Discrete Mathematics and Statistics - CPT107

Xi'an Jiaotong-Liverpool University
西交利物浦大学

# Propositional Logic

Part 5. Propositional Logic

Credits: These slides are based on slides prepared by Frank Wolter.

Reading: Discrete Mathematics for Computer Scientists, J.K. Truss, Sections 2.1 and 2.2.

# Contents

# Propositions

A **proposition** is a *declarative sentence* that is either *true* or *false* (but not both).

A **proposition** is a *possible condition* of the world that is either *true* or *false*

*Examples*:

1+1=2

2+2=5

Today is Tuesday.

The area of logic that deals with propositions is called propositional logic.

Propositional Logic is concerned with propositions and their interrelationships.

# Propositions

In Propositional Logic, there are two types of sentences:

- *simple sentences* and

- *compound sentences*.

Simple sentences express simple facts about the world.

Compound sentences express logical relationships between the simpler sentences of which they are composed.

# Propositional Variables

A variable that represents propositions is called a *propositional variable.*

The simplest formulas are "propositional variables" which we write using letters such as *p, q* or *r* or using indexed letters such as $p_o,p_i,p_2,\ldots$

Propositional variables are called atomic formulas.

# Formulas

More complicated propositional expressions, called **formulas** or *well-formed formulas* (wffs), can be built from the proposition letters using the propositional connectives and parentheses.

A formula is defined "syntactically" — you build a formula following rules.

This has nothing to do with what the formula means (which is called "semantics")

"Parsing" the formula means determining how the formula is built, following the rules.

# Approach

We are going to present the propositional logic as a formal language:

- we first present the syntax of the language

- then the semantics of the language.

# Syntax of Propositional Logic

The *alphabet of propositional logic* consists of

- an infinite set $p, q, r, p_o, p_i, p_2,$ **...** of *atomic formulas* (or propositional variables);
- the logical connectives $\neg$ ("not"), $\wedge$ ("and"), and $\vee$ ("or");
- brackets: ( and ).

# Formulas of propositional logic

The set P of all *formulas of propositional logic* is defined inductively:

- all atomic formulas are formulas;
- if *P* is a formula, then $\neg$P is a formula;
- if P and *Q* are formulas, then (P $\wedge$ *Q)* is a formula;
- if P and *Q* are formulas, then (P v *Q)* is a formula;
- Nothing else is a formula.

# Semantics of Propositional Logic

- the correct meaning

We can associate a unique binary tree to each proposition, called the parse tree.

The formation tree contains all sub-formulas of a formula, starting with the formula at its root and breaking it down into its sub-formulas until you reach the propositional variables at its leaves.

# Assigning Meanings to Formulas

We know that each formula corresponds to a unique binary tree.

We can evaluate the formula by:

- giving each propositional variable an interpretation.

- defining the meaning of each logical connective

- propagate the truth values from the leaves to the root in a unique way, so that we get an unambiguous evaluation of each formula.

# Parse Trees

A parse tree is a tree representation of how the formula is constructed.

- The parse tree of an atomic formula has one node, labelled with the formula.
- The parse tree of a formula $\neg P$ has a root labelled "$\neg$". The child of the root is the parse tree for the formula $P$.
- The parse tree of a formula $(P \wedge Q)$ has a root labelled "$\wedge$". The left child of the root is the parse tree for the formula $P$. The right child of the root is the parse tree for the formula $Q$.
- The parse tree of a formula $(P \vee Q)$ has a root labelled "$\vee$". The left child of the root is the parse tree for the formula $P$. The right child of the root is the parse tree for the formula $Q$.

# Example

Represent the statements:

1. Logic is easy;
2. I eat toast;
3. Logic is easy or I eat toast;
4. It is false that logic is easy;
5. Logic is not easy;
6. Logic is not easy and I do not eat toast.

# Example

Represent the following statements:

1. Peter is in room 3 and Mike is in room 2;
2. Joe is in room 1 or in room 3;
3. If Peter is in room 2, then Mike is in room 3;

# Example

Suppose now we want to reason about the distribution of Peter, Mike and Joe over the rooms 1 to 3.

- “Peter is in at least one of the rooms 1-3”
- “Peter is in room 1 and not in room 2 or 3”
- “Peter is in room 2 and not in room 1 or 3”
- “Peter is in room 3 and not in room 1 or 2”
- “Peter is in exactly one of the rooms 1-3”
- “Peter and Mike are not in the same room”

# Truth Values

An *interpretation* $I$ is a function which assigns to any atomic formula $p_i$ a truth value

$$I(p_i) \in \{0, 1\}.$$

- If $I(p_i) = 1$, then $p_i$ is called *true* under the interpretation $I$.
- If $I(p_i) = 0$, then $p_i$ is called *false* under the interpretation $I$.

Given an assignment $I$ we can compute the truth value of compound formulas step by step using so-called truth tables.

# Truth tables: negation

The negation $\neg P$ of a formula $P$ is true when $P$ is false and false otherwise:

**Definition** Suppose an interpretation $I$ is given and we know the value $I(P)$. Then the value $I(\neg P)$ is computed by

$$I(\neg P) = \begin{cases} 0 & \text{if} \quad I(P) = 1 \\ 1 & \text{if} \quad I(P) = 0 \end{cases}$$

Corresponding truth table:

| $P$ | $\neg P$ |
|---|---|
| 1 | 0 |
| 0 | 1 |

# Truth tables: conjunction

The conjunction $(P \wedge Q)$ is true if and only if both $P$ and $Q$ are true.

**Definition** Suppose an interpretation $I$ is given and we know $I(P)$ and $I(Q)$. Then

$$I(P \wedge Q) = \begin{cases} 1 & \text{if} \quad I(P) = 1 \text{ and } I(Q) = 1 \\ 0 & \text{if} \quad I(P) = 0 \text{ or } I(Q) = 0 \end{cases}$$

Corresponding truth table:

| $P$ | $Q$ | $(P \wedge Q)$ |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

# Truth tables: disjunction

The disjunction $(P \vee Q)$ is true if and only $P$ is true or $Q$ is true.

**Definition** Suppose an interpretation $I$ is given and we know $I(P)$ and $I(Q)$. Then

$$I(P \vee Q) = \begin{cases} 1 & \text{if} \quad I(P) = 1 \text{ or } I(Q) = 1 \\ 0 & \text{if} \quad I(P) = 0 \text{ and } I(Q) = 0 \end{cases}$$

Corresponding truth table:

| $P$ | $Q$ | $(P \vee Q)$ |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

# Truth under an interpretation

So, given an interpretation $I$, we can compute the truth value of any formula $P$ under $I$.

- If $I(P) = 1$, then $P$ is called true under the interpretation $I$.
- If $I(P) = 0$, then $P$ is called false under the interpretation $I$.

# Example

List the Interpretations $I$ such that $P = ((p_1 \vee \neg p_2) \wedge p_3)$ is true under $I$.

# Truth table for $(\neg P \vee Q)$

| $P$ | $Q$ | $\neg P$ | $(\neg P \vee Q)$ |
|---|---|---|---|
| 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 |

$(\neg P \vee Q)$ represents the assertion 'if $P$ is true, then $Q$ is true'.

Define a 'new' connective $\rightarrow$ by:

$$(P \rightarrow Q) = (\neg P \vee Q).$$

implication

P → Q is true always except when P is true and Q is false

# Truth table for $((P \to Q) \wedge (Q \to P))$

| $P$ | $Q$ | $(P \to Q)$ | $(Q \to P)$ | $((P \to Q) \wedge (Q \to P))$ |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |

$((P \to Q) \wedge (Q \to P))$ represents the assertion '$P$ is true if and only if $Q$ is true'. Define a 'new' connective $\leftrightarrow$ by:

$$(P \leftrightarrow Q) := ((P \to Q) \wedge (Q \to P)).$$

# Tautology

**Definition** A tautology is a formula which is true under all interpretations.

**Example** All formulas of the form $(P \lor \neg P)$ are tautologies, because

$$I(P \lor \neg P) = 1$$

for all interpretations $I$:

| $P$ | $\neg P$ | $(P \lor \neg P)$ |
|---|---|---|
| 1 | 0 | 1 |
| 0 | 1 | 1 |

Define a 'new' connective $\top$ by $\boxed{\top := (p_1 \lor \neg p_1).}$

# Contradiction

**Definition** A contradiction is a formula which is false under all interpretations.

**Example** All formulas of the form $(P \wedge \neg P)$ are contradictions, because

$$I(P \wedge \neg P) = 0$$

for all interpretations $I$:

| $P$ | $\neg P$ | $(P \wedge \neg P)$ |
|---|---|---|
| 1 | 0 | 0 |
| 0 | 1 | 0 |

Define a 'new' connective $\perp$ by $\boxed{\perp := (p_1 \wedge \neg p_1).}$

# Semantic consequence

**Definition** Suppose $\Gamma$ is a finite set of formulas and $P$ is a formula. Then $P$ follows from $\Gamma$ ("is a semantic consequence of $\Gamma$") if the following implication holds for every interpretation $I$:

$$\text{If } I(Q) = 1 \text{ for all } Q \in \Gamma, \text{ then } I(P) = 1.$$

This is denoted by

$$\Gamma \models P.$$

# Example

Show $\{(p_1 \wedge p_2)\} \models (p_1 \vee p_2)$.

# Observations

The following conditions are equivalent:

1. $\{P\} \models Q$;
2. $(P \rightarrow Q)$ is a tautology;
3. $(P \land \neg Q)$ is a contradiction.

# Logical equivalence

**Definition** Two formulas $P$ and $Q$ are called equivalent if they have the same truth value under every possible interpretation. In other words, $P$ and $Q$ are equivalent if $I(P) = I(Q)$ for every interpretation $I$. This is denoted by

$$P \equiv Q.$$

# Laws for equivalences

The following equivalences can be checked by truth tables:

- Associative laws:

$$(P \vee (Q \vee R)) \equiv ((P \vee Q) \vee R), \quad (P \wedge (Q \wedge R)) \equiv ((P \wedge Q) \wedge R);$$

- Commutative laws:

$$(P \vee Q) \equiv (Q \vee P), \quad (P \wedge Q) \equiv (Q \wedge P);$$

- Identity laws:

$$(P \vee \bot) \equiv P, \ (P \vee \top) \equiv \top, \ (P \wedge \top) \equiv P, \ (P \wedge \bot) \equiv \bot;$$

- Distributive laws:

$$(P \wedge (Q \vee R)) \equiv ((P \wedge Q) \vee (P \wedge R)),\ (P \vee (Q \wedge R)) \equiv ((P \vee Q) \wedge (P \vee R));$$

- Complement laws:

$$P \vee \neg P \equiv \top,\ \neg\top \equiv \bot,\ \neg\neg P \equiv P, P \wedge \neg P \equiv \bot,\ \neg\bot \equiv \top;$$

- De Morgan's laws:

$$\neg(P \vee Q) \equiv (\neg P \wedge \neg Q),\ \neg(P \wedge Q) \equiv (\neg P \vee \neg Q).$$

# On logical equivalence

**Theorem** The relation $\equiv$ is an equivalence relation on $\mathcal{P}$.

**Proof**

- $\equiv$ is reflexive, since $(P \leftrightarrow P)$ is a tautology
- $\equiv$ is transitive, since $P \equiv Q$ and $Q \equiv R$ implies $P \equiv R$.
- $\equiv$ is symmetric, since $P \equiv Q$ implies $Q \equiv P$.

Xi'an Jiaotong-Liverpool University 西交利物浦大学

# End of lecture

- **Summary**
  - Syntax: formulas and formal representations
  - Semantics: interpretations and truth tables
  - Tautologies
  - Contradictions
  - Semantic consequence
  - Logical equivalence

- **Reading**
  - Discrete Mathematics for Computer Scientists, J.K. Truss, Sections 2.1 and 2.2.