Mathematical Logic

Lecture 1: Introduction to Logic

Dr. CHOUDER R

Department of Computer Science

Level: First Year LMD — Licence Computer Science

Academic Year 2025-2026

Course Objectives

By the end of this lecture, students should be able to:

- Explain what logic studies
- Understand why logic is important in CS
- Distinguish informal vs formal reasoning
- Understand syntax vs semantics
- Identify propositions
- Translate informal statements into logical formulas
- Prepare students for propositional and predicate logic

Logic and Reasoning

Logic is often defined as the science of right reasoning.

Key ideas:

- The science of logic begins with a value judgment:

  "What is right reasoning?"

- Logic is closely related to:

$$\text{Logic} \longleftrightarrow \text{Reasoning} \longleftrightarrow \text{Truth}$$

- **Reasoning** is a discourse that establishes the truth of a fact.
- Reasoning is a structured process:
  - a series of propositions or arguments
  - logically linked together
  - leading to a conclusion

Examples of Reasoning

Example 1 — Propositional Reasoning (Modus Ponens)

- **Premises:**
  - If the exam scores are published, then the academic year is over. $(P \rightarrow Q)$
  - The exam scores are published. (P)
- **Conclusion:** Therefore, the academic year is over. (Q)

Example 2 — Predicate Reasoning

- **Premises:**
  - All students are registered. $(\forall x\ Student(x) \rightarrow Registered(x))$
  - Amine is a student. $(Student(Amine))$
- **Conclusion:** Therefore, Amine is registered. $(Registered(Amine))$

Takeaway: Formal logic translates natural language reasoning into precise symbolic formulas, reducing ambiguity.

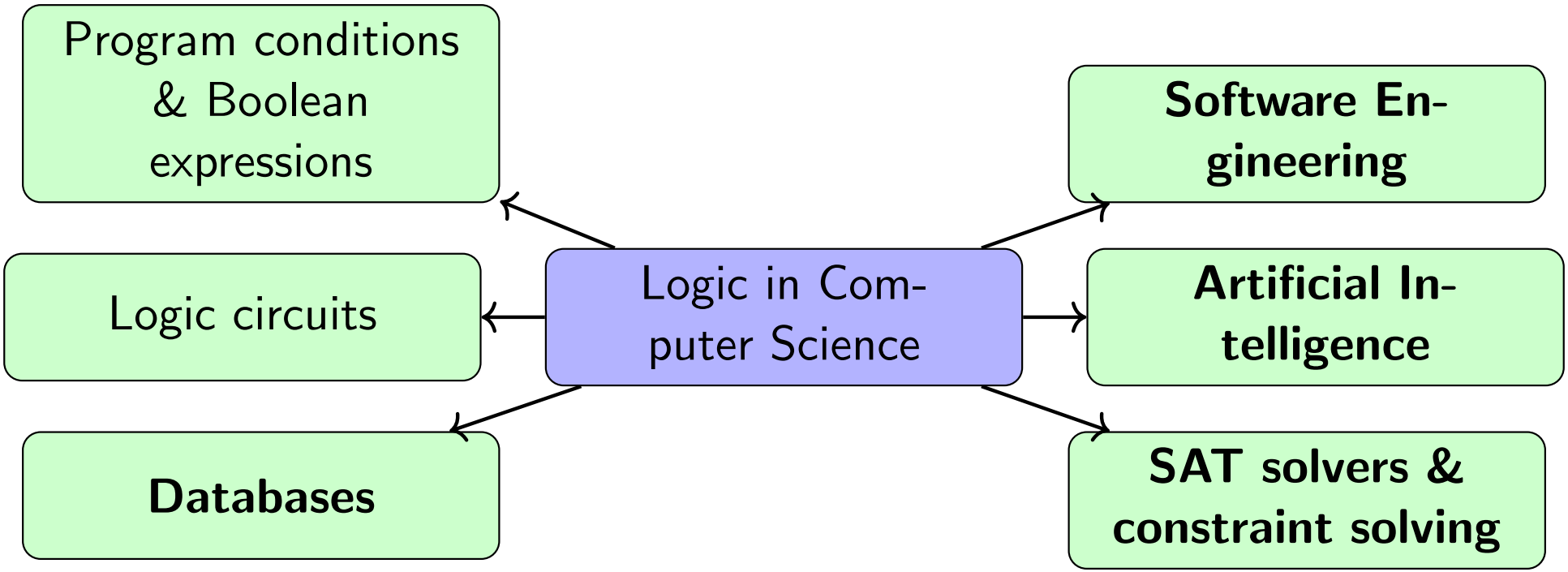
Why Study Logic in Computer Science?

Logic appears everywhere in Computer Science:

- Program conditions and Boolean expressions.
- Logic circuits (based on Boolean algebra).
- **Databases:** query languages.
- **Software Engineering:** modeling tools for developing applications and in the problems of specification and verification of programs.
- **Artificial Intelligence (AI):** logic is used to represent facts and to infer conclusions.

Logic provides the formal foundation for reasoning about programs and systems.

Applications of Logic in Computer Science



Logic Inside a Program — Example

Example code:

Program condition

if (x $>$ 0 AND y $>$ 0) then print("positive")

Logical form:

$$(P \wedge Q) \rightarrow R$$

where:

- P: x $>$ 0
- Q: y $>$ 0
- R: the program prints "positive"

Example evaluation:

- x=3, y=4 $\rightarrow$ $(P \wedge Q)$ = True $\rightarrow$ R executes
- x=-1, y=4 $\rightarrow$ $(P \wedge Q)$ = False $\rightarrow$ R does not execute

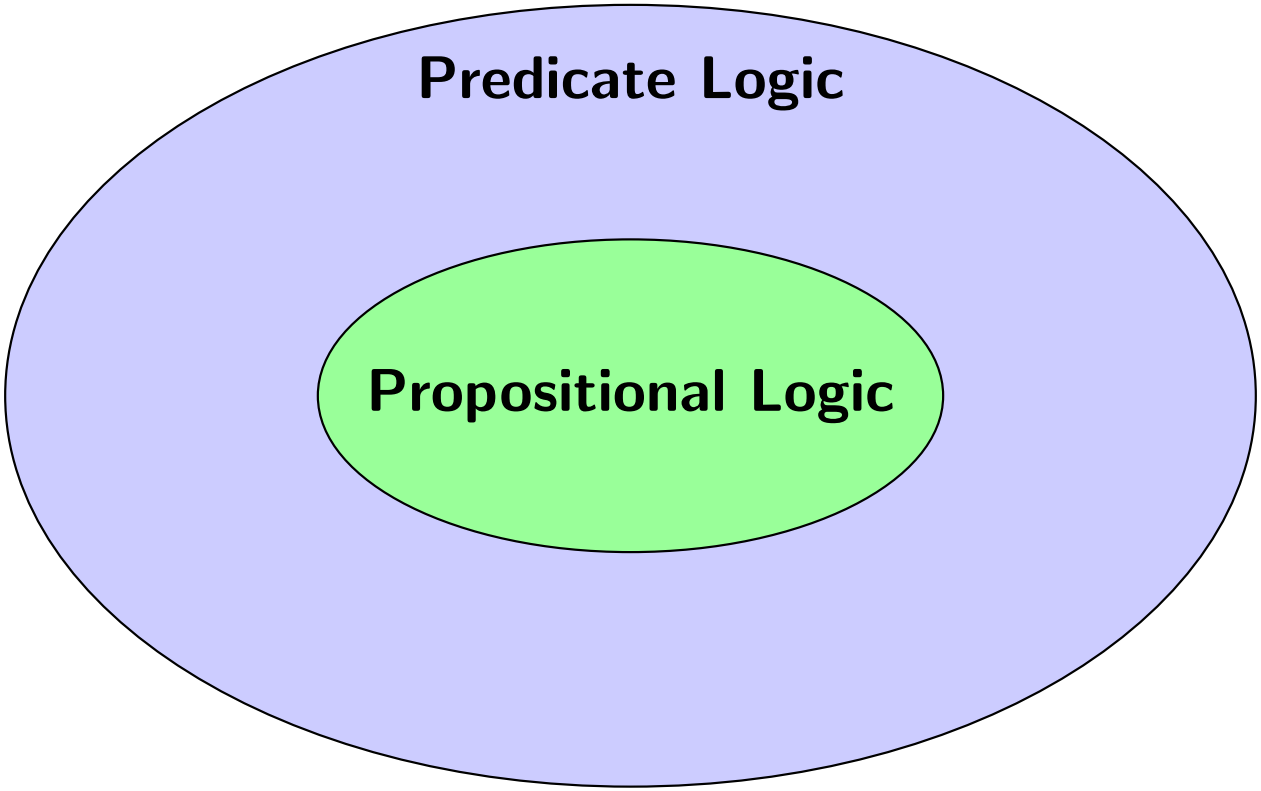
Programs are full of logical formulas.

Main Branches of Logic

- **Propositional logic** — reasoning with propositions and logical connectives
- **Predicate logic** (first-order logic) — reasoning with objects, predicates, and quantifiers

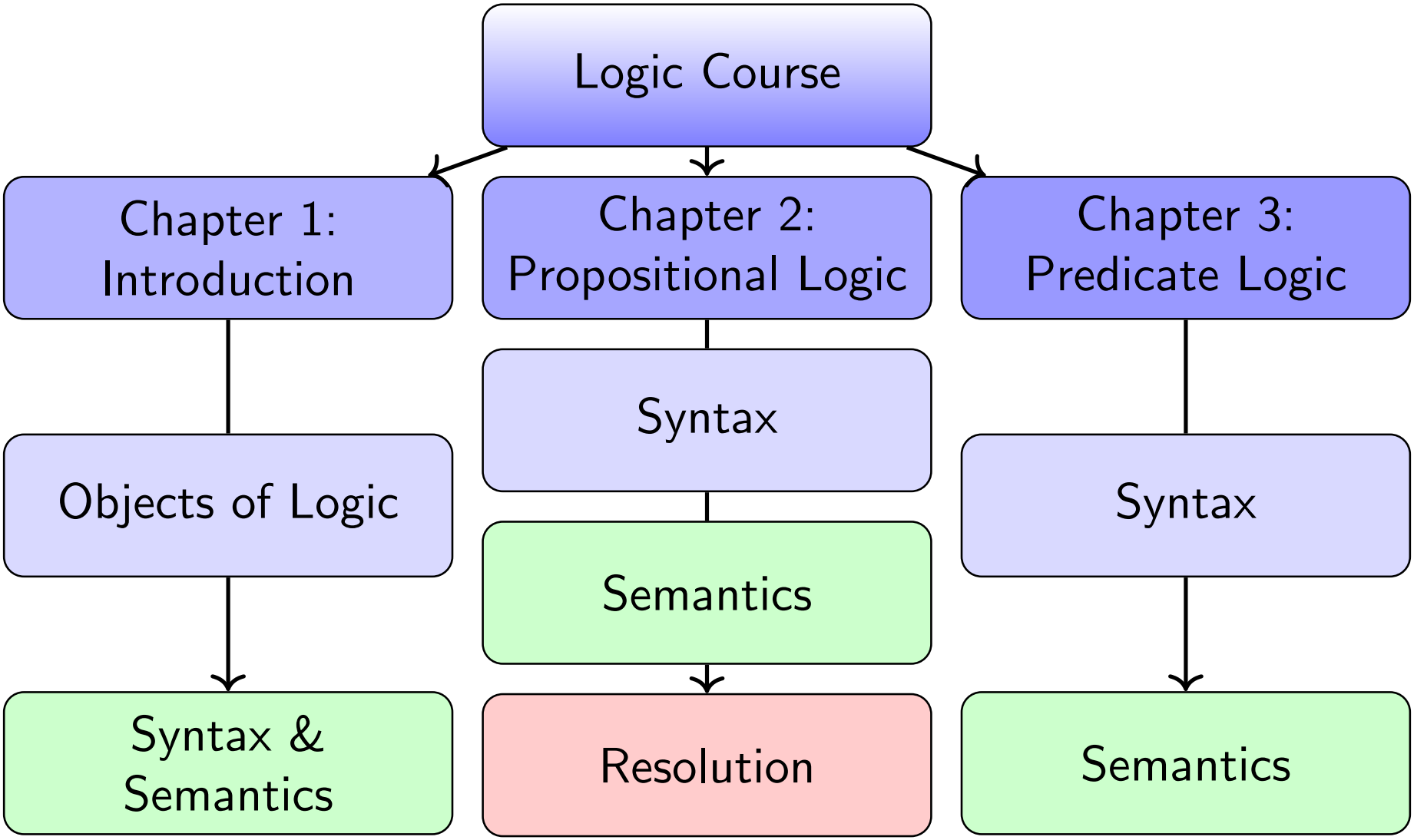
We will start with propositional logic and then extend to predicate logic.

Main Branches of Logic



Note: Propositional logic is a special case of predicate logic where statements do not contain quantifiers or variables.

Logic Course — Roadmap



Informal vs Formal Reasoning

Informal reasoning

- Natural language
- May be ambiguous
- Context dependent

Formal reasoning

- Symbolic language
- Exact rules
- No ambiguity
- Machine-checkable

Language of Logic

The language of logic is a formal (non-natural) language designed to avoid ambiguity present in natural language.

It uses precise logical structures and symbols instead of everyday words.

Lexical ambiguity: a word or expression may have more than one meaning.

Example:

Ambiguous sentence

He looks at someone with brown glasses.

Who has the brown glasses? — He? or the person he looks at?

Example: Ambiguity in Natural Language

Sentence:

Every student read a book

Possible meanings:

- Same book for all students
- Different book for each student

Formal logic removes ambiguity.

From Sentence to Logical Formula

Sentence:

If it rains, then the road is wet

Define:

- P: it rains
- Q: road is wet

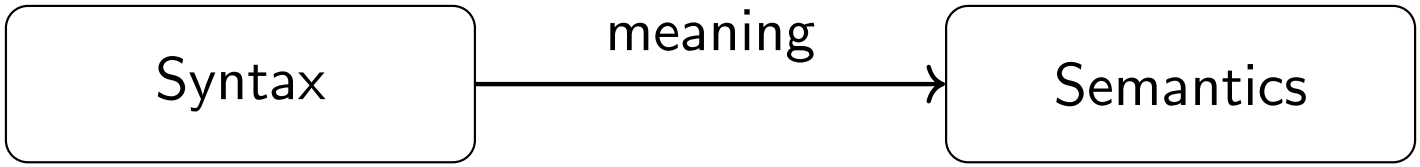
Formula:

$$P \rightarrow Q$$

Logical representation of the statement.

Syntax vs Semantics

To formally and properly define the language of logic, we must distinguish between:



Syntax = structure and form Semantics = interpretation and truth

Example in CS: Boolean expression in a program:

```
if (x > 0 AND y > 0)
```

Syntax question: Is the condition correctly formed?
Semantics question: Is it true for given values of x and y?

Syntax — Intuition from Natural Language

Syntax in English is the arrangement of words in a specific order.

Changing the order may change the meaning.

Example:

- Only Policeman fights crime
- Policeman fights only crime

In logic, syntax defines how formulas must be written.

Semantics — Meaning of Language

Semantics studies the meaning of expressions.

- Linguistic meaning: how words and sentences convey meaning
- Formal semantics: uses logic and mathematics to provide precise frameworks
- Links syntax (structure) to interpretation and truth

Example: the difference between

- "Only policeman fights crime"
- "Policeman fights only crime"

is captured by semantics.

Syntax — Example

Formula:

$$(P \wedge Q) \rightarrow R$$

Syntax question:

- Are parentheses correct?
- Are symbols valid?
- Is it well-formed?

Like grammar in a language.

Semantics — Example

Same formula:

$$(P \wedge Q) \rightarrow R$$

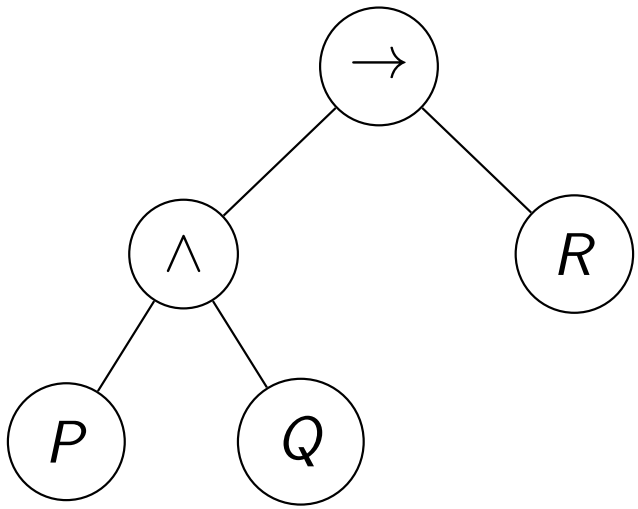
Semantics question:

- Is it TRUE or FALSE?
- Depends on values of P, Q, R

Truth depends on interpretation.

Tree Representation of a Formula

$$(P \wedge Q) \rightarrow R$$



Shows syntactic structure.

Propositions — Definition

A proposition is:

- A declarative statement
- Has a truth value
- True or False (not both)

Examples of Propositions

- $2 + 3 = 5 \rightarrow$ True
- 7 is even $\rightarrow$ False
- Paris is in France $\rightarrow$ True
- $10 < 3 \rightarrow$ False

Not Propositions

- Questions: Is it raining?
- Commands: Close the door
- Open formulas: $x > 2$

Reason: no fixed truth value

Open Sentence vs Proposition

$$x > 3$$

Not a proposition.

If we assign x $= 5 \rightarrow$ True If we assign x $= 1 \rightarrow$ False

Needs a value $\rightarrow$ becomes proposition.

Mini Exercise — Logical Thinking

Decide which are propositions and determine their truth value:

1. 10 is prime
2. $x + 2 = 4$
3. Stop!
4. The server is running

Optional practical exercise: Write a small Boolean condition in a program or analyze a simple logic gate circuit.

Mini Exercise

Task: Decide which statements are propositions and determine their truth value.

Statement	**Proposition?**	**Truth Value / Comment**
10 is prime		
$x + 2 = 4$		
Stop!		
The server is running		

Mini Exercise

Task: Decide which statements are propositions and determine their truth value.

Statement	**Proposition?**	**Truth Value / Comment**
10 is prime	✓ Yes	False
$x + 2 = 4$		
Stop!		
The server is running		

Mini Exercise

Task: Decide which statements are propositions and determine their truth value.

Statement	**Proposition?**	**Truth Value / Comment**
10 is prime	✓ Yes	False
$x + 2 = 4$	✗ No	Depends on x
Stop!		
The server is running		

Mini Exercise

Task: Decide which statements are propositions and determine their truth value.

Statement	**Proposition?**	**Truth Value / Comment**
10 is prime	✓ Yes	False
$x + 2 = 4$	✗ No	Depends on x
Stop!	✗ No	Command, no truth value
The server is running		

Mini Exercise

Task: Decide which statements are propositions and determine their truth value.

Statement	**Proposition?**	**Truth Value / Comment**
10 is prime	✓ Yes	False
$x + 2 = 4$	✗ No	Depends on x
Stop!	✗ No	Command, no truth value
The server is running	✓ Yes	True or False depending on server

Mini Exercise

Practical Exercise: Analyze or write a Boolean condition in a program.

Mini Exercise

Practical Exercise: Analyze or write a Boolean condition in a program.

Example — Program condition:

Python code

```
x = 3; y = 5
if x + y > 7:
    print("Sum is greater than 7")
```

Mini Exercise

Practical Exercise: Analyze or write a Boolean condition in a program.

Example — Program condition:

Python code

```
x = 3; y = 5
if x + y > 7:
    print("Sum is greater than 7")
```

Analysis: Condition $x + y > 7$ is a proposition. Truth value: True.

Summary

In this lecture:

- Motivation for studying logic
- Logic in computer science
- Informal vs formal reasoning
- Syntax vs semantics
- Definition of propositions

Next lecture: **Propositional Logic — Syntax and Connectives**.