CSI2101 Discrete Structures Winter 2009: Propositional Logic: normal forms, boolean functions and circuit design

Lucia Moura

Winter 2009

Truth assignments, tautologies and satisfiability

Definition

Let X be a set of propositions (also called propositional variables).
A **truth assignment** (to X) is a function $\tau : X \to \{true, false\}$ that assigns to each propositional variable a truth value.
If the truth value of a compound propositional (or propositional formula) 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.
A compound proposition P is **satisfiable** if there is a truth assignment that satisfies P.
A compound proposition P is **unsatisfiable (or a contradiction)** if it is not satisfiable. there is a truth assignment that satisfies P.

Examples: tautology, satisfiable, unsatisfiable

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

- $(x \lor y) \land \neg x \land \neg y$
- $z \lor y \lor x \lor (\neg x \land \neg y \land \neg z)$
- $(x \to y) \leftrightarrow (\neg x \lor y)$

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.

Summary of important logical equivalences

Textbook's Table 6: identity, domination, idempotent, double negation, commutative, associative, distributive, De Morgan's, absorption and negation laws (page 24).
Table 7: logical equivalences involving conditional statements and
Table 8: logical equivalences involving biconditionals (page 25).

Normal forms for compound propositions

- A literal is a compound proposition that consists of a proposition or the negation of a proposition.
- 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. A compound proposition is in **conjunctive normal form** (CNF) if it is a clause or a conjunction of two or more clauses.

Disjunctive normal form (DNF)

	x	y	z	$x \vee y \rightarrow \neg x \wedge z$
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	$x \vee y \rightarrow \neg x \wedge z$
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:
$\neg(\neg x \wedge y \wedge \neg z) \wedge \neg(x \wedge \neg y \wedge \neg z) \wedge \neg(x \wedge \neg y \wedge z) \wedge \neg(x \wedge y \wedge \neg z) \wedge \neg(x \vee y \vee z)$
apply DeMorgan's law to obtain its CNF:
$(x \vee \neg y \vee z) \wedge (\neg x \vee y \vee z) \wedge (\neg x \vee y \vee \neg z) \wedge (\neg x \vee \neg y \vee z) \wedge (\neg x \wedge \neg y \wedge \neg z)$

Boolean functions and the design of digital circuits

Let $B = \{false, true\}$ (or $B = \{0, 1\}$). A function $f : B^n \longrightarrow B$ 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 $\{\neg, \wedge, \vee\}$.

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 $\{\wedge, \vee, \neg\}$.

Give a digital circuit that computes it, using only $\{\wedge, \neg\}$.
Is this always possible? Why?

Give a digital circuit that computes it, using only $\{\vee, \neg\}$.
Is this always possible? Why?