# **Chapter 1: The Foundations – Logic and Proofs**

This chapter introduces the fundamental concepts of logic, reasoning, and proof techniques, which are essential for **mathematical problem-solving, programming, and algorithm analysis**.

## **1.1 Propositional Logic**

Propositional logic deals with **statements (propositions)** that are either **true (T) or false (F)** but not both.

### **Basic Logical Operators:**

1. **Negation (¬p)**: "Not p" (opposite truth value).
2. **Conjunction (p ∧ q)**: "p AND q" (true only if both p and q are true).
3. **Disjunction (p ∨ q)**: "p OR q" (true if at least one of p or q is true).
4. **Implication (p → q)**: "If p, then q" (false only if p is true and q is false).
5. **Biconditional (p ↔ q)**: "p if and only if q" (true when both have the same truth value).

### **Example: Truth Table for p → q**

| **p** | **q** | **p → q** |
| --- | --- | --- |
| T | T | T |
| T | F | F |
| F | T | T |
| F | F | T |

## **1.2 Applications of Propositional Logic**

Used in **circuit design, programming conditions, and AI decision-making**.

### **Example: Conditional Statements**

* "If it rains, then the ground is wet." → **(rains → wet ground)**
* "If a number is even, then it is divisible by 2." → **(even → divisible by 2)**

## **1.3 Propositional Equivalences**

Logical expressions can be simplified using **logical identities**.

### **Example: De Morgan’s Laws**

* **¬(p ∧ q) ≡ ¬p ∨ ¬q**
* **¬(p ∨ q) ≡ ¬p ∧ ¬q**

## **1.4 Predicates and Quantifiers**

Extends logic with **variables** and **quantifiers**.

* **Universal Quantifier (∀x P(x))**: "For all x, P(x) is true."
* **Existential Quantifier (∃x P(x))**: "There exists an x such that P(x) is true."

### **Example: Universal vs. Existential**

* **∀x (x > 0 → x² > 0)** → "For all x, if x is positive, then x² is positive."
* **∃x (x² = 4)** → "There exists an x such that x² = 4" (x = ±2).

## **1.5 Rules of Inference**

Used to derive conclusions from given premises.

### **Example: Modus Ponens (p → q, p ⊢ q)**

1. "If it is raining, then the ground is wet." (**p → q**)
2. "It is raining." (**p**)
3. ∴ "The ground is wet." (**q**)

## **1.6 Introduction to Proofs**

Mathematical proofs verify the correctness of statements.

### **Types of Proofs:**

* **Direct Proof**: Assume p is true, then show q is true.
* **Proof by Contradiction**: Assume ¬q is true and derive a contradiction.
* **Proof by Induction**: Used for statements about integers.

### **Example: Direct Proof**

**Theorem**: If n is even, then n² is even.  
**Proof**: Let n = 2k (where k is an integer), then n² = (2k)² = 4k² = 2(2k²), which is even. ✅

## **1.7 Proof Methods and Strategy**

Choosing the right proof method based on the problem.

### **Example: Proof by Contradiction**

**Theorem**: √2 is irrational.  
**Proof**: Assume √2 is rational, meaning √2 = a/b (where a, b are integers with no common factors). Squaring both sides:  
**2b² = a²**  
This means a² is even, so a is even → a = 2k. Substituting, we get b is also even.  
This contradicts our assumption that a and b have no common factors. ∴ √2 is irrational. ✅

### **Conclusion**

This chapter builds the foundation for **logical reasoning, proof writing, and mathematical rigor**, which are crucial for **computer science, AI, and algorithm analysis**.

## **Mathematical Symbols**

Here are the **symbols** used in the explanation along with their meanings:

### **Logical Symbols (Propositional Logic)**

| **Symbol** | **Meaning** | **Example** |
| --- | --- | --- |
| **¬p** | Negation ("NOT p") | ¬(x > 5) → "x is not greater than 5" |
| **∧** | Conjunction ("AND") | (p ∧ q) → "p and q are both true" |
| **∨** | Disjunction ("OR") | (p ∨ q) → "p is true or q is true (or both)" |
| **→** | Implication ("If p, then q") | (p → q) → "If p is true, then q must be true" |
| **↔** | Biconditional ("p if and only if q") | (p ↔ q) → "p is true if and only if q is true" |

### **Quantifier Symbols (Predicate Logic)**

| **Symbol** | **Meaning** | **Example** |
| --- | --- | --- |
| **∀** | Universal Quantifier ("For all") | ∀x (x > 0 → x² > 0) → "For all x, if x is positive, then x² is positive" |
| **∃** | Existential Quantifier ("There exists") | ∃x (x² = 4) → "There exists an x such that x² = 4" |

### **Set Theory Symbols**

| **Symbol** | **Meaning** | **Example** |
| --- | --- | --- |
| **∈** | Element of | x ∈ A → "x belongs to set A" |
| **∉** | Not an element of | x ∉ A → "x does not belong to set A" |
| **⊆** | Subset | A ⊆ B → "A is a subset of B" |
| **⊂** | Proper Subset | A ⊂ B → "A is a subset of B but not equal to B" |
| **∩** | Intersection ("AND") | A ∩ B → "Elements common in both A and B" |
| **∪** | Union ("OR") | A ∪ B → "All elements in A or B or both" |
| **∅** | Empty Set | A = ∅ → "Set A has no elements" |

### **Mathematical Symbols**

| **Symbol** | **Meaning** | **Example** |
| --- | --- | --- |
| **≡** | Logical equivalence | ¬(p ∧ q) ≡ (¬p ∨ ¬q) → "De Morgan’s Law" |
| **⊢** | Logical derivation (Inference) | p → q, p ⊢ q → "From p → q and p, we infer q" |
| **√** | Square Root | √4 = 2 |
| **∴** | Therefore | ∴ x = 5 (conclusion of proof) |

**The Foundation (Mathematics): Logic and Proof**

- The rules of logic specify the meaning of mathematical statement

- These rules help us understand and reason with mathematical statement

- Logic is the basis of all mathematical reasoning and of all automated reasoning

- It has practical applications to

- The Design of Computer machine

- The specification of systems

- Artificial Intelligence

- Computer Programming

- Programming Language

- Other area of Computer Science

- Many other field of studies

- To understand mathematics, understand what makes up a correct mathematical arguments, that is, a proof

- Once we prove a mathematical statement is true, we called it a theorem

- A collection of theorems on a topics organize what we know about this topics

- To learn a mathematical topics, a person need to actively construct mathematical arguments on this topic

- Moreover, knowing the proof of a theorem often makes it possible to modify the result to fit new situation

- Proofs are important throughout mathematics and computer science

- In fact, proofs are used to

- Verify that computer programs produce the correct output for all possible input values

- Show that algorithms always produce the correct result

- Establish the security of a system

- Create artificial intelligence

- Furthermore, automated reasoning systems have been created to allow computers to construct their own proofs

**Introduction**

- The rules of logic give precise meaning of mathematical statements

- These rules are used to distinguish between valid and invalid mathematical arguments

1. Proposition – Example 2
2. Definition 1 – Example 2
3. Definition 2 – Example 1
4. Definition 3 – Example 2
5. Definition 4 – Example 2
6. Definition 5 (Conditional Statement) – Example 2
7. Converse, Contrapositive, and Inverse – Example 1
8. Definition 6 – Example 1
9. Implicit use of Bi-conditional
10. Truth Tables of Compound Propositions – Example 1
11. Precedence of Logical Operator
12. Logic and Bit Operation
13. Definition 7 – Example 2
14. Exercises