equiv \underline{x}$

Additional logical equivalences for implication:

Main: $p \to q \equiv \neg p \vee q$ (this may be seen as the definition of $\to$)
$p \to q \equiv \neg q \to \neg p$ (contraposition)

$p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$ (this may be seen as the definition of $\leftrightarrow$)
$p \leftrightarrow q \equiv \neg q \leftrightarrow \neg p \equiv \neg p \leftrightarrow \neg q$

Extra: $(p \to q) \wedge (p \to r) \equiv p \to (q \wedge r)$
$(p \to q) \vee (p \to r) \equiv p \to (q \vee r)$
$(p \to r) \wedge (q \to r) \equiv (p \vee q) \to r$
$(p \to r) \vee (q \to r) \equiv (p \wedge q) \to r$

$p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$
$\neg (p \leftrightarrow q) \equiv p \leftrightarrow \neg q \equiv \neg p \leftrightarrow q$

Note: Proving $P \equiv Q$ is equivalent to proving "$P \leftrightarrow Q$ is a tautology", that is, $P \leftrightarrow Q$ is True under every possible assignment of truth values to the propositional variables.

Exercise 7: Let us introduce one more logical connective – Sheffer stroke – defined as follows $x \mid y \overset{def}{\equiv} \neg (x \wedge y)$ (sometimes, it is called NAND).
Prove that $(x \mid y) \mid (x \mid y) \equiv x \wedge y$.

$\underline{(x \mid y) \mid (x \mid y)} \equiv$ (definition of $\mid$) $\equiv \neg(x \wedge y) \mid \neg(x \wedge y) \equiv$ (definition of $\mid$) $\equiv$
$\equiv \neg\big(\neg(x \wedge y) \wedge \neg(x \wedge y)\big) \equiv$ (idempotency) $\equiv \neg(\neg(x \wedge y)) \equiv$ (double negation) $\equiv$
$\equiv \underline{x \wedge y}$

Note: When the number of variables is small (say, 1–3 variables), it may be easier to compare truth tables rather than to look for equivalences.

Disjunctive Normal Form: Given a proposition defined with a formula (say, $x \wedge y \to z$), we can write down its truth table. What if we are given the truth table alone? Can we, having this table, construct a formula for that proposition? We also want the construction algorithm to be very "mechanical", so that even a dumb computer could do it without help from the human. One way to perform such a construction is known as constructing a disjunctive normal form of a proposition (DNF). (There are other ways; DNF is just one of the simplest.)

Building DNF for proposition P:

1) Look at every row of the P's truth table where P's truth value is True (or 1). We will write down a term for each such row. DNF will be a disjunction of all the obtained terms.

2) A term corresponding to a certain truth assignment – a row in the truth table – is a conjunction of all the variables P depends on, where a variable is included in the conjunction negated if it is False in the corresponding assignment and included as is if it is True.

Example: Write down a formula for proposition P defined as follows:

x	y	z	P
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

(Note: DNF is usually not the most concise form of a proposition. It can be "minimized" either using Equivalences or "Karnaugh maps".)

$$P = \underbrace{\neg x \wedge \neg y \wedge \neg z}_{\text{term 1}} \vee \underbrace{\neg x \wedge y \wedge \neg z}_{\text{term 2}} \vee \underbrace{\neg x \wedge y \wedge z}_{\text{term 3}} \vee \underbrace{x \wedge y \wedge z}_{\text{term 4}}$$

Note: DNF can be constructed for every proposition, and it uses only $\wedge$, $\vee$, and $\neg$. Thus, any proposition can be expressed in terms of $\wedge, \vee, \neg$. Consequently, the system of logical connectives $\{\vee, \wedge, \neg\}$ is functionally complete. System $\{\vee, \wedge\}$ is incomplete (negation cannot be expressed in terms of $\vee$ and $\wedge$). On the other hand, the system $\{\mid\}$ (recall Sheffer stroke) is complete (any proposition can be written using only Sheffer stroke).

Rules of inference:

What if we need to prove consequence of two propositions? ($P \Rightarrow Q$)
For one thing, we can just use logical equivalences and prove that $P \to Q$ is a tautology. For another thing, we can apply rules of inference to infer Q from P.

def: A rule of inference is an implication of the form $(x_1 \wedge x_2 \wedge x_3 \wedge \ldots \wedge x_k) \to y$, where $k = 0, 1, 2, 3, \ldots$, but most often $k = 2$ or 3. Rules of inference are usually written in the following form:

$$\begin{array}{l} x_1 \\ x_2 \\ \vdots \\ x_k \\ \hline \therefore\ y \end{array}$$

Examples:

1) $k=0 \Rightarrow x_1 \wedge x_2 \wedge \ldots \wedge x_k$ is just True (to understand why, recall the identity equivalence for conjunction $p \wedge T \equiv p$)

Hence, the rule of inference looks as follows

$$\begin{array}{l} \text{True} \\ \hline \therefore\ y \end{array} \quad \left(\text{or} \ \begin{array}{l} \\ \hline \therefore\ y \end{array}, \text{ but empty space above the line stands for True} \right).$$

For this to be a rule of inference, (True) $\to y$ must be a tautology (by definition of a rule of inference). It is such iff y is a tautology. Hence, the following are examples of rules of inference with $k=0$:

$$\begin{array}{l} \text{True} \\ \hline \therefore\ \text{True} \end{array}$$ - a rather useless rule of inference (we will see it later, while doing inferences)

$$\begin{array}{l} \text{True} \\ \hline \therefore\ x \vee \neg x \end{array}, \quad \begin{array}{l} \text{True} \\ \hline \therefore\ \neg(x \oplus x) \end{array}, \quad \ldots$$

2) We have seen the following logical equivalence:

$$(p \to q) \wedge (p \to r) \equiv p \to (q \wedge r)$$

We can "rewrite" it as a rule of inference (this can be done for all equivalences):

$$\begin{array}{l} p \to q \\ p \to r \\ \hline \therefore\ p \to (q \wedge r) \end{array}$$

Most popular rules of inference:

$$\begin{array}{l} p \\ p \to q \\ \hline \therefore q \end{array}$$ - modus ponens

$$\begin{array}{l} p \vee q \\ \neg p \\ \hline \therefore q \end{array}$$ - disjunctive syllogism

$$\begin{array}{l} \neg q \\ p \to q \\ \hline \therefore \neg p \end{array}$$ - modus tolens

$$\begin{array}{l} p \\ q \\ \hline \therefore p \wedge q \end{array}$$ - conjunction

$$\begin{array}{l} p \vee q \\ \neg p \vee r \\ \hline \therefore q \vee r \end{array}$$ - resolution

$$\begin{array}{l} p \wedge q \\ \hline \therefore p \end{array}$$ - simplification (aka "conjunctive simplification")

$$\begin{array}{l} p \to q \\ q \to r \\ \hline \therefore p \to r \end{array}$$ - hypothetical syllogism

$$\begin{array}{l} p \\ \hline \therefore p \vee q \end{array}$$ - addition (aka "disjunctive amplification")

$$\begin{array}{l} p \to r \\ q \to r \\ \hline \therefore (p \vee q) \to r \end{array}$$ - proof by cases

$$\begin{array}{l} p \to q \\ q \to p \\ \hline \therefore p \leftrightarrow q \end{array}$$ - biconditional introduction

$$\begin{array}{l} p \leftrightarrow q \\ p \\ \hline \therefore q \end{array}$$ - biconditional ellimination

Note: You can do any proof using only modus tolens. But it is hard. We need a variety of rules of inference to write shorter proofs.

We will use rules of inference to write proofs. For now, we are going to prove arguments of the form "If (x_1 and x_2 and ... and x_k) then y." x_i are our premises, and y is the conclusion. An argument can be written in the same form as the rules of inference:

$$\begin{array}{l} x_1 \\ \vdots \\ x_k \\ \hline \therefore y \end{array}$$

Example of an argument (not a proof so far):

$$\begin{array}{ll} \left.\begin{array}{l} \neg p \\ p \vee \neg q \\ r \leftrightarrow q \end{array}\right\} & \text{premises} \\ \hline \therefore \neg r \quad \} & \text{conclusion} \end{array}$$

An argument is valid or correct if the corresponding proposition $\overbrace{(x_1 \wedge \dots \wedge x_k)}^{\text{premises}} \to \overbrace{y}^{\text{conclusion}}$ is a tautology. We will prove validity / corectness using rules of inference.

Exercise 8: Prove the argument

$$\frac{(\psi \vee t) \wedge (\neg t \vee \varphi)}{\therefore (\psi \vee \varphi)} \quad \begin{matrix} \} \text{ premise} \\ \} \text{ conclusion} \end{matrix}$$

Proof:

1. $(\psi \vee t) \wedge (\neg t \vee \varphi)$ (premise)
2. $(\psi \vee t)$ (conjunctive simplification at step 1)
   missing step; first, need to use commutativity/associativity
3. $(\neg t \vee \varphi)$ (conjunctive simplification at step 1)
4. $(t \vee \psi)$ (commutativity at step 2)
5. $(\psi \vee \varphi)$ (resolution at steps 4 and 3). Conclusion.

Make sure resolution is the reader / on the slides; if not, do not use it.

$\square$ (– "end of proof")

Exercise 9: Prove the argument from Exercise 8 without using resolution. (Can use only those rules, given in the instructor's lecture notes.)

Proof:

1. $(\psi \vee t) \wedge (\neg t \vee \varphi)$ (premise)
2. $(\psi \vee t)$ (conjunctive simplification at step 1)
   missing step; first, need to use commutativity/associativity
3. $(\neg t \vee \varphi)$ (conjunctive simplification at step 1)
4. $(\psi \vee t) \vee \varphi$ (disjunctive amplification at step 2)
5. $(\psi \vee \varphi) \vee t$ (associativity and commutativity at step 4)
6. $(\neg t \vee \varphi) \vee \psi$ (disjunctive amplification at step 3)
7. $(\psi \vee \varphi) \vee \neg t$ (associativity and commutativity at step 6)
8. $((\psi \vee \varphi) \vee t) \wedge ((\psi \vee \varphi) \vee \neg t)$ (conjunction at steps 5 and 7)
   Notice: 5 and 7 a "glued" together before distributivity is used
9. $(\psi \vee \varphi) \vee (t \wedge \neg t)$ (distributive law at step 8)
10. $(\psi \vee \varphi) \vee F$ (inverse at step 9)
11. $(\psi \vee \varphi)$ (identity at step 10) Conclusion.

$\square$

Note: distributive law, inverse, identity – have not been explicitly given as rules of inference. But we know corresponding logical equivalences and can easily "convert" them to the rules of inference (like it was done in the second example on Page 7).

That is not the distributivity used at step 9

$$x \wedge (y \vee z) \equiv (x \wedge y) \vee (x \wedge z) \quad \sim \quad \frac{x \wedge (y \vee z)}{\therefore (x \wedge y) \vee (x \wedge z)}, \quad \boxed{\frac{(x \wedge y) \vee (x \wedge z)}{\therefore x \wedge (y \vee z)}}$$

Logical Equivalence — Related Rules of Inference

Quantifiers discussed in another document