Propositional Logic

Lucia Moura

Winter 2017

Proposition

A proposition is a declarative sentence that is either true or false.

Which ones of the following sentences are propositions?

- Ottawa is the capital of Canada.
- Buenos Aires is the capital of Brazil.
- $2 + 2 = 4$
- $2 + 2 = 5$
- if it rains, we don't need to bring an umbrella.
- $x + 2 = 4$
- $x + y = z$
- When does the bus come?
- Do the right thing.

Propositional variable and connectives

We use letters $p, q, r, \ldots$ to denote **propositional variables** (variables that represent propositions).

We can form new propositions from existing propositions using **logical operators** or **connectives**. These new propositions are called **compound propositions**.

Summary of connectives:

name	nickname	symbol
negation	NOT	$\neg$
conjunction	AND	$\wedge$
disjunction	OR	$\vee$
exclusive-OR	XOR	$\oplus$
implication	implies	$\rightarrow$
biconditional	if and only if	$\leftrightarrow$

Meaning of connectives

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \oplus q$	$p \rightarrow q$	$p \leftrightarrow q$
T	T	F	T	T	F	T	T
T	F	F	F	T	T	F	F
F	T	T	F	T	T	T	F
F	F	T	F	F	F	T	T

WARNING:
Implication ($p \rightarrow q$) causes confusion, specially in line 3: "F $\rightarrow$ T" is true.
One way to remember is that the rule to be obeyed is
"if the premise p is true then the consequence q must be true."
The only truth assignment that falsifies this is $p = T$ and $q = F$.

Truth tables for compound propositions

Construct the truth table for the compound proposition:
$(p \vee \neg q) \rightarrow (p \wedge q)$

p	q	$\neg q$	$p \vee \neg q$	$p \wedge q$	$(p \vee \neg q) \rightarrow (p \wedge q)$
T	T	F			
T	F	T			
F	T	F			
F	F	T			

Propositional Equivalences

A basic step is math is to replace a statement with another with the same truth value (equivalent).
This is also useful in order to reason about sentences.
Negate the following phrase:

"Miguel has a cell phone and he has a laptop computer."

- p="Miguel has a cell phone"
  q="Miguel has a laptop computer."
- The phrase above is written as $(p \wedge q)$.
- Its negation is $\neg(p \wedge q)$, which is logically equivalent to $\neg p \vee \neg q$. (De Morgan's law)
- This negation therefore translates to:
  "Miguel does not have a cell phone or he does not have a laptop computer."

Truth assignments, tautologies and satisfiability

Definition

Let X be a set of propositions.
A **truth assignment** (to X) is a function $\tau : X \rightarrow \{true, false\}$ that assigns to each propositional variable a truth value. (A truth assignment corresponds to one row of the truth table)
If the truth value of a compound proposition under truth assignment τ is $true$, we say that τ **satisfies** P, otherwise we say that τ **falsifies** P.

- A compound proposition P is a **tautology** if every truth assignment satisfies P, i.e. all entries of its truth table are $true$.
- A compound proposition P is **satisfiable** if there is a truth assignment that satisfies P; that is, at least one entry of its truth table is true.
- A compound proposition P is **unsatisfiable (or a contradiction)** if it is not satisfiable; that is, all entries of its truth table are false.

Examples: tautology, satisfiable, unsatisfiable

For each of the following compound propositions determine if it is a tautology, satisfiable or unsatisfiable:

- $(p \vee q) \wedge \neg p \wedge \neg q$
- $p \vee q \vee r \vee (\neg p \wedge \neg q \wedge \neg r)$
- $(p \rightarrow q) \leftrightarrow (\neg p \vee q)$

Logical implication and logical equivalence

Definition

A compound proposition p **logically implies** a compound proposition q (denoted $p \Rightarrow q$) if $p \rightarrow q$ is a tautology.
Two compound propositions p and q are **logically equivalent** (denoted $p \equiv q$, or $p \Leftrightarrow q$) if $p \leftrightarrow q$ is a tautology.

Theorem

Two compound propositions p and q are logically equivalent if and only if p logically implies q and q logically implies p.

In other words: two compound propositions are logically equivalent if and only if they have the same truth table.

Logically equivalent compound propositions

Using truth tables to prove that $(p \to q)$ and $\neg p \vee q$ are logically equivalent, i.e.

$$(p \to q) \equiv \neg p \vee q$$

p	q	$\neg p$	$\neg p \vee q$	$p \to q$
T	T	F	T	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

What is the problem with this approach?

Truth tables versus logical equivalences

Truth tables grow exponentially with the number of propositional variables!

A truth table with n variables has 2^n rows.

Truth tables are practical for small number of variables, but if you have, say, 7 variables, the truth table would have 128 rows!

Instead, we can prove that two compound propositions are logically equivalent by using known logical equivalences ("equivalence laws").

Summary of important logical equivalences I

TABLE 6 Logical Equivalences.

Equivalence	*Name*
$p \wedge \mathbf{T} \equiv p$ $p \vee \mathbf{F} \equiv p$	Identity laws
$p \vee \mathbf{T} \equiv \mathbf{T}$ $p \wedge \mathbf{F} \equiv \mathbf{F}$	Domination laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent laws
$\neg(\neg p) \equiv p$	Double negation law
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative laws
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative laws
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive laws
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's laws
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption laws
$p \vee \neg p \equiv \mathbf{T}$ $p \wedge \neg p \equiv \mathbf{F}$	Negation laws

Note T is the compound composition that is always true, and F is the compound composition that is always false.

Summary of important logical equivalences II

TABLE 7 Logical Equivalences Involving Conditional Statements.

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

TABLE 8 Logical Equivalences Involving Biconditionals.

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

Rosen, page 24-25.

Proving new logical equivalences

Use known logical equivalences to prove the following:

1. Prove that $\neg(p \to q) \equiv p \wedge \neg q$.
2. Prove that $(p \wedge q) \to (p \vee q)$ is a tautology.

Normal forms for compound propositions

- A literal is a propositional variable or the negation of a propositional variable.
- A term is a literal or the conjunction (and) of two or more literals.
- A clause is a literal or the disjunction (or) of two or more literals.

Definition

A compound proposition is in **disjunctive normal form** (DNF) if it is a term or a disjunction of two or more terms. (i.e. an OR of ANDs).
A compound proposition is in **conjunctive normal form** (CNF) if it is a clause or a conjunction of two or more clauses. (i.e. and AND of ORs)

Disjunctive normal form (DNF)

	x	y	z	$x \vee y \to \neg x \wedge z$
1	F	F	F	T
2	F	F	T	T
3	F	T	F	F
4	F	T	T	T
5	T	F	F	F
6	T	F	T	F
7	T	T	F	F
8	T	T	T	F

The formula is satisfied by the truth assignment in row 1 **or**
by the truth assignment in row 2 **or** by the truth assignment in row 4.
So, its DNF is : $(\neg x \wedge \neg y \wedge \neg z) \vee (\neg x \wedge \neg y \wedge z) \vee (\neg x \wedge y \wedge z)$

Conjunctive normal form (CNF)

	x	y	z	$x \vee y \to \neg x \wedge z$
1	F	F	F	T
2	F	F	T	T
3	F	T	F	F
4	F	T	T	T
5	T	F	F	F
6	T	F	T	F
7	T	T	F	F
8	T	T	T	F

The formula is **not** satisfied by the truth assignment in row 3 **and** in row 5 **and** in row 6 **and** in row 7 **and** in row 8. So:, it is log. equiv. to:
$\neg(\neg x \wedge y \wedge \neg z) \wedge \neg(x \wedge \neg y \wedge \neg z) \wedge \neg(x \wedge \neg y \wedge z) \wedge \neg(x \wedge y \wedge \neg z) \wedge \neg(x \wedge y \wedge z)$
apply DeMorgan's law to obtain its CNF:
$(x \vee \neg y \vee z) \wedge (\neg x \vee y \vee z) \wedge (\neg x \vee y \vee \neg z) \wedge (\neg x \vee \neg y \vee z) \wedge (\neg x \vee \neg y \vee \neg z)$

Boolean functions and the design of digital circuits

Let $B = \{false, true\}$ (or $B = \{0, 1\}$). A function $f : B^n \rightarrow B$ is called a boolean function of degree n.

Definition

A compound proposition P with propositions $x_1, x_2, \ldots, x_n$ represents a Boolean function f with arguments $x_1, x_2, \ldots, x_n$ if for any truth assignment τ, τ satisfies P if and only if
$f(\tau(x_1), \tau(x_2), \ldots, \tau(x_n)) = true$.

Theorem

Let P be a compound proposition that represents a boolean function f. Then, a compound proposition Q also represents f if and only if Q is logically equivalent to P.

Complete set of connectives (functionally complete)

Theorem

Every boolean formula can be represented by a compound proposition that uses only connectives $\{\neg, \wedge, \vee\}$ (i.e. $\{\neg, \wedge, \vee\}$ is **functionally complete** *).*

Proof: use DNF or CNF!

This is the basis of circuit design:

In digital circuit design, we are given a **functional specification** of the circuit and we need to construct a **hardware implementation**.

functional specification $=$ number n of inputs $+$ number m of outputs $+$ describe outputs for each set of inputs (i.e. m boolean functions!)

Hardware implementation uses logical gates: or-gates, and-gates, inverters.

The functional specification corresponds to m boolean functions which we can represent by m compound propositions that uses only $\{\neg, \wedge, \vee\}$, that is, its hardware implementation uses inverters, and-gates and or-gates.

Boolean functions and digital circuits

Consider the boolean function represented by $x \vee y \rightarrow \neg x \wedge z$.

Give a digital circuit that computes it, using only $\{\wedge, \vee, \neg\}$.
This is always possible since $\{\wedge, \vee, \neg\}$ is functionally complete (e.g. use DNF or CNF).

Give a digital circuit that computes it, using only $\{\wedge, \neg\}$.
This is always possible, since $\{\wedge, \neg\}$ is **functionally complete**:
Proof: Since $\{\wedge, \vee, \neg\}$ is functionally complete, it is enough to show how to express $x \vee y$ using only $\{\wedge, \neg\}$:
$(x \vee y) \equiv \neg(\neg x \wedge \neg y)$

Give a digital circuit that computes it, using only $\{\vee, \neg\}$.
Prove that $\{\vee, \neg\}$ is **functionally complete**.