Mathematical Logic

Lecture 2: Propositional Logic — Syntax

Dr. CHOUDER R

Department of Computer Science

Level: First Year LMD — Licence Computer Science

Academic Year 2025-2026

Introduction

In propositional logic (also called **propositional calculus**) we use:

- **Statements (propositions)**
- **Logical connectives**

Key facts:

- Propositions are combined using logical connectives
- Each formula is built from propositional variables
- The formalization power is limited compared to predicate logic
- **Syntax** defines how expressions are correctly written

The Proposition

Definition

A proposition (assertion) is a statement that is:

- either **true**
- or **false**
- but not both

A statement is a sentence to which a truth value can be assigned.

Examples of Propositions

- Algiers is the capital of Algeria
- $5 - 2 = 3$
- $1 + 2 \geq 7$
- I am a student

Each of these has a definite truth value.

Not Propositions

Sentences that are not truth-valued:

- Questions — What is your family name?
- Commands — Restart your laptop!
- Exclamations — Oh, that is excellent!
- Pure expressions — $3 + 7$

Remark: Future statements may be considered propositions in classical logic since they can be true or false — interpretation may depend on context.

Propositional Variables

A propositional variable is denoted by:

$$p, q, r, \ldots$$

If needed:

$$p_1, p_2, p_3, \ldots$$

Each variable takes exactly one truth value:

True or **False**

Logical Connectives

Logical connectives build new propositions from others.

Main Connectives

Connective	Symbol	Meaning	Example
Negation	$\neg p$	not p	not raining
Conjunction	$p \wedge q$	p and q	$x < 5 \wedge x > 0$
Disjunction	$p \vee q$	p or q	$x < 5 \vee x > 0$
Implication	$p \Rightarrow q$	if p then q	If network OK $\Rightarrow$ internet works
Biconditional	$p \Leftrightarrow q$	iff	$x > 0 \Leftrightarrow$ positive

Operator Precedence

To avoid ambiguity (example: $p \wedge q \vee r$), we use:

Precedence order (highest first):

1. $\neg$
2. $\wedge$
3. $\vee$
4. $\Rightarrow$
5. $\Leftrightarrow$

Associativity (common convention):

- $\wedge, \vee$ usually left-associative. **Example:** $p \wedge q \wedge r \equiv (p \wedge q) \wedge r$.
- $\Rightarrow$ right-associative. **Example:** $p \Rightarrow q \Rightarrow r \equiv p \Rightarrow (q \Rightarrow r)$
- $\Leftrightarrow$ often non-associative. **Example:** $p \Leftrightarrow q \Leftrightarrow r$
  (ambiguous, must use parentheses) $(p \Leftrightarrow q) \Leftrightarrow r \neq p \Leftrightarrow (q \Leftrightarrow r)$
- Parentheses are always recommended.

Exercise — Translation

Translate into propositional logic:

1. It's not raining
2. It's raining and not snowing
3. If it rains then I take an umbrella
4. It rains or it snows, but not both
5. It's not raining, so I'm not taking the umbrella

Solution — Translation into Propositional Logic

Let:

- R : "It is raining"
- S : "It is snowing"
- U : "I take an umbrella"

1. It's not raining: $\neg R$
2. It's raining and not snowing: $R \wedge \neg S$
3. If it rains then I take an umbrella: $R \Rightarrow U$
4. It rains or it snows, but not both: $(R \vee S) \wedge \neg(R \wedge S)$
5. It's not raining, so I'm not taking the umbrella: $\neg R \Rightarrow \neg U$

Atomic Formula

A formula is **atomic** if it is:

- a propositional variable
- or a truth constant T or F

Propositional Formula

A propositional formula may contain:

- propositional variables
- unary connective: $\neg$
- binary connectives: $\wedge, \vee, \Rightarrow, \Leftrightarrow$
- parentheses

From Propositions to Propositional Logic

In propositional logic, we build formulas step by step:

- Start from simple **atomic propositions** $(p, q, r, \dots)$
- Combine them using **logical connectives** $(\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow)$
- Add parentheses to ensure correct structure
- Gradually construct more complex formulas from simpler ones

Well-Formed Formulas (WFF)

Construction rules:

1. Every atomic variable is a WFF
2. If p is WFF then $\neg p$ is WFF
3. If p, q are WFF then:

$$p \wedge q,\ p \vee q,\ p \Rightarrow q,\ p \Leftrightarrow q$$

   are WFF
4. If p is WFF then (p) is WFF

WFF — Examples

- p — WFF
- $p \Rightarrow q$ — WFF
- $p \neg q$ — not a WFF
- $\neg(\wedge q)$ — not a WFF

Common Student Mistakes

Students often make these mistakes:

- Forgetting parentheses: $p \land q \Rightarrow r$ (unclear grouping)
- Using invalid sequences: $\neg \land p$
- Omitting connectives: (pq) instead of $(p \land q)$
- Misplacing negation: $\neg(p \land q)$ vs $\neg p \land q$

Parsing Tree

Parsing trees help verify formula structure.

- Internal nodes = connectives
- Binary connectives $\rightarrow$ binary nodes
- Negation $\rightarrow$ unary node
- Leaves $\rightarrow$ propositional variables
- Each formula has a unique parsing tree

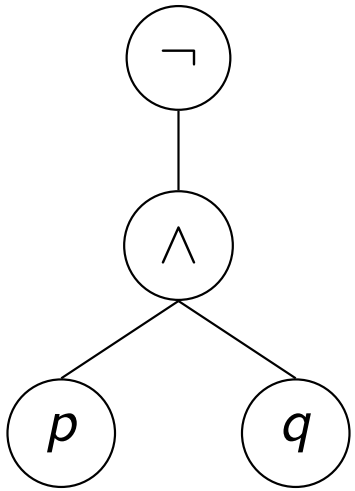
Parsing Tree : Example

Formula: $\neg(p \wedge q)$

Step-by-step:

- Main connective $= \neg$
- Subformula inside $= p \wedge q$
- Connective inside $= \wedge$

Parsing tree:



Depth of a Formula

Depth = number of branches from root to farthest leaf

Convention used here: atomic depth = 0

Formula	Depth
A	0
(A)	1
$((A) \wedge (B))$	2
$(\neg A) \wedge (B)$	2

Complexity of a Formula

Complexity = number of connectives

Formula	Complexity
A	0
$(A) \wedge (B)$	1
$(A \wedge B) \Rightarrow (C \wedge D)$	3

Subformulas

Definition: A **subformula** of a formula F is any part of F that is itself a formula.

Example:

$$F = (p \wedge (q \vee r))$$

Subformulas include:

$$p, \quad q, \quad r, \quad (q \vee r), \quad (p \wedge (q \vee r))$$

Importance: Subformulas are useful for:

- Parsing trees
- Proofs by induction on formulas
- Understanding complexity and depth

Example — Structure

Formula

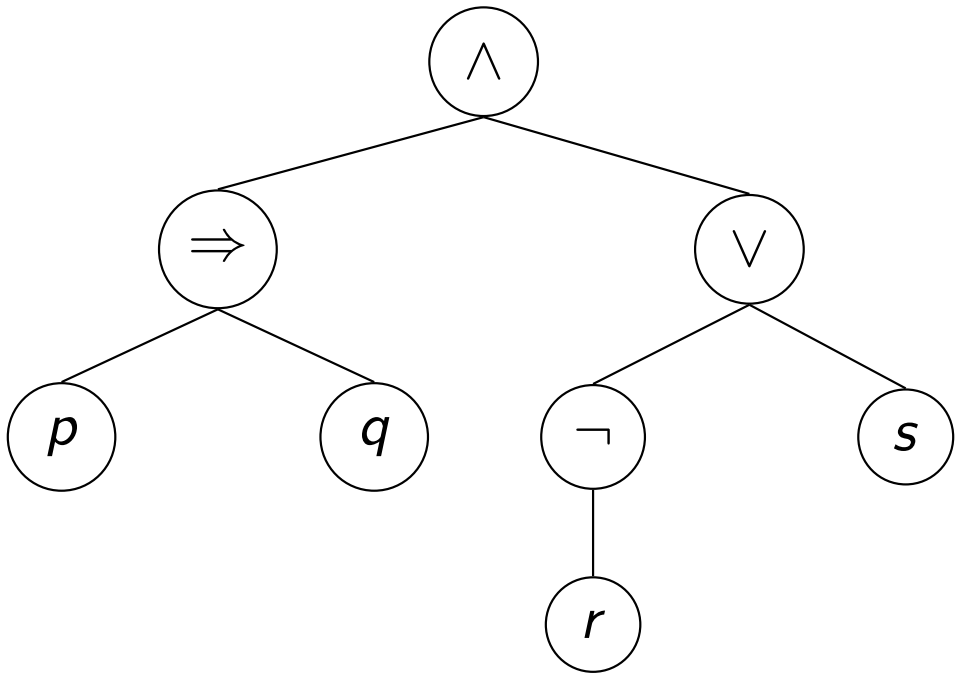
$$(p \Rightarrow q) \wedge (\neg r \vee s)$$

Step-by-step structure

- Main connective $= \wedge$
- Left subformula $= (p \Rightarrow q)$
- Right subformula $= (\neg r \vee s)$
- Inside right branch: negation of r

Example — Parsing Tree

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



Example — Subformulas and Measures

Subformulas

$$p,\ q,\ r,\ s,\ (p \Rightarrow q),\ (\neg r),\ (\neg r \vee s),\ (p \Rightarrow q) \wedge (\neg r \vee s)$$

Depth

Longest path:

$$\wedge \rightarrow \vee \rightarrow \neg \rightarrow r$$

Depth $= 3$

Complexity

Number of connectives $= 4$

$$\Rightarrow,\ \wedge,\ \neg,\ \vee$$

Exercise — Try Yourself

Parse the formula:

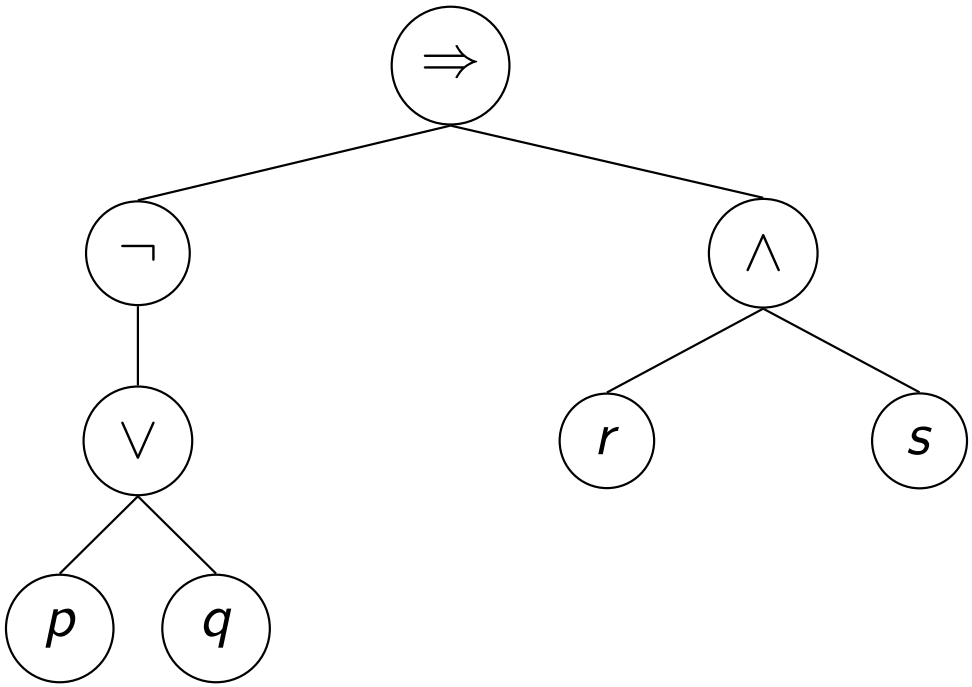
$$\neg(p \vee q) \Rightarrow (r \wedge s)$$

- Identify main connective
- Split into subformulas
- Draw the parsing tree

Solution on next slide

Solution

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



Substitution

Substitution = replace a variable with a formula.

Example:
Original:

$$p \Rightarrow q$$

Substitute $p := (r \wedge s)$
Result:

$$(r \wedge s) \Rightarrow q$$

Multiple Substitution

Example:
Formula:

$$p \vee q$$

Substitute:

$$p := (A \Rightarrow B), \quad q := \neg C$$

Result:

$$(A \Rightarrow B) \vee \neg C$$

Exercise

Formula:

$$F = \neg((p \wedge q) \Rightarrow r) \vee (\neg q \wedge r)$$

Tasks:

1. Check if F is a WFF.
2. Draw the parsing tree for F.
3. Compute the depth of F.
4. Compute the complexity of F.

Step 1 — Check WFF

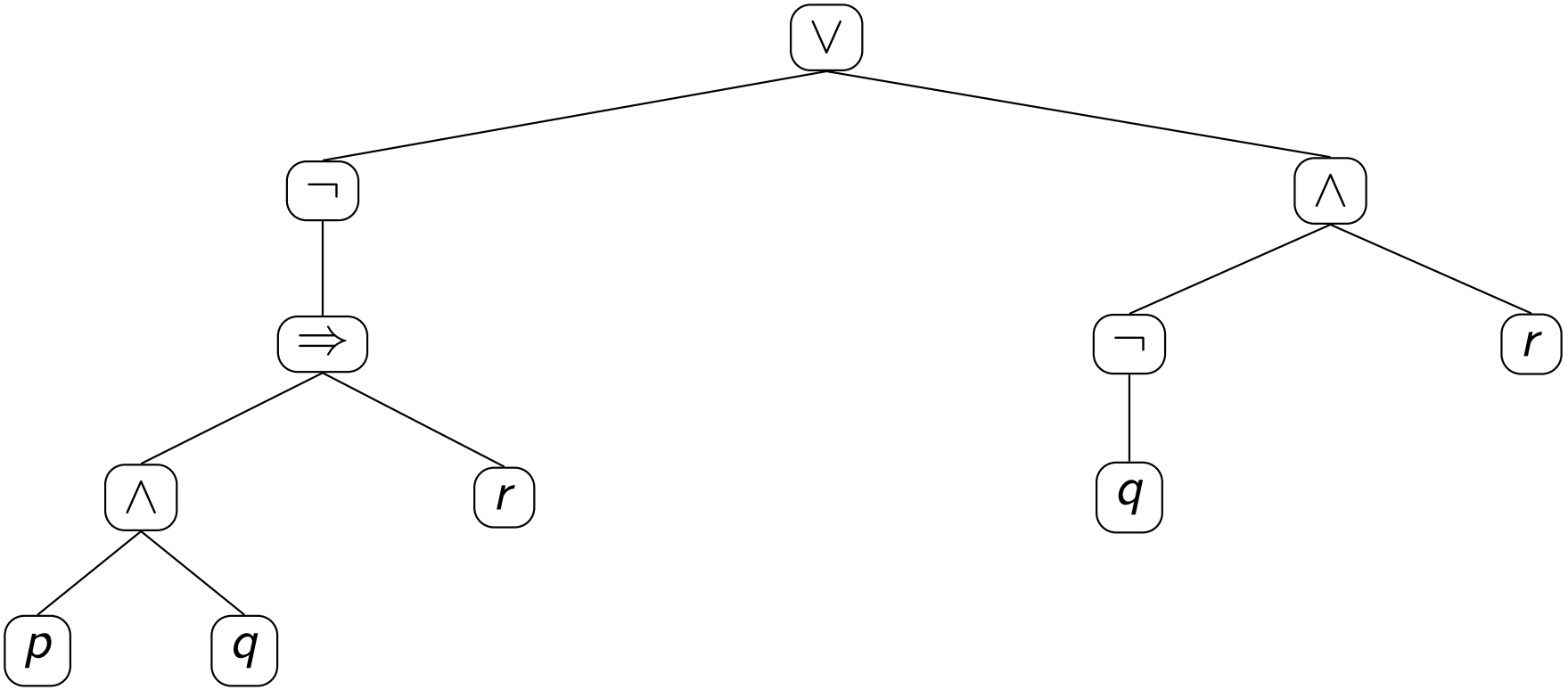
$$F = \neg((p \wedge q) \Rightarrow r) \vee (\neg q \wedge r)$$

- p, q, r atomic $\Rightarrow$ WFF
- $p \wedge q$ WFF
- $(p \wedge q) \Rightarrow r$ WFF
- $\neg((p \wedge q) \Rightarrow r)$ WFF
- $\neg q$ WFF
- $(\neg q \wedge r)$ WFF
- Whole formula uses $\vee$ on two WFFs

Conclusion: F is a WFF.

Step 2 — Parsing Tree of F

$$F = \neg((p \wedge q) \Rightarrow r) \vee (\neg q \wedge r)$$



Step 3 — Depth of the Formula

Longest branch:

$$\vee \to \neg \to \Rightarrow \to \wedge \to p$$

- Atomic depth $= 0$
- $\wedge$ above $p, q \to$ depth 1
- $\Rightarrow \to$ depth 2
- $\neg \to$ depth 3
- Root $\vee \to$ depth 4

Depth of $F = 4$

Step 4 — Complexity of the Formula

Count all connectives:

- $\vee \rightarrow 1$
- outer $\neg \rightarrow 1$
- $\Rightarrow \rightarrow 1$
- $\wedge$ (left) $\rightarrow 1$
- $\neg$ (right) $\rightarrow 1$
- $\wedge$ (right) $\rightarrow 1$

Total complexity = 6