Logic of Computer Science
Lecture 3: Natural Deduction in Propositional Logic

Emmanuel Kwesi Tandoh

University of Mines and Technology (UMaT)

June 2025

UNIVERSITY OF MINES AND TECHNOLOGY (UMaT)

Knowledge | Truth | Excellence

Real-World Reasoning Scenarios

- **Detective Work:** "If the window was broken and there are muddy footprints, then the intruder entered through the window." Use evidence to conclude facts.
- **Debugging Code:** "If error X occurs and condition Y holds, then function Z fails." Combine conditions to pinpoint bugs.
- **Everyday Decisions:** "If it's raining and I have no umbrella, then I'll get wet." Simple reasoning guides choices.

Motivation

- Natural deduction mirrors the clear steps you already use in everyday thinking.
- Learning these rules helps you write *precise* arguments in math, programming, and AI.
- Forms the foundation of:
  - **Automated Theorem Proving** (verifying software correctness)
  - **Formal Verification** (checking hardware designs)
  - **Logical AI** (making machines "reason").
- Empowers you to break complex problems into simple, provable steps.

Learning Objectives

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

By the end of this lesson you will be able to:

- Read and write clear natural deduction proofs.
- Use Introduction (I) and Elimination (E) rules for $\rightarrow, \wedge, \vee, \neg$.
- Build proofs for everyday-like scenarios.
- Identify and avoid common proof mistakes.

Proof Structure Notation

Definition

Each proof line has:

- A *statement* (formula).
- A *rule name* (e.g. "$\rightarrow$E").
- References to prior lines used.

Subproofs: Indented blocks for temporary assumptions, later discharged.

Introduction Rules (I)

Build new formulas from known facts:

Connective	Rule	When to use
A, B	$\wedge$I	Combine A and B to infer $A \wedge B$.
A or B	$\vee$I$_1/_2$	From A infer $A \vee B$, or from B.
$A \to B$	$\to$I	Assume A, derive B, conclude $A \to B$.
$\neg A$	$\neg$I	Assume A, derive contradiction $\perp$, infer $\neg A$.

Elimination Rules (E)

Break down complex formulas:

Connective	Rule	When to use
$A \wedge B$	$\wedge E_1/_2$	From $A \wedge B$, get A (or B).
$A \vee B$	$\vee E$	From $A \vee B$ and two cases to derive C.
$A, A \rightarrow B$	$\rightarrow E$	Modus Ponens: infer B.
$A, \neg A$	$\neg E$	Derive $\perp$ (contradiction).
$\perp$	$\perp E$	From $\perp$, infer any formula.

Quick-Check: Which rule extracts A from $A \wedge B$? $\wedge E_1$.

Worked Example: Modus Ponens

Scenario: You know "If you study, you'll pass" and "You studied."

Premises:

- $P \to (Q \to R)$ (If you study P, then if you understand Q, you pass R.)
- P (You studied.)
- Q (You understand.)

Proof:

- $Q \to R$ $\quad \to$E using (1) and (2)
- R $\quad \to$E using (1) and (3)

Proof by Cases ($\vee$E)

Use when: You know "A or B" and want C.

Structure:

- $A \vee B$

2. *Case 1: assume A*
   - ... derive C
3. *Case 2: assume B*
   - ... derive C

- C $\quad \vee$E, using line 1 and results from both cases

Proof by Contradiction ($\neg$I & E)

- Assume the opposite of A, derive a contradiction $\perp$, conclude $\neg A$.
- From $\perp$, you can infer anything (explosion).

Example: Prove $\neg(P \wedge \neg P)$.

1. Assume $P \wedge \neg P$.
2. Extract P and $\neg P$, derive $\perp$.
3. Conclude $\neg(P \wedge \neg P)$ by $\neg$I.

In-Class Practice

1. From $P \wedge (Q \vee R)$ derive $(P \wedge Q) \vee (P \wedge R)$.
2. Prove $(P \to Q) \to (\neg Q \to \neg P)$ using the rules.
3. *Challenge:* From $\neg P \to \neg Q$ and Q, derive P.

Summary & Next Steps

- Natural deduction formalizes clear, step-by-step reasoning.
- Core rules: Intro/Elim for $\rightarrow, \wedge, \vee, \neg$; proof by cases; contradiction.
- **Next:** Soundness/completeness and predicate logic.

References I

- M. Huth and M. Ryan, *Logic in Computer Science: Modelling and Reasoning about Systems*, Cambridge University Press, 2004.
- D. Gries and F. B. Schneider, *A Logical Approach to Discrete Math*, Springer-Verlag, 1993.
- K. H. Rosen, *Discrete Mathematics and Its Applications*, 7th ed., McGraw-Hill, 2011.
- P. Halmos, *Naive Set Theory*, Van Nostrand, 1960.

Thank You!