Logic I

Gudfit

Contents

Chapter 1

Ideas & Motivations

Welcome to Logic (with some theory) by me (Gudfit). The point of these notes is to cover everything I think is important as I build up to my current knowledge, while keeping it free and accessible for everyone from kids to adults.

I aim for each set of notes to be max 55 pages, as rigorous as possible, and far-reaching too.

That means I'll cover the axioms and proofs of the most interesting stuff, plus I'll pull in other subjects we've already touched on to show how math builds on itself like funky Lego. These notes build on my existing informal logic, algebra I, and geometry I notes; they're aimed at keeping the proofs, ideas, and build-up of logic as informal as possible.

It'll be a mix of quick ideas and concepts, but in the appendix for each section, I'll go rigorous with the key axioms pulled from a bunch of books.

Chapter 2

The Language of Reasoning

Our previous discussion of informal logic provided the tools to evaluate the structure of arguments. We now turn to the language in which these arguments are expressed. Mathematics is a language used to encode ideas into sequences of symbols. These symbols represent objects, actions, and concepts. To master mathematics, we must first understand the rules of its language.

2.1 Syntax and Semantics

A language has two fundamental aspects. The meaning behind a particular arrangement of symbols is its semantics, while the grammatical rules governing how symbols may be validly composed is the syntax.

In mathematics, we refer to objects by giving them names. A variable is a symbol that stands in for an object that has not yet been specified. We can assign a name to a particular object with the $:=$ symbol. For example, if we write $x := 5$, we are assigning the value 5 to the variable x. We can also define the golden ratio as $\phi := \frac{1+\sqrt{5}}{2}$. The objects and variables in an expression are its terms. For instance, 5, $x + 2$, and $\sqrt{b^2 - 4ac}$ are all terms.

Remark. On notation. We typically denote variables using single Latin or Greek letters. Common choices include:

- Lowercase: $a, b, c, i, j, k, m, n, p, q, x, y, z$
- Uppercase: $A, B, C, D, M, N, P, Q, R, X, Y, Z$
- Greek: $\alpha, \beta, \gamma, \delta, \epsilon, \theta, \lambda, \mu, \pi, \sigma, \tau, \phi, \omega$

As established in our study of informal logic, arguments are constructed from statements. Mathematical reasoning is concerned exclusively with declarative sentences which express a complete thought that can be judged true or false.

Definition 2.1.1. ***Proposition***. A proposition is a declarative sentence that is unambiguously either true or false. The truth or falsehood of a proposition is its truth value.

Note. The term "proposition" is synonymous with the term "statement" used in the preceding notes on informal logic.

Example 2.1.1. The following are propositions:

1. $2 + 3 = 5$. (True)
2. The integer 7 is even. (False)
3. For any real number x, $x^2 \geq 0$. (True)
4. There exists a prime number greater than 10^{100}. (True)

In contrast, many mathematical expressions are not propositions because they do not possess a truth value. Imperative sentences (commands) and interrogative sentences (questions) are not propositions.

Example 2.1.2. The following are not propositions:

1. $x^2 - 9$. (This is a term; its value depends on x, but it is not a sentence that can be true or false.)
2. Solve the equation $x^2 - 9 = 0$. (An imperative.)
3. Is π a rational number? (An interrogative.)
4. $x > 5$. (This is a sentence, but its truth value depends on the variable x. It is a predicate, not a proposition. We will address such sentences later.)

Definition 2.1.2. ***Atomic Statement.*** A statement is called atomic if it cannot be broken down into smaller statements that still obey the language's syntax.

For example, '5 is a prime number' is an atomic statement. In contrast, '5 is a prime number and 4 is an even number' is a compound statement, built from two atomic components. We will systematically analyse these sentences, extract their logical essence, and use them to build rigorous arguments.

2.2 Truth Values

In the previous section, we defined a proposition as a sentence that is unambiguously either true or false. We call this property the sentence's truth value. The concept of a truth value allows us to abstract away the specific content of a statement and focus solely on its logical status.

Definition 2.2.1. ***Truth Value.*** The truth value of a proposition is its attribute of being true or false. We use the symbol $\top$ (read "top") to denote *true* and $\bot$ (read "bot") to denote *false*.

Example 2.2.1.

- The proposition "Every integer is a rational number" has a truth value of $\top$.
- The proposition "The number π terminates" has a truth value of $\bot$.

The foundation of classical logic, which we will be developing, rests on a crucial assumption about the nature of propositions.

The Principle of Bivalence. A proposition is either true or false, but not both. In other words, every proposition must be assigned exactly one of the truth values $\top$ or $\bot$.

This principle is what underpins the clause "unambiguously either true or false" from our initial definition. It helps us clarify why certain declarative sentences fail to be propositions. For a sentence to qualify as a proposition, it must be possible to assign it a unique truth value. Sentences that prevent this assignment, even if grammatically correct, are excluded from our logical system. Let's examine two such cases.

Semantically Meaningless Sentences. Consider the following sentence:

"The theory of relativity eats breakfast loudly."

This sentence is syntactically perfect according to the rules of English grammar. It has a subject, verb, object, and adverb. However, it is semantically void; it carries no coherent meaning. A scientific theory is an abstract concept and cannot perform a physical action like eating. Because the sentence describes a nonsensical state of affairs, we cannot assess its truth. Is it $\top$? No, that's absurd. Is it $\bot$? If it were, its negation, "The theory of relativity does not eat breakfast loudly," would have to be true, which is equally nonsensical. Since no truth value can be assigned, it is not a proposition.

Self-Contradictory Sentences (Paradoxes). A more subtle challenge arises from sentences that are semantically meaningful but inherently self-contradictory. Consider a sentence written as the sole entry on a piece of paper:

"The only proposition on this page is false."

Let's attempt to assign a truth value to this sentence, which we will call P.

1. **Assume P is true ($\top$).** If P is true, then what it asserts must be correct. The sentence asserts that "The only proposition on this page is false." Since P is the only proposition on the page, this means P must be false. This contradicts our assumption that P is true.
2. **Assume P is false ($\bot$).** If P is false, then what it asserts must be incorrect. The assertion is that "The only proposition on this page is false." The negation of this would be that the proposition is, in fact, true. This means P must be true. This again contradicts our assumption.

In both cases, we arrive at a contradiction. The sentence cannot be $\top$ and it cannot be $\bot$. It violates the Principle of Bivalence and therefore cannot be a proposition.

Therefore, for a declarative sentence to be a proposition in our system, it must not only be syntactically well-formed but also semantically meaningful and free from the kind of self-contradiction that makes assigning a truth value impossible.

Logical Connectives

The propositions we have seen so far have been atomic, meaning they represent a single, indivisible idea. However, the power of logic comes from our ability to form complex arguments by combining these atomic statements. These new, larger statements are called compound statements, and they are formed using logical connectives.

Each connective is a rule that determines the truth value of a compound statement based solely on the truth values of its constituent parts. We can precisely define these rules using a truth table, which exhaustively lists the output truth value for every possible combination of input truth values.

Negation. The simplest way to modify a proposition is to deny it. Consider the statement:

"The integer 7 is an even number." ($\bot$)

Its negation is:

"The integer 7 is not an even number." ($\top$)

Negation is a unary connective, meaning it operates on a single proposition. It inverts the truth value of its input. We denote negation with the symbol $\neg$.

p	$\neg p$
$\top$	$\bot$
$\bot$	$\top$

Table 2.1: Truth table for negation.

Conjunction and Disjunction. We can also combine two or more propositions. Consider this compound statement:

"A square has four sides, and a triangle has three vertices."

This statement, formed with "and," is true because both of its atomic parts are true. If either part were false, our intuition tells us the entire statement would be false. This "and" connective is called a logical conjunction, denoted by the symbol $\wedge$. A conjunction $p \wedge q$ is true if and only if both p and q are true.

In English, conjunction can be expressed in many ways besides "and." Words like "but," "yet," "while," and "moreover" often serve the same logical function. For example, "A square has four sides, but a triangle has three vertices" is logically equivalent to the sentence above.

The logical dual to conjunction is disjunction, which corresponds to an "inclusive or." Consider:

"A pentagon has five sides, or a circle has four corners."

This statement is true because at least one of its parts ("A pentagon has five sides") is true. A disjunction, denoted by the symbol $\vee$, is false only when both of its components are false.

p	q	$p \wedge q$	$p \vee q$
$\top$	$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$	$\top$
$\bot$	$\top$	$\bot$	$\top$
$\bot$	$\bot$	$\bot$	$\bot$

Table 2.2: Truth tables for conjunction ($\wedge$) and disjunction ($\vee$).

Logical Duality. The relationship between conjunction and disjunction is an example of a deeper principle known as duality. Notice how their columns in the truth table seem to be inverted mirrors of each other.

Definition 2.2.2. ***Logical Duality.*** Two binary logical connectives, f and g, are said to be **logically dual** if the negation of the output of one yields the same truth value as applying the other to the negated inputs. Formally, f and g are duals if the truth table output for $\neg f(p, q)$ is identical to the output for $g(\neg p, \neg q)$.

Conjunction and disjunction are a dual pair. While we will not prove it here, it can be shown that negating a conjunction is equivalent to the disjunction of the negations, and vice versa. This powerful concept is a cornerstone of Boolean algebra, which we will explore in detail later.

Conditional Statements. Perhaps the most important connective for mathematical reasoning is the implication or conditional statement. This allows us to express a dependency between two propositions.

Consider the statement:

"If the employee finishes the project by Friday, then they will receive a bonus."

This sentence establishes a relationship where the second part (receiving a bonus) must occur whenever the first part (finishing the project) is satisfied. We call this connective the material implication, denoted by the arrow $\rightarrow$. In a statement $p \rightarrow q$:

- p is called the *antecedent* (or hypothesis/premise).
- q is called the *consequent* (or conclusion).

The logic of the implication $\top \rightarrow \bot$ is often counter-intuitive to beginners because it is only false in one specific scenario: when the promise is broken.

1. **Case 1 (p is $\top$, q is $\top$):** The employee finishes on Friday, and gets the bonus. The statement was true.
2. **Case 2 (p is $\top$, q is $\bot$):** The employee finishes on Friday, but does *not* get the bonus. The contract was violated. The statement is false.
3. **Case 3 (p is $\bot$, q is $\top$):** The employee does not finish on Friday, but receives a bonus anyway (perhaps for other good work). The statement "If you finish... you get a bonus" was not a lie; it simply didn't specify what happens if you *don't* finish. The statement remains true.
4. **Case 4 (p is $\bot$, q is $\bot$):** The employee does not finish, and does not get the bonus. The statement holds true.

Cases 3 and 4 illustrate a concept called *vacuous truth.* If the antecedent is false, the implication is automatically considered true, regardless of the consequent.

Necessary and Sufficient Conditions. In English, implications are often phrased without using the standard "if... then..." structure. Understanding these variations is vital for translating natural language into logic. Consider the implication $p \to q$, where p is "It is raining" and q is "The ground is wet."

Phrasing	**Logic Interpretation**
"If it rains, the ground is wet."	Standard form.
"The ground is wet if it rains."	q if p.
"It raining is sufficient for the ground to be wet."	p is *sufficient* for q.
"The ground being wet is necessary for it to rain."	q is *necessary* for p.[1]
"It rains only if the ground is wet."	p only if q.

Table 2.3: Common English forms of the conditional $p \to q$.

Biconditional Statements. Finally, we have the *material equivalence*, also known as the *biconditional*. This describes a relationship where two propositions always share the same truth value—either both are true, or both are false. We denote this with the double arrow $\leftrightarrow$. The statement $p \leftrightarrow q$ is read as "p if and only if q" (often abbreviated as iff).

Example 2.2.2. "A polygon is a triangle (p) if and only if it has exactly three sides (q)."

If p is true, q must be true. If p is false (the polygon is a square), q must be false (it does not have three sides).

The truth tables for these connectives are summarized below.

p	q	$p \to q$	$p \leftrightarrow q$
$\top$	$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$	$\bot$
$\bot$	$\top$	$\top$	$\bot$
$\bot$	$\bot$	$\top$	$\top$

Table 2.4: Truth tables for conditional ($\to$) and biconditional ($\leftrightarrow$) statements.

With these connectives defined, we have the primitive tools required to construct complex logical structures. However, to do so rigorously, we need a formal method for building these structures.

2.3 A Primer on Recursion

Before we can provide a perfectly rigorous definition for a proposition, we must first introduce a foundational concept that lies at the heart of logic and computer science: *recursion*. In essence, recursion is a method of solving a problem by defining it in terms of a smaller version of itself.

To understand this, let's consider a familiar mathematical function: the factorial. The factorial of a non-negative integer n, denoted $n!$, is the product of all positive integers up to n. For example, we know that $4! = 4 \times 3 \times 2 \times 1 = 24$. But how would we describe the process of calculating it?

[1] This phrasing is subtle. It means that if the ground is *not* wet, it cannot possibly be raining (via the contrapositive, which we will study later).

We can say that to calculate 4!, we need to multiply 4 by the result of $3 \times 2 \times 1$. But notice that $3 \times 2 \times 1$ is just 3!. So, we can restate the problem:

$$4! = 4 \times 3!$$

This is a recursive step. We have defined the problem of calculating 4! in terms of a "smaller" instance of the exact same problem: calculating 3!. Of course, this leads to the question, what is 3!? Following the same logic, $3! = 3 \times 2!$, and so on.

This process can't continue forever. If we just keep breaking down the problem into smaller subproblems, we will never arrive at an answer. We need a stopping point—a version of the problem so simple that we can state its answer directly. For factorials, we define this stopping point as $1! = 1$. This is our anchor.

Thus the two key components of any recursive definition includes:

1. **The Base Case:** An explicit solution for the simplest possible version(s) of the problem. This provides the necessary stopping condition. For factorial, the base case is $1! = 1$.
2. **The Recursive Step:** A rule that defines a more complex instance of the problem in terms of a simpler instance. For factorial, this is the rule $n! = n \times (n-1)!$ for $n > 1$.

Combining these, we can write a complete, formal definition for the factorial function:

$$n! = \begin{cases} 1 & \text{if } n = 1 \\ n \times (n-1)! & \text{if } n > 1 \end{cases}$$

This definition encapsulates the entire process. To find 4!, the rule sends us to 3!, then to 2!, until we finally hit the base case at 1!. Once we have that concrete answer, we can work our way back up: $2 \times 1! = 2$, then $3 \times 2 = 6$, and finally $4 \times 6 = 24$.

The Structure of Propositions: A Recursive Definition

Using recursion lets us build a precise, syntactic definition of what constitutes a well-formed proposition.

Definition 2.3.1. ***Proposition (Formal Syntactic Definition)***. A string of symbols λ is a proposition (or well-formed formula, WFF) if and only if it can be constructed by applying the following rules:

1. **Base Cases:** Any atomic statement variable $(p, q, r, \ldots)$ is a proposition. The constants $\top$ and $\bot$ are also propositions.
2. **Recursive Step:** If ϕ and ψ are propositions, then the following strings, with parentheses included, are also propositions:
   - $(\neg\phi)$
   - $(\phi \wedge \psi)$
   - $(\phi \vee \psi)$
   - $(\phi \rightarrow \psi)$

- $(\phi \leftrightarrow \psi)$

A string of symbols is a proposition only if it can be generated by a finite number of applications of these rules.

Note. The use of parentheses is crucial for avoiding ambiguity. For example, $(p \land (q \lor r))$ is structurally different from $((p \land q) \lor r)$.

p	q	r	$q \lor r$	$p \land (q \lor r)$	$p \land q$	$(p \land q) \lor r$
$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$
$\top$	$\top$	$\bot$	$\top$	$\top$	$\top$	$\top$
$\top$	$\bot$	$\top$	$\top$	$\top$	$\bot$	$\top$
$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$\bot$	$\top$	$\top$	$\top$	$\bot$	$\bot$	$\top$
$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\bot$	$\top$
$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\bot$	$\top$
$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$

Table 2.5: Truth table comparing $(p \land (q \lor r))$ and $((p \land q) \lor r)$.

For readability, we often drop the outermost parentheses when the meaning is clear.

This definition provides a mechanical procedure to verify if a statement is well-formed. There are two complementary ways to think about this procedure.

Recursive Decomposition. We can verify a formula by treating it as a complex object and breaking it down until we reach the base cases. This is a top-down approach. If the entire formula can be deconstructed into valid components according to the rules, it is a well-formed proposition. This process generates a parse tree.

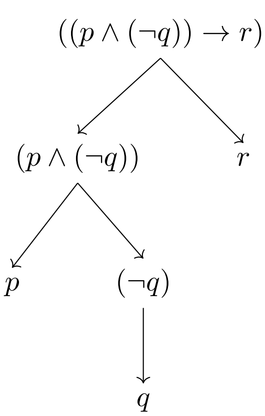

Figure 2.1: A parse tree showing the recursive decomposition of $((p \land (\neg q)) \to r)$. Arrows indicate decomposition from complex to atomic.

Inductive Construction. Alternatively, we can think from the bottom-up. We begin with our base cases (the atomic propositions) and use the recursive rules to build larger and larger formulas.

This "bootstrapping" process is analogous to how we calculated the factorial: starting with a known value and building up. If we can demonstrate a sequence of steps that constructs the target formula, we have proven it is well-formed.

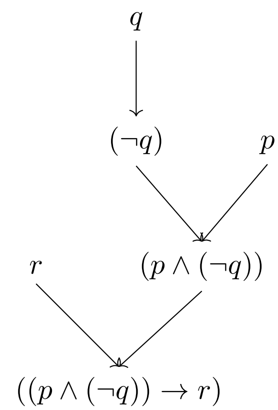

Figure 2.2: A build tree showing the inductive construction of $((p \wedge (\neg q)) \to r)$. Arrows indicate construction from atomic to complex.

Definition 2.3.2. *Propositional Formula*. A propositional formula is an expression built according to the recursive rules for propositions, which contains at least one propositional variable. It becomes a proposition once every variable within it is substituted with a specific proposition (like $\top$, $\bot$, or "2 is even").

For instance, $p \to q$ is a propositional formula. If we substitute p with "$\pi > 3$" ($\top$) and q with "Earth is flat" ($\bot$), the formula becomes the proposition "$\pi > 3 \to$ Earth is flat", which has a truth value of $\bot$.

Logical Equivalence

Now that we have a formal method for constructing propositions, a natural question arises: can two propositional formulas that are written differently still mean the same thing? The answer is yes.

Definition 2.3.3. *Logical Equivalence*. Two propositional formulas, ϕ and ψ, are logically equivalent if they have the exact same truth value for every possible assignment of truth values to their variables. We denote this relationship with the symbol $\equiv$, writing $\phi \equiv \psi$.

The most direct way to verify logical equivalence is to construct a single truth table for both formulas and compare their final output columns. If the columns are identical, the formulas are equivalent.

Consider the statement $p \to q$. It turns out this is logically equivalent to $(\neg p) \vee q$. While they appear different, their underlying logical meaning is the same. Let's verify this with a truth table.

The Cost of Verification. While definitive, the truth table method has a significant practical weakness: it does not scale well. Consider a formula with n distinct propositional variables.

- For $n = 1$ variable (p), we need $2^1 = 2$ rows.
- For $n = 2$ variables (p, q), we need $2^2 = 4$ rows.

p	q	$p \to q$	$\neg p$	$(\neg p) \lor q$
$\top$	$\top$	$\top$	$\bot$	$\top$
$\top$	$\bot$	$\bot$	$\bot$	$\bot$
$\bot$	$\top$	$\top$	$\top$	$\top$
$\bot$	$\bot$	$\top$	$\top$	$\top$

Table 2.6: Truth table demonstrating that $p \to q \equiv (\neg p) \lor q$. The identical final columns (in blue) confirm their equivalence.

- For $n = 3$ variables (p, q, r), we need $2^3 = 8$ rows.

Each new variable we add doubles the number of rows required to check every possibility. If we let $R(n)$ be the number of rows for n variables, we can express this relationship with a recurrence relation:

$$R(n) = \begin{cases} 1 & \text{if } n = 0 \\ 2 \cdot R(n-1) & \text{if } n \geq 1 \end{cases}$$

This shows that verifying an equivalence for a formula with n variables requires constructing a truth table with $R(n) = 2^n$ rows. This "combinatorial explosion" makes the method computationally infeasible for formulas with even a moderate number of variables. This limitation motivates the need for an alternative method: formal proof, a central topic in later chapters (see §chapter 4).

Logical Nonequivalence

Proving that two formulas are *not* equivalent is a much simpler task. To establish equivalence, we must show that the formulas match in *every* possible case. To show nonequivalence, we only need to find a *single* case where they differ.

Definition 2.3.4. ***Logical Nonequivalence***. Two propositional formulas, ϕ and ψ, are not logically equivalent if there exists at least one assignment of truth values to their variables for which their resulting truth values are different. This one case is called a *counterexample*. We denote this relationship by $\phi \not\equiv \psi$.

A common misconception is that an implication $(p \to q)$ is equivalent to its converse $(q \to p)$. We can disprove this by demonstrating a single counterexample. The truth table below reveals all possible outcomes.

p	q	$p \to q$	$q \to p$
$\top$	$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$	$\top$
$\bot$	$\top$	$\top$	$\bot$
$\bot$	$\bot$	$\top$	$\top$

Table 2.7: Truth table for an implication and its converse.

The highlighted row provides our counterexample. When p is true and q is false:

- The implication $p \to q$ evaluates to $\bot$.
- Its converse $q \to p$ evaluates to $\top$.

Since we have found an assignment where their truth values differ ($\bot \neq \top$), this single case is sufficient proof that $p \to q \not\equiv q \to p$.

2.4 Exercises

1. **Classification.** For each of the following sentences, determine if it is a proposition, a term, or neither. If it is a proposition, state its truth value ($\top$ or $\bot$). If it is neither, briefly explain why.
   (a) The sum of the angles in any triangle is 180 degrees.
   (b) $(x + y)^2$.
   (c) Will you pass this course?
   (d) There exists a prime number greater than 100.
   (e) This statement is false.
   (f) Add 5 to both sides of the equation.

2. **Translation.** Let p be the proposition "It is raining," let q be "The wind is blowing," and let r be "The ground is wet." Translate the following English sentences into propositional formulae using these variables and the logical connectives $\neg, \wedge, \vee, \to, \leftrightarrow$.
   (a) It is not raining, but the wind is blowing.
   (b) If it is raining, then the ground is wet.
   (c) The ground is wet only if it is raining.
   (d) For the ground to be wet, it is sufficient that it is raining.
   (e) Either it is raining or the wind is blowing, but not both.
   (f) The ground is wet if and only if it is raining.

3. **Tautologies.** A proposition that is true for every possible assignment of truth values to its variables is called a *tautology*. A proposition that is always false is a *contradiction*. By constructing a truth table, determine whether the following formula is a tautology, a contradiction, or neither.
$$((p \to q) \wedge \neg q) \to \neg p$$

4. **De Morgan's Laws.** In the text, a brief mention was made of the duality between conjunction and disjunction. This relationship is formalised by De Morgan's Laws.
   (a) Use a truth table to prove the logical equivalence $\neg(p \wedge q) \equiv (\neg p \vee \neg q)$.
   (b) Consider the English sentence: "It is not the case that the test is both easy and long." Using part (a), write an equivalent sentence that does not use the word "both".

5. **The Contrapositive.** For an implication $p \to q$, we can define three related conditional statements:
   - The converse: $q \to p$.
   - The inverse: $\neg p \to \neg q$.
   - The contrapositive: $\neg q \to \neg p$.

   Use truth tables to prove that an implication is logically equivalent to its contrapositive, but is not logically equivalent to its converse or its inverse.

6. **Well-Formed Formulae.** According to the formal syntactic definition of a proposition, determine which of the following strings are well-formed formulae (WFFs). For any that are not, explain which rule is violated. For the first valid WFF you identify, draw a parse tree showing its recursive decomposition.

(a) $(p \wedge q \vee r)$
(b) $(\neg(p \rightarrow q))$
(c) $(p \leftrightarrow (q \rightarrow (r \wedge (\neg s))))$
(d) $(p \wedge \rightarrow q)$

7. **A Logic Puzzle.** You are on an island inhabited by two tribes: the Knights, who always tell the truth, and the Knaves, who always lie. You meet two inhabitants, A and B. A says, "If I am a Knight, then B is a Knight." Can you determine what A and B are?

   Remark. Let p be the proposition "A is a Knight" and q be "B is a Knight". The statement made by A must be consistent with their identity. That is, the biconditional $p \leftrightarrow (p \rightarrow q)$ must be true.

8. **Necessary and Sufficient Conditions.** Translate the following statements into the form $p \rightarrow q$. Clearly define what propositions your variables p and q represent.
   (a) A necessary condition for a number to be divisible by 6 is that it is divisible by 3.
   (b) Being over 18 is a sufficient condition for being eligible to vote.
   (c) You can access the network only if you have a password.

9. **The Biconditional.** The text defines the biconditional $p \leftrightarrow q$ as being true when p and q have the same truth value. Prove, using a truth table, that this is logically equivalent to the conjunction of an implication and its converse. That is, prove:

$$(p \leftrightarrow q) \equiv ((p \rightarrow q) \wedge (q \rightarrow p))$$

10. ⋆ **Functional Completeness.** A set of logical connectives is called *functionally complete* if all other connectives can be expressed using only connectives from that set. Consider the "NAND" connective, denoted by the Sheffer stroke |, with the following truth table:

p	q	$p \vert q$
$\top$	$\top$	$\bot$
$\top$	$\bot$	$\top$
$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\top$

   Notice that $p|q$ is equivalent to $\neg(p \wedge q)$. Show that the set $\{|\}$ is functionally complete by finding formulae, using only the NAND connective, that are logically equivalent to:
   (a) $\neg p$
   (b) $p \wedge q$
   (c) $p \vee q$

   Remark. For part (a), consider what happens when both inputs to NAND are the same, i.e., $p|p$. For part (b), remember that double negation, $\neg(\neg\phi)$, is equivalent to ϕ.

11. ⋆ **Infinite Sequence Paradox.** Suppose we have an infinite sequence of sentences $S_0, S_1, S_2, \ldots, S_i, \ldots$ where each sentence asserts that every sentence following it is false. In this definition, i ranges over all of the natural numbers 0, 1, 2, ... That is, $S_i :=$ "S_j is false for all $j > i$." What are the truth values of the sentences in this sequence?

12. ⋆ **Finitude of Beliefs.** Determine the truth value of the following sentence. "You have finitely many beliefs."

   Remark. The "You" above refers to the reader of this notes.

Chapter 3

Axiomatic Propositional Logic

While the method of truth tables provides a definitive way to verify logical equivalence, it is computationally expensive and offers limited insight into the structural reasons for an equivalence. The core of mathematical advancement lies not in exhaustive verification, but in deduction: the process of deriving new truths from established ones. To formalise this process, we move from the semantic approach of truth tables to a syntactic, axiomatic system.

3.1 The Structure of Proof Systems

At a general level, logic provides a framework for expressing mathematical statements and verifying proofs. While we focus here on classical logic, there exist numerous other systems (such as temporal, modal, and intuitionistic logic), designed to reason about time, knowledge, or constructive existence. Regardless of the specific domain, any formal treatment of mathematics relies on the concept of a *proof system*.

A proof system is composed of two distinct classes of objects: mathematical statements and proofs. Both are represented as finite strings of symbols over a fixed alphabet. The relationship between them is governed by two fundamental functions:

1. **Semantics (Truth):** A rule determining whether a given statement is true or false. In our previous discussion, this was achieved via truth tables.
2. **Verification (Syntax):** A mechanical procedure that decides if a specific string is a valid proof for a given statement. Crucially, this verification must be efficiently computable; a proof is useless if its validity cannot be checked in a reasonable amount of time.

We distinguish between the existence of a proof and the truth of a statement using two key properties. A system is sound if no false statement has a valid proof; that is, the verification procedure never accepts a justification for an untruth. Conversely, a system is complete if every true statement possesses a valid proof.

Example 3.1.1. Primality. Consider the statement type "The integer n is composite" (not prime).

- **Semantics:** The statement is true if n has a divisor d such that $1 < d < n$.
- **Proof:** A valid proof string could be the binary representation of such a divisor d.
- **Verification:** To verify the proof, one simply divides n by d and checks if the remainder is zero. This is computationally efficient, even if finding the divisor d initially is difficult.

3.2 Axioms and Proofs

To construct a proof system for propositional logic, we begin with a set of fundamental statements, called axioms, which as noted in my previous notes are statements that are accepted as true without proof. These axioms serve as the bedrock upon which all subsequent logical deductions are built. The axioms of classical propositional logic specify the behaviour of the logical connectives.

The axioms presented in Table 3.1 encode our foundational assumptions. Each is a logical equivalence, establishing that the two expressions are interchangeable in all contexts. The first five axiom pairs define a structure known as a Boolean algebra.

Table 3.1: The axioms of classical logic.

Axiom	**Conjunctive Form**	**Disjunctive Form**
Identity	$\top \land p \equiv p$	$\bot \lor p \equiv p$
Complement	$\neg p \land p \equiv \bot$	$\neg p \lor p \equiv \top$
Commutativity	$p \land q \equiv q \land p$	$p \lor q \equiv q \lor p$
Associativity	$p \land (q \land r) \equiv (p \land q) \land r$	$p \lor (q \lor r) \equiv (p \lor q) \lor r$
Distributivity	$p \land (q \lor r) \equiv (p \land q) \lor (p \land r)$	$p \lor (q \land r) \equiv (p \lor q) \land (p \lor r)$
	Disintegration Rules	
Conditional	$p \to q \equiv \neg p \lor q$	
Biconditional	$p \leftrightarrow q \equiv (p \to q) \land (q \to p)$	

From these axioms, we can prove other statements, which we call theorems.

Definition 3.2.1. ***Theorem*.** A theorem is a proposition that has been proven to be true based on the axioms and previously proven theorems.

A proof is a sequence of logical statements, each justified by an axiom, a definition, or a previously established theorem, that culminates in the statement of the theorem being proven. The complement axiom in Table 3.1 states that $\neg p$ behaves in a specific way with respect to p. The following theorem establishes that this behaviour is unique to $\neg p$.

***Theorem* 3.2.1. Uniqueness of Complements.** For any propositions p and q, if $p \land q \equiv \bot$ and $p \lor q \equiv \top$, then $\neg p \equiv q$.

Proof. Let p and q be arbitrary propositions. Assume that $p \land q \equiv \bot$ and $p \lor q \equiv \top$. We will prove $\neg p \equiv q$ by showing that both $\neg p$ and q are equivalent to the same expression. First, we establish an equivalence for $\neg p$.

$$
\begin{aligned}
\neg p &\equiv \top \land \neg p && \text{by Identity} \\
&\equiv (\neg p) \land \top && \text{by Commutativity} \\
&\equiv (\neg p) \land (p \lor q) && \text{by assumption, } p \lor q \equiv \top \\
&\equiv (\neg p \land p) \lor (\neg p \land q) && \text{by Distributivity} \\
&\equiv \bot \lor (\neg p \land q) && \text{by Complement} \\
&\equiv \neg p \land q && \text{by Identity}
\end{aligned}
$$

Thus, $\neg p \equiv \neg p \wedge q$. Similarly, we establish an equivalence for q.

$$\begin{aligned}
q &\equiv \top \wedge q && \text{by Identity} \\
&\equiv q \wedge \top && \text{by Commutativity} \\
&\equiv q \wedge (p \vee \neg p) && \text{by Complement} \\
&\equiv (q \wedge p) \vee (q \wedge \neg p) && \text{by Distributivity} \\
&\equiv (p \wedge q) \vee (\neg p \wedge q) && \text{by Commutativity} \\
&\equiv \bot \vee (\neg p \wedge q) && \text{by assumption, } p \wedge q \equiv \bot \\
&\equiv \neg p \wedge q && \text{by Identity}
\end{aligned}$$

This gives us $q \equiv \neg p \wedge q$. As both $\neg p$ and q are equivalent to $\neg p \wedge q$, they are equivalent to each other: $\neg p \equiv q$. ■

Remark. The symbol ■ at the end of a proof is a modern substitute for the traditional *Q.E.D.*, an initialism for the Latin phrase *quod erat demonstrandum*, meaning "what was to be shown."

A proof provides more than a formal verification; it reveals why a statement is a necessary consequence of the axiomatic system. From Theorem 3.2.1, we can derive a simple but important corollary.

***Corollary* 3.2.1.** $\top \equiv \neg\bot$ and $\bot \equiv \neg\top$.

Proof. By the Identity axiom, we have $\bot \wedge \top \equiv \bot$. Similarly, by Commutativity and Identity, we have $\bot \vee \top \equiv \top \vee \bot \equiv \top$. We have satisfied the premises of Theorem 3.2.1 with $p := \bot$ and $q := \top$. Therefore, we conclude $\top \equiv \neg\bot$.

The second part, $\bot \equiv \neg\top$, follows from an identical argument with $p := \top$ and $q := \bot$. ■

***Corollary* 3.2.2.** *Negative Equivalence.* For any propositions p and q, if $p \equiv q$, then $\neg p \equiv \neg q$.

Proof. Let p and q be propositions such that $p \equiv q$. Observe the following chain of equivalences.

$$\begin{aligned}
q \wedge \neg p &\equiv p \wedge \neg p && \text{by assumption, } p \equiv q \\
&\equiv \bot && \text{by Complement}
\end{aligned}$$

And similarly for the disjunctive case:

$$\begin{aligned}
q \vee \neg p &\equiv p \vee \neg p && \text{by assumption, } p \equiv q \\
&\equiv \top && \text{by Complement}
\end{aligned}$$

Having satisfied the premises of Theorem 3.2.1 (with p in the theorem being our q, and q in the theorem being our $\neg p$), we conclude that $\neg q \equiv \neg p$. ■

***Corollary* 3.2.3.** For any propositions p, q, r, s such that $p \equiv q$ and $r \equiv s$, the following hold:

1. $p \wedge r \equiv q \wedge s$
2. $p \vee r \equiv q \vee s$
3. $p \to r \equiv q \to s$
4. $p \leftrightarrow r \equiv q \leftrightarrow s$

Remark. The proofs for corollary 3.2.3 are straightforward applications of substitution and are omitted.

We now have the tools to prove several fundamental theorems of propositional logic.

***Theorem* 3.2.2. Double Negation.** For any proposition p, we have $p \equiv \neg\neg p$.

Proof. Let p be a proposition. By the Commutativity and Complement axioms, we have $\neg p \wedge p \equiv p \wedge \neg p \equiv \bot$. Similarly, $\neg p \vee p \equiv p \vee \neg p \equiv \top$. The premises of Theorem 3.2.1 are met with p from the theorem being our $\neg p$, and q from the theorem being our p. Therefore, $p \equiv \neg(\neg p)$. ∎

***Theorem* 3.2.3. Idempotence.** For any proposition p, we have $p \wedge p \equiv p$ and $p \vee p \equiv p$.

Proof. Let p be a proposition. For the conjunctive statement:

$$\begin{aligned} p \wedge p &\equiv (p \wedge p) \vee \bot && \text{by Identity} \\ &\equiv (p \wedge p) \vee (p \wedge \neg p) && \text{by Complement} \\ &\equiv p \wedge (p \vee \neg p) && \text{by Distributivity} \\ &\equiv p \wedge \top && \text{by Complement} \\ &\equiv p && \text{by Identity} \end{aligned}$$

An analogous chain of reasoning establishes the disjunctive case:

$$\begin{aligned} p \vee p &\equiv (p \vee p) \wedge \top && \text{by Identity} \\ &\equiv (p \vee p) \wedge (p \vee \neg p) && \text{by Complement} \\ &\equiv p \vee (p \wedge \neg p) && \text{by Distributivity} \\ &\equiv p \vee \bot && \text{by Complement} \\ &\equiv p && \text{by Identity} \end{aligned}$$

Therefore, both equivalences hold as desired. ∎

***Theorem* 3.2.4. Domination.** For any proposition p, we have $\top \vee p \equiv \top$ and $\bot \wedge p \equiv \bot$.

Proof. Let p be a proposition. We first prove the disjunctive fragment.

$$\begin{aligned} \top \vee p &\equiv (\neg p \vee p) \vee p && \text{by Complement} \\ &\equiv \neg p \vee (p \vee p) && \text{by Associativity} \\ &\equiv \neg p \vee p && \text{by Idempotence, Theorem 3.2.3} \\ &\equiv \top && \text{by Complement} \end{aligned}$$

The conjunctive fragment follows a similar path.

$$\begin{aligned} \bot \wedge p &\equiv (\neg p \wedge p) \wedge p && \text{by Complement} \\ &\equiv \neg p \wedge (p \wedge p) && \text{by Associativity} \\ &\equiv \neg p \wedge p && \text{by Idempotence, Theorem 3.2.3} \\ &\equiv \bot && \text{by Complement} \end{aligned}$$

Thus, the domination laws are proven. ∎

***Theorem* 3.2.5. De Morgan's Laws.** For any propositions p and q, $\neg(p \wedge q) \equiv \neg p \vee \neg q$ and $\neg(p \vee q) \equiv \neg p \wedge \neg q$.

Proof. Let p and q be propositions. We prove $\neg(p \wedge q) \equiv \neg p \vee \neg q$ by applying Theorem 3.2.1. We must show that $(p \wedge q) \wedge (\neg p \vee \neg q) \equiv \bot$ and $(p \wedge q) \vee (\neg p \vee \neg q) \equiv \top$.

First, the conjunctive branch:

$$\begin{aligned}
(p \wedge q) \wedge (\neg p \vee \neg q) &\equiv p \wedge (q \wedge (\neg p \vee \neg q)) && \text{by Associativity} \\
&\equiv p \wedge ((q \wedge \neg p) \vee (q \wedge \neg q)) && \text{by Distributivity} \\
&\equiv p \wedge ((q \wedge \neg p) \vee \bot) && \text{by Complement} \\
&\equiv p \wedge (q \wedge \neg p) && \text{by Identity} \\
&\equiv p \wedge (\neg p \wedge q) && \text{by Commutativity} \\
&\equiv (p \wedge \neg p) \wedge q && \text{by Associativity} \\
&\equiv \bot \wedge q && \text{by Complement} \\
&\equiv \bot && \text{by Domination, Theorem 3.2.4}
\end{aligned}$$

Next, the disjunctive branch:

$$\begin{aligned}
(p \wedge q) \vee (\neg p \vee \neg q) &\equiv ((p \wedge q) \vee \neg p) \vee \neg q && \text{by Associativity} \\
&\equiv (\neg p \vee (p \wedge q)) \vee \neg q && \text{by Commutativity} \\
&\equiv ((\neg p \vee p) \wedge (\neg p \vee q)) \vee \neg q && \text{by Distributivity} \\
&\equiv (\top \wedge (\neg p \vee q)) \vee \neg q && \text{by Complement} \\
&\equiv (\neg p \vee q) \vee \neg q && \text{by Identity} \\
&\equiv \neg p \vee (q \vee \neg q) && \text{by Associativity} \\
&\equiv \neg p \vee \top && \text{by Complement} \\
&\equiv \top && \text{by Domination, Theorem 3.2.4}
\end{aligned}$$

As the premises for Theorem 3.2.1 are satisfied, we conclude $\neg(p \wedge q) \equiv \neg p \vee \neg q$. The proof of $\neg(p \vee q) \equiv \neg p \wedge \neg q$ is analogous and is left as an exercise. ■

3.3 Rules of Inference

The axioms presented in Table 3.1 are equivalences, allowing for the substitution of one formula for another. Our reasoning in the preceding proofs, however, relied on steps not directly justified by these axioms. For instance, we assumed the premises of a conditional to prove its conclusion. Such steps are intuitively sound, but our formal system currently lacks axioms to permit these one-way, inferential arguments. To remedy this, we introduce rules of inference.

A rule of inference takes the form $\Gamma \vdash \phi$, where Γ is a set of propositions called premises, and ϕ is the conclusion that follows from them. The symbol $\vdash$, known as the turnstile, signifies that we can prove ϕ by assuming the statements in Γ are true and applying the axioms, other rules of inference, and any previously proven theorems. If Γ is empty, written $\vdash \phi$, the conclusion can be derived without any additional assumptions. The fundamental rules are summarised in Table 3.2.

The deduction rule is a technical rule that connects the meta-logical symbol $\vdash$ with the logical connective $\to$. It formalises the parallel between a deductive statement "q follows from p" and a conditional statement "if p then q".

If you've read my informal logic notes you would know that Modus ponens[1] is the cornerstone of

[1] *Modus ponens* is short for the Latin phrase *modus ponendo ponens*, literally "the method of putting by placing."

Table 3.2: The rules of inference.

Rule Name	**Notation**	**Description**
Deduction Rule	$(p \vdash q) \vdash (p \to q)$	If, by assuming p, we can prove q, then $p \to q$ is a theorem.
Modus Ponens	$p, (p \to q) \vdash q$	If we have $p \to q$ and we know p, we may deduce q.
Modus Tollens	$\neg q, (p \to q) \vdash \neg p$	If we have $p \to q$ but also $\neg q$, we may infer $\neg p$.
Reductio ad Absurdum	$(\neg p \vdash q), (\neg p \vdash \neg q) \vdash p$	If $\neg p$ leads to a contradiction, we conclude p.

deductive reasoning. Without it, the conclusion of a conditional statement would not be meaningfully conditioned on its premise, rendering hypothetical arguments powerless. Its counterpart, modus tollens,[2] allows for counterfactual reasoning by denying an antecedent when its consequent is known to be false.

The rule of reductio ad absurdum[3] provides the foundation for proof by contradiction. To prove a proposition p, we may hypothetically assume its negation, $\neg p$. If this assumption allows us to derive both some proposition q and its negation $\neg q$, we have derived a contradiction ($q \land \neg q \equiv \bot$). Since our axioms and rules are truth-preserving, this is impossible if the initial premises are true. The only possibility is that our hypothetical assumption, $\neg p$, was false. Therefore, p must be true.

This collection of rules, though small, is sufficient to represent any expressible argument in propositional logic. It is not, however, a minimal set. Modus tollens, for instance, can be derived as a theorem from the other rules.

***Theorem* 3.3.1. Modus Tollens**. For any propositions p and q, we have $\neg q, (p \to q) \vdash \neg p$.

Proof. Let p and q be arbitrary propositions. Assume $\neg q$ and $p \to q$. We know from the definition of the conditional and De Morgan's laws that an implication is equivalent to its contrapositive.[4] Thus, from $p \to q$, we have $\neg q \to \neg p$. Since we have assumed $\neg q$, we may apply modus ponens to $\neg q$ and $\neg q \to \neg p$ to conclude $\neg p$. ■

Note. It is noteworthy that all of propositional logic can be encoded using only two connectives (e.g., $\neg$ and $\to$) and a minimal set of three axioms alongside modus ponens.

One of the most fundamental principles of classical logic, the Law of the Excluded Middle, is a direct consequence of our axioms.

***Theorem* 3.3.2. Law of the Excluded Middle**. For any proposition p, $p \lor \neg p$ is a tautology.

Proof. By the Complement axiom, we have $\neg p \lor p \equiv \top$. By the Commutativity axiom, $\neg p \lor p \equiv p \lor \neg p$. Therefore, $p \lor \neg p \equiv \top$, which means it is always true. ■

There are several classical syllogisms that have been studied since antiquity. Before discussing these, we will first prove three important theorems.

[2] *Modus tollens* is short for the Latin phrase *modus tollendo tollens*, literally "the method of removing by taking away."

[3] *Reductio ad absurdum* is a Latin phrase meaning "reduction to absurdity." This method has also been called *argumentum ad absurdum* and has been a staple of logical reasoning since at least the time of Plato's dialogues.

[4] The equivalence $p \to q \equiv \neg q \to \neg p$ can be proved independently, for example by a truth table or from the axioms using only propositional reasoning and modus ponens. In particular, the proof of this equivalence does *not* rely on modus tollens.

Hilbert's System

The logical system established thus far is user-friendly but not minimal. A more condensed axiomatisation can be constructed with a shorter list of axioms and only a single rule of inference, at the cost of more tedious proofs. Studying one such minimal system, attributed to Hilbert and Frege, provides valuable insight into the foundational structure of logic. A modern version of their system can be built upon the following three axioms, which we now prove as theorems from our current, richer set of rules.

***Theorem* 3.3.3. Hilbert's First Axiom**. For any propositions ϕ and ψ, we have $\vdash \phi \to (\psi \to \phi)$.

Proof. Let ϕ and ψ be arbitrary propositions.

1. Assume ϕ.
2. We wish to show $\psi \to \phi$. To this end, assume ψ.
3. From our first assumption, we already have ϕ.
4. Thus, under the assumption of ψ, we have derived ϕ. This means we have shown $\psi \vdash \phi$.
5. By the Deduction Rule, it follows that $\psi \to \phi$.
6. We derived $\psi \to \phi$ from our initial assumption of ϕ, so we have shown $\phi \vdash (\psi \to \phi)$.
7. Applying the Deduction Rule a final time, we conclude $\vdash \phi \to (\psi \to \phi)$.

■

***Theorem* 3.3.4. Hilbert's Second Axiom**. For any propositions ϕ, ψ, and ξ, we have $\vdash (\phi \to (\psi \to \xi)) \to ((\phi \to \psi) \to (\phi \to \xi))$.

Proof. Let ϕ, ψ, and ξ be arbitrary propositions. Our proof will be a nested series of assumptions and deductions.

1. Assume $\phi \to (\psi \to \xi)$.
2. Assume $\phi \to \psi$.
3. Assume ϕ.
4. From assumption (1) and (3) by modus ponens, we have $\phi, (\phi \to (\psi \to \xi)) \vdash \psi \to \xi$. We now have $\psi \to \xi$.
5. From assumption (2) and (3) by modus ponens, we have $\phi, (\phi \to \psi) \vdash \psi$. We now have ψ.
6. From our results in (4) and (5) by modus ponens, we have $\psi, (\psi \to \xi) \vdash \xi$.
7. We have derived ξ from the assumption of ϕ in step (3). Thus, we have shown $\phi \vdash \xi$. By the Deduction Rule, this yields $\phi \to \xi$.
8. We derived $\phi \to \xi$ from the assumption of $\phi \to \psi$ in step (2). Thus, we have shown $(\phi \to \psi) \vdash (\phi \to \xi)$. By the Deduction Rule, this yields $(\phi \to \psi) \to (\phi \to \xi)$.
9. Finally, we derived $(\phi \to \psi) \to (\phi \to \xi)$ from the assumption of $\phi \to (\psi \to \xi)$ in step (1). A final application of the Deduction Rule yields the desired theorem.

■

***Theorem* 3.3.5. Hilbert's Third Axiom**. For any propositions ϕ and ψ, we have $\vdash (\neg\psi \to \neg\phi) \to (\phi \to \psi)$.

Proof. Let ϕ and ψ be arbitrary propositions.

1. Assume $\neg\psi \to \neg\phi$.

2. We wish to show $\phi \to \psi$. To this end, assume ϕ.
3. We will now argue by contradiction to show ψ. Assume $\neg\psi$.
4. From assumption (1) and (3) by modus ponens, we have $\neg\psi, (\neg\psi \to \neg\phi) \vdash \neg\phi$. We now have $\neg\phi$.
5. However, from assumption (2), we also have ϕ.
6. We have derived both ϕ and $\neg\phi$ from the assumption of $\neg\psi$. This is a contradiction.
7. By reductio ad absurdum, our assumption in (3) must be false. Therefore, we conclude ψ.
8. We derived ψ from the assumption of ϕ in step (2), so we have shown $\phi \vdash \psi$. By the Deduction Rule, this yields $\phi \to \psi$.
9. Finally, we derived $\phi \to \psi$ from the assumption of $\neg\psi \to \neg\phi$ in step (1). The Deduction Rule gives our final result.

■

Classical Syllogisms

We now examine several classical syllogisms (a traditional term for an argument where a conclusion is inferred from a collection of premises), to hone our skills in deductive reasoning and proof-writing.

The following theorem allows for the construction of extended chains of conditional reasoning. Combined with modus ponens, it forms the basis for any non-trivial argument.

***Theorem* 3.3.6. Hypothetical Syllogism**. For any propositions p, q, r, we have $(p \to q), (q \to r) \vdash p \to r$.

Proof. Let p, q, and r be arbitrary propositions. Assume $p \to q$ and $q \to r$. We will show that $p \vdash r$. Assume p. Since we have $p \to q$, we may infer q by modus ponens. Now, since we have q and $q \to r$, a second application of modus ponens yields r. Thus, we have shown $p \vdash r$. By the Deduction Rule, we conclude $p \to r$. ■

The next theorem can be seen as the converse of the Deduction Rule. Together, they establish the formal equivalence between the meta-logical turnstile $\vdash$ and the logical connective $\to$.

***Theorem* 3.3.7. Implication Elimination**. For any propositions p and q, we have $(p \to q) \vdash (p \vdash q)$.

Proof. Let p and q be arbitrary propositions, and assume $p \to q$. We must now show that $p \vdash q$. To this end, assume p. From our initial assumption $p \to q$ and this new assumption p, we may derive q by modus ponens. Thus, $p \vdash q$ is established. ■

***Theorem* 3.3.8. Conjunction Introduction**. For any propositions p and q, we have $p, q \vdash p \wedge q$.

Proof. Let p and q be arbitrary propositions. Assume p and assume q. We proceed by reductio ad absurdum. To this end, assume $\neg(p \wedge q)$.

$$\begin{aligned} \neg(p \wedge q) &\equiv \neg p \vee \neg q && \text{by De Morgan's Law, Theorem 3.2.5} \\ &\equiv p \to \neg q && \text{by Conditional Disintegration} \end{aligned}$$

We now have $p \to \neg q$. Since we assumed p, modus ponens yields $\neg q$. However, we also assumed q. We have thus derived a contradiction (q and $\neg q$). Our assumption $\neg(p \land q)$ must be false. Therefore, we conclude $p \land q$. ∎

Table 3.3: A summary of useful theorems and syllogisms.

Name	**Notation**	**Alternative Name**
Modus Tollens	$\neg q, (p \to q) \vdash \neg p$	
Hypothetical Syllogism	$(p \to q), (q \to r) \vdash p \to r$	
Implication Elimination	$(p \to q) \vdash (p \vdash q)$	Consolidation Rule
Conjunction Introduction	$p, q \vdash p \land q$	Adjunction
Conjunction Elimination	$p \land q \vdash p$	Simplification
Disjunction Introduction	$p \vdash p \lor q$	Addition
Disjunction Elimination	$(p \to r), (q \to r), (p \lor q) \vdash r$	Proof by Cases
Ex Falso Quodlibet	$p, \neg p \vdash q$	Principle of Explosion
Constructive Dilemma	$(\alpha \to \gamma), (\beta \to \delta), (\alpha \lor \beta) \vdash \gamma \lor \delta$	

The results from this section, along with other useful theorems, are summarised in Table 3.3. The proofs for the unproven theorems are left as important exercises for the reader.

3.4 Exercises

1. **Completing Proofs.** The text provided proofs for the first part of several theorems. Using only the axioms from Table 3.1 and previously proven theorems, provide formal proofs for the second part of the following:
   (a) De Morgan's Law: Prove $\neg(p \lor q) \equiv \neg p \land \neg q$.
   Remark. Follow the structure of the proof in Theorem 3.2.5, but demonstrate the uniqueness of $\neg(p \lor q)$.
   (b) Idempotence: Prove $p \lor p \equiv p$.

2. **The Absorption Laws.** Prove the following logical equivalences, which are known as the absorption laws. Justify each step with an axiom or a previously established theorem.
   (a) $p \land (p \lor q) \equiv p$
   (b) $p \lor (p \land q) \equiv p$
   Remark. For part (a), begin by using the Identity axiom to write p as $p \lor \bot$. For part (b), begin by writing p as $p \land \top$.

3. **Fundamental Syllogisms.** Provide formal proofs for the following rules from Table 3.3. You may use axioms, previously proven theorems, and the rules of inference.
   (a) Conjunction Elimination (Simplification): $p \land q \vdash p$.
   (b) Disjunction Introduction (Addition): $p \vdash p \lor q$.

4. **Proof by Cases.** The rule of Disjunction Elimination, $(p \to r), (q \to r), (p \lor q) \vdash r$, is the formal basis for proof by cases. Construct a formal proof for this rule.
   Remark. Your proof might involve assuming $\neg r$ and using modus tollens on the first two premises to derive $\neg p$ and $\neg q$. Then, use conjunction introduction and De Morgan's law.

5. **The Contrapositive.** In the proof of Theorem 3.3.1, the equivalence between an implication and its contrapositive was used. Prove this fundamental equivalence from the axioms.

$$(p \to q) \equiv (\neg q \to \neg p)$$

Remark. Use the Conditional disintegration rule on both sides of the equivalence and then apply axioms such as Double Negation and Commutativity.

6. **The Principle of Explosion.** Prove the principle of *Ex Falso Quodlibet* (from a falsehood, anything follows): $p, \neg p \vdash q$.

Remark. Assume p and $\neg p$. Use Disjunction Introduction to get $\neg p \vee q$. Express this as a conditional and apply modus ponens.

7. **Justifying a Proof.** The following is a proof of Theorem 3.3.3, $\vdash \phi \to (\psi \to \phi)$, with the justifications removed. Provide the correct justification (e.g., "Assumption," "Modus Ponens on steps x, y," "Deduction Rule on steps x-y") for each step.

   1. Assume ϕ.
   2. Assume ψ.
   3. We have ϕ.
   4. Therefore, $\psi \to \phi$.
   5. Therefore, $\phi \to (\psi \to \phi)$.

8. **The Constructive Dilemma.** Provide a formal proof for the Constructive Dilemma, stated in Table 3.3:

$$(\alpha \to \gamma), (\beta \to \delta), (\alpha \vee \beta) \vdash \gamma \vee \delta$$

Remark. This is an advanced application of Proof by Cases. Consider the two cases that arise from the premise $\alpha \vee \beta$. In the case where α holds, what can you deduce about $\gamma \vee \delta$? What about the case where β holds?

9. **Conditional Proofs.** Using the rules of inference and the axiomatic system, prove the following theorem:

$$(p \to q) \vdash ((p \wedge r) \to (q \wedge r))$$

Remark. Use the Deduction Rule twice. Assume $p \to q$ and then assume $p \wedge r$. Use Conjunction Elimination and Modus Ponens to derive q. Then reintroduce r to form the final consequent.

10. $\star$ **The Exclusive Or.** The "inclusive or" ($\vee$) is true if at least one of its arguments is true. The "exclusive or" (XOR, denoted $\oplus$) is true if and only if exactly one of its arguments is true. We can define XOR using our existing connectives as follows:

$$p \oplus q := (p \vee q) \wedge \neg(p \wedge q)$$

Using this definition and the axioms of logic, prove the following properties of XOR:

   (a) Commutativity: $p \oplus q \equiv q \oplus p$
   (b) Self-Inverse: $p \oplus p \equiv \bot$
   (c) Identity: $p \oplus \bot \equiv p$
   (d) Negation: $p \oplus \top \equiv \neg p$

Chapter 4

First Order Logic

"I am in a charming state of confusion."

Ada Lovelace

The language we have described thus far is often called propositional logic because its basic syntactic unit is the proposition. Treating the proposition as the most granular accessible referent keeps the language manageable, but it prevents us from being as expressive as required for many arguments. For example, consider two statements from physics regarding an object's motion. Let us introduce some propositions:

- $a :=$ "The object is accelerating."
- $f :=$ "The net force on the object is zero."
- $r :=$ "The object is at rest."

One physical law can be expressed as follows:

$$\neg a \to f \tag{4.1}$$

This is a statement of Newton's First Law. A different, more specific claim might be:

$$\neg a \to r \tag{4.2}$$

While these two statements, Equation 4.1 and Equation 4.2, share an identical syntactic form, they convey remarkably different ideas. The first is a universal law, claiming something about the sum of all forces acting on the object. The second is an observation about the object's velocity, a claim which is not universally true (an object with no acceleration could be moving at a constant velocity). Propositional logic cannot capture the internal relationship between these concepts; f and r are treated as entirely distinct atomic statements, obscuring the underlying physics they describe.

4.1 A More Expressive Language

This limitation on expressivity is a significant drawback. The missing component in our language is the ability to distinguish the object of our speech from the predicate description we make about it when we declare a proposition. Consider the following mathematical syllogism:

Every prime number greater than 2 is odd.
The number 7 is a prime number greater than 2.
$\therefore$ The number 7 is odd.

This argument appears to be a simple application of modus ponens. Yet, a proof in propositional logic could not invoke modus ponens, as there is no way to symbolise the first sentence to obtain a conditional $p \to q$ where the premise is "The number 7 is a prime number greater than 2." To resolve this, we must augment our language to syntactically distinguish between predicates and the terms they describe. This augmented system is known as first-order logic or predicate logic.

Definition 4.1.1. ***Term***. As noted in the precalc algebra notes a term is a symbol denoting an object. Specific terms (e.g., the natural number 5, the constant π), are called constants. Placeholder terms denoting objects that have not been specifically determined are called variables. A term on its own does not form a complete sentence and has no truth value.

Definition 4.1.2. ***Predicate***. Let $x_1, \dots, x_n$ be variable symbols. We say $\phi(x_1, \dots, x_n)$ is an n-ary predicate if replacing each of the n variables $x_1, \dots, x_n$ with terms $t_1, \dots, t_n$ results in a proposition $\phi(t_1, t_2, \dots, t_n)$ that has a truth value.

- A 1-ary (monadic) predicate describes a property, e.g., $\mathit{IsPrime}(x)$.
- A 2-ary (dyadic) predicate describes a relation between two terms, e.g., $\mathit{IsGreaterThan}(x, y)$.
- An n-ary predicate describes a relation among n terms.

The collection of all terms to which our predicates can refer is our universe of discourse.

With these definitions, we can formalise atomic propositions. If P is the monadic predicate for being prime and 7 is a term, the proposition "7 is prime" is written $P(7)$. If G is the dyadic predicate for "is greater than", and π and 3 are terms, then $G(\pi, 3)$ is the proposition "π is greater than 3". These are the atomic propositions of first-order logic. We can combine them using the logical connectives from propositional logic to form molecular propositions, e.g., $P(x) \to O(x)$, where $O(x)$ is the predicate "x is odd". However, we still cannot translate "Every prime number greater than 2 is odd." For this, we need to express quantities.

Quantification

Let $\phi(x)$ be a predicate containing a variable x. We introduce two symbols, called quantifiers, to express claims about the extent to which $\phi(x)$ is true over the universe of discourse.

The Universal Quantifier. The universal quantification of the variable x in ϕ is denoted $\forall x\, \phi(x)$ and asserts that any constant substituted for x will result in a true proposition. It is read, "For all x, $\phi(x)$."

Intuitively, we can think of universal quantification as a process of exhaustive verification. To determine if $\forall x\, \phi(x)$ is true, we examine every object in the universe. If we find a single object for which ϕ is false (a counterexample), the entire statement is false. If we check every object and find no counterexamples, the statement is true.

Alternatively, if the universe of discourse were a finite set (collection) $\{a_1, a_2, \dots, a_n\}$, the universal quantifier would be equivalent to a conjunction over all elements:

$$\forall x\, \phi(x) \equiv \phi(a_1) \wedge \phi(a_2) \wedge \cdots \wedge \phi(a_n)$$

```
def forall(universe, predicate):
    for x in universe:
        if not predicate(x):
            return False
    return True
```

Figure 4.1: A hypothetical implementation of $\forall x\, \phi(x)$. If it returns False, then there is at least one x in the universe such that $\phi(x)$ is false, which is equivalent to $\forall x\, \phi(x) \equiv \bot$. Otherwise, $\forall x\, \phi(x) \equiv \top$.

The Existential Quantifier. The existential quantification of x is denoted $\exists x\, \phi(x)$ and asserts that there is at least one constant that, when substituted for x, makes $\phi(x)$ true. It is read, "There exists an x such that $\phi(x)$."

Intuitively, this corresponds to a search process. To determine if $\exists x\, \phi(x)$ is true, we search through the universe for an object that satisfies ϕ. If we find such an object (a witness), the statement is true. If we exhaust the entire universe without finding a witness, the statement is false.

If the universe were the finite set $\{a_1, a_2, \ldots, a_n\}$, the existential quantifier would be equivalent to a disjunction over all elements:

$$\exists x\, \phi(x) \equiv \phi(a_1) \vee \phi(a_2) \vee \cdots \vee \phi(a_n)$$

```
def exists(universe, predicate):
    for x in universe:
        if predicate(x):
            return True
    return False
```

Figure 4.2: A hypothetical implementation of $\exists x\, \phi(x)$. If it returns True, then there is at least one x in the universe such that $\phi(x)$ is true, which is equivalent to $\exists x\, \phi(x) \equiv \top$. Otherwise, $\exists x\, \phi(x) \equiv \bot$.

The portion of a formula to which a quantifier applies is its scope, typically denoted by parentheses. A variable that falls within the scope of a quantifier is bound to that quantifier. A variable that is not bound to any quantifier is called free. A formula with free variables is not a proposition; it only becomes a proposition once its free variables are either replaced by terms or bound by a quantifier.

The Unique Existential Quantifier. It is often useful to assert that exactly one object satisfies a predicate. We denote this with the unique existential quantifier, $\exists!$. It is defined in terms of the other quantifiers:

$$\exists! x\, \phi(x) \quad :\equiv \quad \exists x\, (\phi(x) \wedge \forall y\, (\phi(y) \to y = x))$$

This reads, "There exists a unique x such that $\phi(x)$." It asserts both existence and uniqueness.

Well-Formed Formulæ

We can now extend our recursive definition of a proposition to encompass these new syntactic elements.

Definition 4.1.3. ***Well-Formed Formula (WFF)***. A string of symbols λ is a well-formed formula if and only if it can be constructed by applying the following rules:

1. **Base Cases:**
   - The constants $\top$ and $\bot$ are WFFs.
   - If ψ is an n-ary predicate and $t_1, \ldots, t_n$ are terms, then $\psi(t_1, \ldots, t_n)$ is a WFF (an atomic formula).
2. **Recursive Step:** If ϕ and ψ are WFFs, then the following are also WFFs:
   - $(\neg\phi)$, $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \to \psi)$, $(\phi \leftrightarrow \psi)$
   - $\forall x\, \phi$
   - $\exists x\, \phi$

A WFF with no free variables is called a sentence. Only sentences are propositions in first-order logic, as they are the only WFFs guaranteed to have a definite, unambiguous truth value.

Applications and Equivalences

With this formal language, we can now symbolise the prime number argument. Let $P(x)$ be "x is a prime number greater than 2" and $O(x)$ be "x is odd".

1. "Every prime number greater than 2 is odd" is formalised as $\forall x\, (P(x) \to O(x))$.
2. "The number 7 is a prime number greater than 2" is formalised as $P(7)$.
3. From these, we can deduce $O(7)$, "The number 7 is odd."

The deduction is no longer blocked by syntactic limitations.

Quantifier Negation. The universal and existential quantifiers are duals, connected by negation in a manner analogous to De Morgan's laws for conjunction and disjunction.

Theorem **4.1.1. Quantifier Negation**. For any predicate $\phi(x)$, the following equivalences hold:

1. $\neg(\forall x\, \phi(x)) \equiv \exists x\, (\neg\phi(x))$
2. $\neg(\exists x\, \phi(x)) \equiv \forall x\, (\neg\phi(x))$

Proof. We give a semantic (informal) argument based on the standard meanings of $\forall$ and $\exists$.

(1) $\neg(\forall x\, \phi(x)) \equiv \exists x\, (\neg\phi(x))$.

By the semantics of the universal quantifier, $\forall x\, \phi(x)$ means "for every element a in the universe, $\phi(a)$ is true." Formally:

$$\forall x\, \phi(x) \equiv \bigwedge_{a \in U} \phi(a) \equiv \phi(a_1) \wedge \phi(a_2) \wedge \phi(a_3) \wedge \ldots$$

where U is the universe of discourse and $\bigwedge$ denotes a (possibly infinite) conjunction over all elements of U. Now negate this:

$$\begin{aligned}
\neg(\forall x\, \phi(x)) &\equiv \neg\left(\bigwedge_{a\in U} \phi(a)\right) && \text{(definition of } \forall\text{)} \\
&\equiv \bigvee_{a\in U} \neg\phi(a) && \text{(De Morgan's law for (possibly infinite) conjunctions)} \\
&\equiv \exists x\, (\neg\phi(x)) && \text{(definition of } \exists \text{ as a disjunction).}
\end{aligned}$$

Thus $\neg(\forall x\, \phi(x)) \equiv \exists x\, (\neg\phi(x))$.

(2) $\neg(\exists x\, \phi(x)) \equiv \forall x\, (\neg\phi(x))$.

Similarly, by the semantics of the existential quantifier, $\exists x\, \phi(x)$ means "there is at least one element a in the universe such that $\phi(a)$ is true." Formally:

$$\exists x\, \phi(x) \quad \equiv \quad \bigvee_{a\in U} \phi(a),$$

i.e., a (possibly infinite) disjunction over all elements of U. Now negate this:

$$\begin{aligned}
\neg(\exists x\, \phi(x)) &\equiv \neg\left(\bigvee_{a\in U} \phi(a)\right) && \text{(definition of } \exists\text{)} \\
&\equiv \bigwedge_{a\in U} \neg\phi(a) && \text{(De Morgan's law for (possibly infinite) disjunctions)} \\
&\equiv \forall x\, (\neg\phi(x)) && \text{(definition of } \forall \text{ as a conjunction).}
\end{aligned}$$

Thus $\neg(\exists x\, \phi(x)) \equiv \forall x\, (\neg\phi(x))$. ■

These equivalences are fundamental for manipulating quantified statements. From them, we can see that $\forall x\, \phi(x) \equiv \neg(\exists x\, (\neg\phi(x)))$ and $\exists x\, \phi(x) \equiv \neg(\forall x\, (\neg\phi(x)))$.

Traditional Syllogisms. First-order logic can express the four classical categorical propositions from Aristotelian logic. Given geometric predicates like $\mathit{IsSquare}(x)$ and $\mathit{IsRectangle}(x)$:

Table 4.1: The four categorical propositions.

Type	**Statement**	**First-Order Form**
A	All squares are rectangles	$\forall x\, (\mathit{IsSquare}(x) \to \mathit{IsRectangle}(x))$
E	No square is a circle	$\forall x\, (\mathit{IsSquare}(x) \to \neg \mathit{IsCircle}(x))$
I	Some rectangle is a square	$\exists x\, (\mathit{IsRectangle}(x) \land \mathit{IsSquare}(x))$
O	Some rectangle is not a square	$\exists x\, (\mathit{IsRectangle}(x) \land \neg \mathit{IsSquare}(x))$

Remark. Note the use of $\to$ in the universal forms versus $\land$ in the existential forms. A common error is to write $\forall x\, (S(x) \land P(x))$, which makes the much stronger claim that everything in the universe is both S and P.

Just as in propositional logic, there exist formulæ in first-order logic that are true for any arbitrary predicates. For example, $\forall x\, \phi(x) \rightarrow \exists x\, \phi(x)$ is universally true in any non-empty universe of discourse. The study of such universally valid sentences and the rules of inference for deriving them forms the basis of deductive systems for first-order logic.

4.2 Rules of Inference for Quantifiers

Just as our axiomatic system for propositional logic provided rules for manipulating propositions, first-order logic requires additional rules to govern deductions involving quantifiers. These rules allow us to introduce and eliminate quantifiers, forming the bridge between general statements and specific instances.

The four fundamental rules for manipulating quantifiers are summarised in Table 4.2.

Table 4.2: The rules of inference for quantified expressions.

Rule Name	**Notation**	**Description**
Universal Instantiation (Elimination)	$\forall x\, \phi(x) \vdash \phi(t)$	From a universal statement, we may infer any specific instance. The term t can be any constant or variable.
Universal Generalisation (Introduction)	$\phi(t) \vdash \forall x\, \phi(x)$	If we can prove $\phi(t)$ for an arbitrary term t, we may generalise to a universal statement.
Existential Instantiation (Elimination) (Elimination)	$\exists x\, \phi(x) \vdash \phi(c)$	From an existential statement, we may infer the existence of an instance $\phi(c)$, where c is a new constant symbol.
Existential Generalisation (Introduction)	$\phi(t) \vdash \exists x\, \phi(x)$	If we have established a specific instance $\phi(t)$, we may infer that an object with that property exists.

Remark. Formally, Existential Instantiation is used inside a subproof: from $\exists x\, \phi(x)$ we assume $\phi(c)$ for a new constant c, derive some ψ with no free c, and then conclude ψ from $\exists x\, \phi(x)$. Writing $\exists x\, \phi(x) \vdash \phi(c)$ is only a mnemonic for this pattern, not a standalone rule.

Universal Instantiation (UI) is the most straightforward rule; if a property holds for everything, it must hold for any particular thing we choose to examine. Its counterpart, Universal Generalisation (UG), is more subtle. The restriction that the term t must be arbitrary is crucial; it means that no special assumptions can have been made about t. If our proof for $\phi(t)$ would have worked for any term, we are justified in making the universal claim.

Existential Generalisation (EG) allows us to move from a known instance to a claim of existence. If we can show that 7 is prime, we are justified in concluding that there exists a prime number. Existential Instantiation (EI) is the most restrictive rule. If we know that an object with property ϕ exists, we can give it a temporary name, say c, and proceed with our proof using $\phi(c)$. However, this new constant c cannot have appeared anywhere previously in the proof, and any conclusion we draw from it cannot contain c. This ensures we do not accidentally ascribe properties to c that are not guaranteed by its mere existence.

4.3 Theorems on Quantifier Manipulation

Armed with these rules, we can prove several important theorems that are essential for manipulating formulæ in first-order logic.

The first theorem establishes how quantifiers distribute over the logical connectives $\wedge$ and $\vee$.

***Theorem* 4.3.1. Distribution of Quantifiers over Connectives**. Let $\phi(x)$ and $\psi(x)$ be predicates. The following equivalences hold:

1. $\forall x\,(\phi(x) \wedge \psi(x)) \equiv (\forall x\,\phi(x)) \wedge (\forall x\,\psi(x))$
2. $\exists x\,(\phi(x) \vee \psi(x)) \equiv (\exists x\,\phi(x)) \vee (\exists x\,\psi(x))$

Furthermore, the following one-way implications hold:

4. $(\forall x\,\phi(x)) \vee (\forall x\,\psi(x)) \vdash \forall x\,(\phi(x) \vee \psi(x))$
5. $\exists x\,(\phi(x) \wedge \psi(x)) \vdash (\exists x\,\phi(x)) \wedge (\exists x\,\psi(x))$

Remark. The converses of implications (3) and (4) are not tautologies. You will be asked to demonstrate this in the exercises.

The next theorem concerns formulæ where a quantified predicate is combined with a proposition that does not contain the quantified variable.

***Theorem* 4.3.2. Distribution with Propositions**. Let $\phi(x)$ be a predicate and p be a proposition in which the variable x does not appear free. The following equivalences hold:

1. $\forall x\,(p \wedge \phi(x)) \equiv p \wedge (\forall x\,\phi(x))$
2. $\exists x\,(p \wedge \phi(x)) \equiv p \wedge (\exists x\,\phi(x))$
3. $\forall x\,(p \vee \phi(x)) \equiv p \vee (\forall x\,\phi(x))$
4. $\exists x\,(p \vee \phi(x)) \equiv p \vee (\exists x\,\phi(x))$
5. $\forall x\,(p \to \phi(x)) \equiv p \to (\forall x\,\phi(x))$
6. $\exists x\,(p \to \phi(x)) \equiv p \to (\exists x\,\phi(x))$
7. $\forall x\,(\phi(x) \to p) \equiv (\exists x\,\phi(x)) \to p$
8. $\exists x\,(\phi(x) \to p) \equiv (\forall x\,\phi(x)) \to p$

Proof of (7). Let $\phi(x)$ be a predicate and let p be a proposition in which x does not occur free. We give a semantic argument, but write it as an equational chain in the style used earlier.

$\forall x\,(\phi(x) \to p)$	$\equiv \forall x\,(\neg\phi(x) \vee p)$	by Conditional Disintegration
	$\equiv (\forall x\,\neg\phi(x)) \vee p$	by the semantics of $\forall$: "$\forall x(\neg\phi(x) \vee p)$ is true iff either p is true, or $\neg\phi(x)$ holds for every x"
	$\equiv \neg\exists x\,\phi(x) \vee p$	by Quantifier Negation, Theorem 4.1.1
	$\equiv (\exists x\,\phi(x)) \to p$	by Conditional Disintegration.

■

The final theorem addresses the commutativity of quantifiers.

***Theorem* 4.3.3. Quantifier Order**. Let $\phi(x,y)$ be a predicate with at most two free variables. The following hold:

1. $\forall x \forall y\,\phi(x,y) \equiv \forall y \forall x\,\phi(x,y)$
2. $\exists x \exists y\,\phi(x,y) \equiv \exists y \exists x\,\phi(x,y)$
3. $\exists y \forall x\,\phi(x,y) \vdash \forall x \exists y\,\phi(x,y)$

The order of consecutive quantifiers of the same type is irrelevant. However, the order of mixed quantifiers is critical, and the implication in (3) is not an equivalence. To show $\forall x \exists y\,\phi(x,y) \nvdash \exists y \forall x\,\phi(x,y)$, we need only a single counterexample. Let the universe of discourse be the real numbers and let $\phi(x,y)$ be the predicate $x \leq y$.

- The statement $\forall x \exists y \, (x \leq y)$ reads: "For every real number x, there exists a real number y such that $x \leq y$". This is true; for any x, we can choose $y = x$.
- The statement $\exists y \forall x \, (x \leq y)$ reads: "There exists a real number y such that for every real number x, $x \leq y$". This is false; it asserts the existence of a maximum real number, which does not exist.

Since we have found a case where the premise is true and the conclusion is false, the inference is not universally valid.

4.4 Proof Strategies

> The purpose of life is to prove and to conjecture.
>
> — Paul Erdős

The approach to proving a statement depends principally on its syntactic form. The top-level logical structure of a formula dictates the valid proof strategies available, each corresponding to the rules of inference we have established. The first step in constructing a proof is therefore to analyse this structure.

Quantified Statements

Universal Statements. To prove a statement of the form $\forall x \, \phi(x)$, a proof by exhaustion—checking $\phi(t)$ for every possible term t—is seldom feasible for infinite or large domains. The standard method is to employ Universal Generalisation (Table 4.2). One introduces an arbitrary term, say c, from the universe of discourse, about which no assumptions are made beyond those applicable to any member of the universe. A proof is then constructed for $\phi(c)$. If this argument holds without relying on any specific properties of c, it generalises universally.

Existential Statements. A proof of an existential statement $\exists x \, \phi(x)$ is typically constructive. This involves finding a specific term t, called a witness, and demonstrating that $\phi(t)$ holds. By the rule of Existential Generalisation, this single instance is sufficient to prove the theorem. The primary challenge of a constructive proof often lies in the identification of a suitable witness.

Conditional Statements

Statements of the form $p \to q$ are the most common in mathematics. The most common strategy is a direct proof, which relies on the Deduction Rule. The antecedent p is assumed as a premise, and a deductive argument is constructed to derive the consequent q.

An alternative is to prove the contrapositive, $\neg q \to \neg p$. Since a conditional is logically equivalent to its contrapositive ($p \to q \equiv \neg q \to \neg p$), a proof of the latter constitutes a proof of the former. This method is particularly useful when the negation of the consequent, $\neg q$, provides a more concrete starting point than the original antecedent p.

Conjunctions and Disjunctions

Conjunctions. A proof of a conjunction $p \wedge q$ requires two independent proofs: one for p and one for q. The two results are then combined using the rule of Conjunction Introduction.

Disjunctions. To prove a disjunction $p \vee q$, it is sufficient to prove either p or q. A common and powerful technique is to assume the negation of one disjunct and use it to prove the other. For instance, by assuming $\neg p$ and deriving q, one proves the conditional $\neg p \to q$, which is logically equivalent to $p \vee q$.

Indirect and Non-constructive Proofs

Proof by Contradiction. Proof by contradiction, or reductio ad absurdum, is a powerful indirect method. To prove a proposition p, one assumes its negation, $\neg p$. If this assumption leads to a logical contradiction (that is, a proposition of the form $q \wedge \neg q \equiv \bot$), then the initial assumption must be false. By the rule of reductio ad absurdum, one may then conclude that p is true. This technique is often employed when a direct proof seems elusive or overly complex.

The Principle of Explosion. The principle of explosion, or ex falso quodlibet ($p, \neg p \vdash q$), is a related concept. This principle is most directly applicable when proving a conditional statement $p \to q$ where the antecedent p can be shown to be false. Since a false antecedent vacuously implies any consequent, establishing $p \equiv \bot$ is sufficient to prove the conditional immediately.

4.5 Completeness and Decidability

For any formal logical system, two fundamental questions arise regarding its power and limitations:

1. **The Completeness Problem:** Can we establish a finite set of axioms and rules of inference from which every universally true statement (tautology) of the system can be derived?
2. **The Decision Problem:** Can we find a mechanical procedure, or algorithm, that can determine in a finite amount of time whether any given formula is a tautology?

For propositional logic, the answer to both questions is yes. The axiomatic system presented in Table 3.1 is complete, and the truth-table method provides a straightforward, if tedious, decision procedure.

For first-order logic, the situation is more nuanced.

Completeness. First-order logic is complete. In his seminal 1929 dissertation, Kurt Gödel proved the completeness theorem, which states that there exist axiomatic systems capable of deriving every valid formula of first-order logic. The system detailed in section A.7 is one such example. This result is a cornerstone of modern logic, confirming that deductive proof is a sufficient tool for discovering all universal logical truths.

Decidability. In contrast, first-order logic is undecidable. As proven by Alonzo Church and Alan Turing in 1936, no mechanical procedure exists that can correctly determine for every arbitrary formula of first-order logic whether it is a tautology. This has profound implications for the limits of computation and formal reasoning.

To understand this distinction, consider the idea of a "thinking machine."

- **A Theorem-Enumerator:** Based on a complete axiomatic system, one could theoretically construct a machine that mechanically applies the rules of inference to the axioms in every possible way. Such a machine would systematically write down every provable theorem, and thus every tautology, of first-order logic. If a given formula is a tautology, this machine will eventually produce it.
- **A Theorem-Decider:** One cannot, however, construct a machine that accepts an arbitrary formula as input and is guaranteed to halt with a "yes" or "no" answer to the question of its tautology. If the input formula is a tautology, the enumerating machine would eventually find it. But if the formula is *not* a tautology, the machine would run forever, never producing the formula. We would have no way of knowing whether the proof was just around the corner or would never be found.

This result does not mean that we cannot decide for many individual formulæ whether they are tautologies; it simply means there is no universal, "thoughtless" method that works for all of them.

A Note on the Universe of Discourse. The notion of a tautology in first-order logic requires a final specification. Consider the formula $\exists x\, \phi(x) \to \forall x\, \phi(x)$. If our universe of discourse contained only a single object, this formula would be a tautology: if a property is true for at least one object, and there is only one object, then it must be true for all objects. However, in any universe with two or more objects, this is not a tautology. To resolve this, a formula in first-order logic is defined as **valid** (or a tautology) if and only if it is true in every interpretation for every non-empty universe of discourse. It can be proven that this is equivalent to being true in every interpretation with a countably infinite universe.

Remark. You won't understand what countably infinite means till set theory so infinite universe is just fine for now.

4.6 Exercises

1. **Interpreting Formulæ.** Let the universe of discourse be the set of integers, $\mathbb{Z}$. Let $E(x)$ be "x is even," $P(x)$ be "x is positive," and $G(x, y)$ be "$x > y$". Determine the truth value ($\top$ or $\bot$) of the following sentences.
   (a) $\forall x\, (E(x) \lor E(x + 1))$
   (b) $\exists x\, (P(x) \land \neg E(x))$
   (c) $\forall x \exists y\, G(x, y)$
   (d) $\exists y \forall x\, G(x, y)$
   (e) $\forall x\, (P(x) \to \exists y\, (P(y) \land G(x, y)))$
2. **Quantifier Negation.** For each of the statements below, first translate it into a first-order logic formula. Then, find the negation of the formula by applying the quantifier negation

laws from Theorem 4.1.1 until all negation symbols immediately precede predicates. Finally, translate the negated formula back into a clear and natural English sentence.

(a) All birds can fly. (Let $B(x)$ be "x is a bird" and $F(x)$ be "x can fly".)

(b) There is a student who has read every book in the library. (Let $S(x)$ be "x is a student", $B(y)$ be "y is a book in the library", and $R(x,y)$ be "x has read y".)

3. **Scope and Variables.** For the following formula, identify all free and bound variables. For each bound variable, state the scope of the quantifier that binds it.

$$(\forall x\,(P(x) \to Q(y))) \wedge (\exists y\,(Q(y) \vee R(x,y)))$$

4. **Counterexamples.** The text notes that the converses of implications (3) and (4) in Theorem 4.3.1 are not tautologies.

(a) Provide a specific universe of discourse and definitions for predicates $\phi(x)$ and $\psi(x)$ to show that $\forall x\,(\phi(x) \vee \psi(x)) \vdash (\forall x\,\phi(x)) \vee (\forall x\,\psi(x))$ is not a valid rule of inference.

(b) Similarly, provide a counterexample to show that $(\exists x\,\phi(x)) \wedge (\exists x\,\psi(x)) \vdash \exists x\,(\phi(x) \wedge \psi(x))$ is not valid.

5. **Formal Proof with Quantifiers.** Using the rules of inference from Table 4.2 and the axiomatic system for propositional logic, provide a formal proof for the following theorem:

$$(\forall x\,(\phi(x) \to \psi(x))), (\forall x\,\phi(x)) \vdash \forall x\,\psi(x)$$

6. **Formal Proof of Distribution.** Provide a formal proof for the equivalence stated in Theorem 4.3.1(1):

$$\forall x\,(\phi(x) \wedge \psi(x)) \equiv (\forall x\,\phi(x)) \wedge (\forall x\,\psi(x))$$

Remark. To prove an equivalence $\alpha \equiv \beta$, you must prove the two implications $\alpha \vdash \beta$ and $\beta \vdash \alpha$.

7. **Quantifier Order.** The text established with a counterexample that $\forall x \exists y\,\phi(x,y) \nvdash \exists y \forall x\,\phi(x,y)$. Now, provide a formal proof for the other direction, as stated in Theorem 4.3.3(3):

$$\exists y \forall x\,\phi(x,y) \vdash \forall x \exists y\,\phi(x,y)$$

Remark. Be careful with the restrictions on Existential Instantiation (using a new constant) and Universal Generalisation (using an arbitrary variable).

8. **Proof of a Quantifier-Proposition Equivalence.** Provide a formal proof for the equivalence in Theorem 4.3.2(8):

$$\exists x\,(\phi(x) \to p) \equiv (\forall x\,\phi(x)) \to p$$

Remark. A good strategy is to use the conditional disintegration axiom and the quantifier negation laws, similar to the proof provided for part (7).

9. $\star$ **The Empty Universe.** Our definition of a valid formula requires it to be true in every non-empty universe of discourse. Let's explore what happens if we allow the universe to be empty.

(a) In an empty universe, what are the truth values of $\forall x\,\phi(x)$ and $\exists x\,\phi(x)$? Justify your answers by considering the processes of exhaustive verification and search described in the text, or by their hypothetical implementations in Figure 4.1 and Figure 4.2.

(b) Explain why the statement $\forall x\,\phi(x) \to \exists x\,\phi(x)$, which is a tautology in any non-empty universe, is false in an empty universe. This is the primary reason for the "non-empty" restriction in standard first-order logic.

Appendix A

Appendix: Advanced Propositional Concepts

Remark. This section utilises concepts of set theory and mathematical induction. While the recursive definitions in Chapter 2 provide a foundation, readers unfamiliar with formal inductive proofs may wish to treat this section as optional (Or read my algebra notes and not ignore the prerequisites :( ).

The axiomatic system we have developed provides a powerful framework for deduction. However, a deeper examination reveals subtle distinctions and conventions that are crucial for a complete understanding of its scope and limitations. As noted by logicians such as Kurt Gödel, traditional logic often suffered from two deficiencies: a lack of completeness, presenting only an arbitrary selection of logical laws, and a failure to rigorously reduce these laws to a minimal set of primitive axioms. Our axiomatic approach addresses these issues, but in doing so, it makes a specific and important choice regarding the nature of implication.

A.1 Material versus Strict Implication

The conditional connective, $\to$, which we have used extensively, represents *material implication*. Its meaning is defined entirely by its truth table, which specifies the truth value of $p \to q$ for every combination of truth values of p and q. Such a connective, whose output depends solely on the truth values of its inputs, is called *extensional* or a *truth function*.

This must be carefully distinguished from another notion of implication, often what is meant in natural language, where "p implies q" suggests that q is a *logical consequence* of p. This stronger relationship is known as *strict implication*. It asserts that q can be derived from p through a chain of valid deductions.

A connective is *intensional* if its truth value depends on more than just the truth values of its components; it may depend on their meaning or the relationship between them. Strict implication is intensional. Consider two statements:

1. p_1 := "The Earth is a sphere." ($\top$)
2. q_1 := "The Earth is not a flat disc." ($\top$)
3. q_2 := "France is a republic." ($\top$)

Here, the strict implication $p_1 \to q_1$ is true, because the conclusion that the Earth is not a disc is a logical consequence of it being a sphere. However, the strict implication $p_1 \to q_2$ is false; there is no logical connection between the shape of the Earth and the French system of government.

In both scenarios, the constituent propositions are true. If strict implication were a truth function, its output would have to be the same in both cases. Since it is not, no truth table can be constructed for strict implication. The material implication, $p \to q$, is sufficient for our purposes because in order to conclude q from p, it is enough to know the truth-functional statement "if p is true, then q is true," which is precisely what modus ponens requires.

The Paradoxes of Material Implication

The purely truth-functional definition of material implication leads to several theorems that can seem counter-intuitive if one mistakes "$\to$" for logical consequence. These are often called the paradoxes of material implication.

***Theorem* A.1.1.** A true proposition is implied by any proposition. Formally, $\vdash q \to (p \to q)$.

Proof. This was proven as Hilbert's First Axiom, Theorem 3.3.3. ■

***Theorem* A.1.2. Principle of Explosion**. For any propositions p and q, we have $p, \neg p \vdash q$.

Proof. Assume p and $\neg p$. From $\neg p$, we can use Disjunction Introduction (which the reader should have proved as an exercise) to derive $\neg p \lor q$. By the Conditional Disintegration axiom, this is equivalent to $p \to q$. Applying modus ponens to our assumption p and the derived $p \to q$ yields the conclusion q. ■

***Theorem* A.1.3.** A false proposition implies any proposition. Formally, $\vdash \neg p \to (p \to q)$.

Proof. Assume $\neg p$. We wish to show $p \to q$. To this end, assume p. We now hold both p and $\neg p$. By the Principle of Explosion (Theorem A.1.2), we can conclude q. By the Deduction Rule, from $p \vdash q$ we get $p \to q$. A second application of the Deduction Rule yields the final result. ■

***Corollary* A.1.1.** For any two propositions p and q, at least one implies the other. That is, $\vdash (p \to q) \lor (q \to p)$.

Proof. By the Law of the Excluded Middle (Theorem 3.3.2), p must be either true or false.

1. **Case 1: p is true.** By Theorem A.1.1, $q \to p$ must be true. Therefore, the disjunction is true.
2. **Case 2: p is false.** By Theorem A.1.3, $p \to q$ must be true. Therefore, the disjunction is true.

In all possible cases, the statement holds. ■

These results are only paradoxical if "$\to$" is misinterpreted. For instance, "$\neg p \to (p \to q)$" can be read as: "Assume p is false. Then, if we were to suppose p were true, we would have a contradiction, from which anything follows, including q." This is perfectly logical.

Decidability

The semantic and syntactic approaches to logic offer different capabilities. Our axiomatic system provides a method for generating all tautologies from a finite set of axioms, but it does not, in itself, provide a procedure for determining if an arbitrary formula is *not* a tautology.

In contrast, the semantic method of truth tables provides such a procedure. For any given propositional formula, we can construct its truth table and check if the final column contains only $\top$. This is a purely mechanical procedure that is guaranteed to terminate with a definite "yes" or "no" answer. Because such an algorithm exists, we say that propositional logic is *decidable*.

As noted in chapter 2, this method suffers from a combinatorial explosion: a formula with n variables requires a table with 2^n rows. This makes the procedure impractical for even a moderate number of variables, which motivates the use of the more elegant axiomatic method for constructing proofs. Nevertheless, the existence of a decision procedure is of profound theoretical importance. This simplicity is a direct consequence of our use of extensional connectives. Logics that incorporate intensional concepts, such as strict implication, are often much more complex and may not be decidable.

Functional Completeness

The axiomatic approach relies on selecting a minimal set of undefined, or *primitive*, connectives from which all others can be defined. The possibility of such a selection rests on the property of functional completeness.

Definition A.1.1. ***Functional Completeness***. A set of logical connectives is functionally complete if every possible truth function can be expressed by a propositional formula using only connectives from that set.

For instance, the set $\{\neg, \vee\}$ is functionally complete. As shown in our axiomatic system, the other standard connectives can be defined in terms of them:

$$p \wedge q \equiv \neg(\neg p \vee \neg q)$$
$$p \to q \equiv \neg p \vee q$$

Other functionally complete sets include $\{\neg, \wedge\}$ and $\{\neg, \to\}$. More surprisingly, some single connectives are functionally complete. The Sheffer stroke (NAND), denoted $|$, is one such example, where $p|q \equiv \neg(p \wedge q)$. From this single operator, we can define negation and conjunction:

$$\neg p \equiv p|p \qquad\qquad p \wedge q \equiv \neg(p|q) \equiv (p|q)|(p|q)$$

Since $\neg$ and $\wedge$ form a complete set, it follows that $\{|\}$ is also functionally complete.

Not all sets of connectives, however, are complete. Consider the set $\{\neg, \leftrightarrow\}$. We can prove it is not functionally complete by examining the properties of the truth functions it can generate. Let us define an "even" truth function as one whose truth table output column contains an even number of $\top$s (0, 2, or 4 for a two-variable function).

***Theorem* A.1.4.** Any expression in two variables containing only the connectives $\neg$ and $\leftrightarrow$ represents an even truth function.

Proof. We prove the theorem by establishing three lemmas.

1. **Lemma 1:** The atomic propositions p and q are even.

   Proof. The truth table for p over the domain $\{\top, \bot\} \times \{\top, \bot\}$ yields the output column $(\top, \top, \bot, \bot)$. The truth table for q yields $(\top, \bot, \top, \bot)$. Both have two $\top$s, which is an even number. ∎

2. **Lemma 2:** If a formula $\phi(p, q)$ is even, then $\neg\phi(p, q)$ is also even.

   Proof. In a table with four rows, if ϕ has k true outcomes, then $\neg\phi$ has $4 - k$ true outcomes. If k is even (0, 2, or 4), then $4 - k$ is also even (4, 2, or 0 respectively). ∎

3. **Lemma 3:** If formulae $\phi(p, q)$ and $\psi(p, q)$ are even, then $\phi \leftrightarrow \psi$ is also even.

   Proof. Let t_1 be the number of $\top$s for ϕ, t_2 be the number of $\top$s for ψ, and r be the number of rows where both are $\top$. Since ϕ and ψ are even, t_1 and t_2 are even integers. The formula $\phi \leftrightarrow \psi$ is false only when ϕ and ψ have different truth values. The number of cases where ϕ is $\top$ and ψ is $\bot$ is $t_1 - r$. The number of cases where ψ is $\top$ and ϕ is $\bot$ is $t_2 - r$.
   The total number of false outcomes is $(t_1 - r) + (t_2 - r) = t_1 + t_2 - 2r$. Since t_1, t_2, and $2r$ are all even, their sum is even. Thus, the number of false outcomes is even. As the total number of rows (4) is even, the number of true outcomes must also be even. ∎

Any formula constructed using only the variables p, q and connectives $\neg, \leftrightarrow$ is built by iterated application of these connectives to the atomic propositions. Since the atomic propositions are even, and applying the connectives to even formulae yields another even formula, all such constructible formulae must be even. ∎

The truth table for $p \vee q$ has three $\top$s, making it an odd function. Therefore, $p \vee q$ cannot be expressed using only $\neg$ and $\leftrightarrow$, proving the set is not functionally complete.

We can, however, prove that $\{\neg, \vee\}$ is sufficient to express any truth function whatsoever. To show this, we provide a constructive method to build a formula for any given truth table. This constructed formula is known as the *disjunctive normal form*. Consider an arbitrary truth function $f(p, q, r)$.

p	q	r	$f(p, q, r)$	**Associated Minterm (M_i)**
$\top$	$\top$	$\top$	F_1	$p \wedge q \wedge r$
$\top$	$\top$	$\bot$	F_2	$p \wedge q \wedge \neg r$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\bot$	$\bot$	$\bot$	F_8	$\neg p \wedge \neg q \wedge \neg r$

Table A.1: Minterms for a three-variable truth function.

For each row i of the truth table, we construct a corresponding formula M_i, called a *minterm*. A minterm is a conjunction of the variables or their negations; a variable appears negated if its truth value in that row is $\bot$, and unnegated if $\top$. By construction, each minterm M_i is true if and only if the variables have the exact truth values specified in row i.

To construct the formula for f, we identify all rows $\{i_1, i_2, \ldots, i_k\}$ for which the function's output F_{i_j} is $\top$. The desired formula is the disjunction of the corresponding minterms:

$$f(p, q, r) \equiv M_{i_1} \vee M_{i_2} \vee \cdots \vee M_{i_k}$$

This disjunction is true if and only if at least one of its minterms is true. A minterm M_{i_j} is true only for the specific truth-value assignment of row i_j. Therefore, the entire disjunction is true for precisely those rows where f is specified to be true, and false for all others. This method is generalisable to any number of variables. Since the resulting formula uses only $\neg$, $\vee$, and $\wedge$, and since $\wedge$ can be defined from $\neg$ and $\vee$, we have proven that the set $\{\neg, \vee\}$ is functionally complete.

Notational Conventions

To improve readability and reduce the need for parentheses, we adopt a convention for operator precedence. The connectives bind with decreasing strength in the following order:

$$\neg \quad \text{binds strongest}$$

$$\wedge, \vee$$

$$\rightarrow$$

$$\leftrightarrow \quad \text{binds weakest}$$

Thus, the expression $\neg p \vee q \rightarrow r$ is unambiguously parsed as $((\neg p) \vee q) \rightarrow r$.

An alternative syntactic system, developed by the Polish logician Jan Łukasiewicz, dispenses with parentheses entirely. In Polish Notation, the operator precedes its arguments. For example, $p \rightarrow q$ is written $\rightarrow pq$, and $(p \vee q) \wedge r$ becomes $\wedge \vee pqr$. While less intuitive for human reading, this notation is unambiguous and computationally efficient to parse. We shall, however, retain the more conventional infix notation.

A.2 Formal Semantics and Models

Thus far, we have relied on truth tables to determine validity. To treat logic rigorously, we must distinguish between the syntactic form of a sentence and the reality it describes. This relationship is governed by *semantics*.

Interpretations and Models

A formula such as $p \wedge q$ is neither true nor false in a vacuum; it requires a context. An interpretation $\mathcal{A}$ is a function that assigns a fixed truth value (from $\{\top, \bot\}$) to every propositional variable within a formula or set of formulæ.

Definition A.2.1. ***Model***. If a formula ϕ evaluates to $\top$ under a specific interpretation $\mathcal{A}$, we say that $\mathcal{A}$ is a model for ϕ, denoted as:

$$\mathcal{A} \models \phi$$

If $\mathcal{A}$ is a model for a set of formulæ Γ, written $\mathcal{A} \models \Gamma$, then $\mathcal{A}$ must be a model for every formula in Γ.

Using this concept, we can categorize formulæ based on their truth across interpretations:

- **Satisfiable:** A formula ϕ is satisfiable if there exists at least one model $\mathcal{A}$ such that $\mathcal{A} \models \phi$.

- **Unsatisfiable:** A formula is unsatisfiable (or a contradiction) if it has no models. We denote this by $\phi \equiv \bot$.
- **Tautology (Valid):** A formula is valid if every suitable interpretation is a model. We denote this by $\models \phi$.

Logical Consequence We can now refine our definition of logical consequence. A formula ψ is a logical consequence of a set of formulæ Γ, denoted $\Gamma \models \psi$, if every model of Γ is also a model of ψ. This definition captures the idea that ψ "follows from" Γ purely by virtue of the meanings of the connectives, regardless of the specific truth values of the atomic variables.

A.3 Syntactic Calculi

While semantics deals with truth, logic is equally concerned with proof—a purely syntactic verification of statements. A calculus is a framework for expressing and verifying these proofs. Crucially, the verification of a proof in a calculus is a mechanical process that requires no "intelligence" or semantic intuition; it is akin to executing a computer program.

Hilbert-Style Calculi

A Hilbert-style calculus is a system where the objects of manipulation are formulæ themselves. It is defined by a finite set of derivation rules.

Definition A.3.1. ***Derivation Rule***. A derivation rule R specifies how to infer a conclusion from a set of premises. We write $\{\phi_1, \ldots, \phi_k\} \vdash_R \psi$ if ψ can be derived from the set $\{\phi_1, \ldots, \phi_k\}$ using rule R.

Common rules include *Modus Ponens* ($\{p, p \to q\} \vdash q$) and *Tertium Non Datur* ($\vdash p \lor \neg p$), which asserts that for any formula p, the statement "p or not p" is derivable without premises.

Derivations

A proof within a calculus is called a derivation. Formally, a derivation of a formula ψ from a set of assumptions Γ in a calculus $\mathcal{K}$ is a finite sequence of sets of formulæ $\Gamma_0, \Gamma_1, \ldots, \Gamma_n$ such that:

1. $\Gamma_0 := \Gamma$.
2. Each subsequent set Γ_i is formed by adding a formula derived from a subset of Γ_{i-1} using a rule from $\mathcal{K}$.
3. The final formula derived is ψ.

We write $\Gamma \vdash_{\mathcal{K}} \psi$ to signify that such a derivation exists.

A.4 Soundness and Completeness

We now bridge the gap between the syntactic notion of derivability ($\vdash$) and the semantic notion of logical consequence ($\models$). We seek a system where the set of provable theorems is precisely the set of tautologies.

Definition A.4.1. ***Soundness and Completeness***. Let $\mathcal{K}$ be a logical calculus.

- $\mathcal{K}$ is sound if every derivable statement is a logical consequence. Formally, for any set of formulæ Γ and proposition ϕ:
$$\Gamma \vdash_{\mathcal{K}} \phi \implies \Gamma \models \phi$$
- $\mathcal{K}$ is complete if every logical consequence is derivable. Formally:
$$\Gamma \models \phi \implies \Gamma \vdash_{\mathcal{K}} \phi$$

The soundness of our system can be established by verifying two conditions: first, that every axiom in Table 3.1 is a tautology, and second, that every rule of inference in Table 3.2 preserves tautologies (i.e., if the premises are tautologies, the conclusion must also be a tautology). Verification via truth tables confirms this, and the proof is straightforward.

The more profound result is that our system is also complete. This guarantees that our axioms and rules are powerful enough to prove every truth of propositional logic. We now proceed to prove this cornerstone theorem.

Theorem **A.4.1. Completeness of Propositional Logic**. Every tautology is a theorem of our axiomatic system.

The proof is non-trivial and relies on the properties of minterms, as defined in our discussion of functional completeness, and a crucial auxiliary theorem.

Lemma **A.4.1.** Let E be any propositional formula whose variables are a subset of $\{p_1, \ldots, p_n\}$, and let M_i be any minterm of these n variables. Then, at least one of $\vdash M_i \to E$ or $\vdash M_i \to \neg E$ must be true.

Proof. The proof proceeds by structural induction on the formula E, assuming without loss of generality that E is expressed using only the functionally complete set of connectives $\{\neg, \lor\}$.

Base Case. Let E be an atomic proposition p_k, where $1 \le k \le n$. By the definition of a minterm, M_i is a conjunction that must contain either p_k or $\neg p_k$ as one of its conjuncts. The theorem for Conjunction Elimination ($p \land q \vdash p$) can be extended to conjunctions of any length. Thus, if p_k is a conjunct in M_i, we have $M_i \vdash p_k$. If $\neg p_k$ is a conjunct, we have $M_i \vdash \neg p_k$. By the Deduction Rule, this means either $\vdash M_i \to p_k$ or $\vdash M_i \to \neg p_k$. The base case holds.

Inductive Step (Negation). Assume the lemma holds for an arbitrary formula G. We must show it also holds for $\neg G$. The inductive hypothesis states that either $\vdash M_i \to G$ or $\vdash M_i \to \neg G$.

- If $\vdash M_i \to G$, we can use the Law of Double Negation (Theorem 3.2.2), which states $\vdash G \to \neg\neg G$. By Hypothetical Syllogism (Theorem 3.3.6), it follows that $\vdash M_i \to \neg\neg G$.
- If $\vdash M_i \to \neg G$, the condition is met directly.

In either scenario, one of $\vdash M_i \to (\neg G)$ or $\vdash M_i \to \neg(\neg G)$ is demonstrable. Thus, the lemma holds for $\neg G$.

Inductive Step (Disjunction). Assume the lemma holds for formulae G and H. We must show it holds for $G \vee H$. By the inductive hypothesis, we know ($\vdash M_i \to G$ or $\vdash M_i \to \neg G$) and ($\vdash M_i \to H$ or $\vdash M_i \to \neg H$). This gives three exhaustive cases:

- **Case 1: $\vdash M_i \to G$.** By Disjunction Introduction, we know $\vdash G \to (G \vee H)$. Applying Hypothetical Syllogism, we conclude $\vdash M_i \to (G \vee H)$.
- **Case 2: $\vdash M_i \to H$.** Similarly, we know $\vdash H \to (G \vee H)$. By Syllogism, we conclude $\vdash M_i \to (G \vee H)$.
- **Case 3: $\vdash M_i \to \neg G$ and $\vdash M_i \to \neg H$.** From these two premises, we have $M_i \vdash \neg G$ and $M_i \vdash \neg H$. By Conjunction Introduction (Theorem 3.3.8), we can deduce $M_i \vdash \neg G \wedge \neg H$. By De Morgan's Law (Theorem 3.2.5), $\neg G \wedge \neg H \equiv \neg(G \vee H)$. Therefore, $M_i \vdash \neg(G \vee H)$, which by the Deduction Rule implies $\vdash M_i \to \neg(G \vee H)$.

In all possible cases, either $\vdash M_i \to (G \vee H)$ or $\vdash M_i \to \neg(G \vee H)$ is demonstrable. The lemma is proven by induction. ■

With this lemma established, we can complete the proof of the main theorem.

Proof of Theorem A.4.1. Let E be an arbitrary tautology whose variables are among $\{p_1, \ldots, p_n\}$. As per the functional completeness property, we can express E using only the connectives $\neg$ and $\vee$ without loss of generality.

From A.4.1, for any minterm M_i of the n variables, we know that either $\vdash M_i \to E$ or $\vdash M_i \to \neg E$. We can dismiss the second possibility. Suppose, for the sake of contradiction, that for some minterm M_k, we had $\vdash M_k \to \neg E$. By the soundness of our system, this would imply that the formula $M_k \to \neg E$ is a tautology.

However, E is itself a tautology, meaning it is true in every row of the truth table. Consequently, $\neg E$ is a contradiction and is false in every row. By definition, the minterm M_k is true only in row k of the truth table. In this specific row, the expression $M_k \to \neg E$ evaluates to $\top \to \bot$, which has a truth value of $\bot$. A formula that is false in any row cannot be a tautology. This contradicts our deduction from the soundness property. Therefore, the assumption $\vdash M_k \to \neg E$ must be false.

This leaves only one possibility: for every minterm M_i $(i = 1, \ldots, 2^n)$, it must be that $\vdash M_i \to E$.

The final step is to show that if a formula E is implied by every minterm, then E itself must be a theorem. We achieve this by systematically eliminating each variable. Let M'_j be any minterm of the $n-1$ variables $\{p_1, \ldots, p_{n-1}\}$. The expressions $M'_j \wedge p_n$ and $M'_j \wedge \neg p_n$ are two distinct minterms of the full set of n variables. Our result thus guarantees:

$$\vdash (M'_j \wedge p_n) \to E \quad \text{and} \quad \vdash (M'_j \wedge \neg p_n) \to E$$

From these, we have $M'_j, p_n \vdash E$ and $M'_j, \neg p_n \vdash E$. Let us assume M'_j. We have established that under this assumption, E follows from p_n and also from $\neg p_n$. Since $\vdash p_n \vee \neg p_n$ by the Law of the Excluded Middle (Theorem 3.3.2), we can apply the rule for Disjunction Elimination (Proof by Cases) to conclude that E follows from the assumption of M'_j alone. Thus, we have shown $M'_j \vdash E$. By the Deduction Rule, this implies $\vdash M'_j \to E$.

We have successfully reduced the problem, showing that E is implied by every minterm of $n-1$ variables. We can repeat this reduction argument $n-1$ more times, eliminating $p_{n-1}, \ldots, p_1$ in turn. After eliminating all variables, we are left with a statement implied by a minterm of zero variables, which is simply $\top$. This gives the final result $\vdash \top \to E$, which by the Identity axiom simplifies to $\vdash E$.

Finally, our argument was for a tautology E containing only $\neg$ and $\vee$. Any arbitrary tautology P can be converted to an equivalent tautology P_0 using only these connectives by replacing other connectives $(\wedge, \to, \leftrightarrow)$ with their axiomatic definitions. Since P_0 is a tautology, our proof shows $\vdash P_0$. As P is obtained from P_0 by substituting formulae according to our axioms, if P_0 is demonstrable, so too is P. The theorem is proven. ■

Theories and Axiomatic Systems In mathematical practice, we rarely derive tautologies from empty premises. Instead, we reason within a specific context, such as arithmetic, geometry, or algebra. This context is formalised as a theory.

Definition A.4.2. ***Theory***. A theory T is a set of formulæ, which we call the **axioms** of the theory. A formula ϕ is a theorem of the theory T if it is a logical consequence of the axioms, denoted $T \models \phi$.

If a theory T consists of the axioms $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$, then proving a theorem ϕ within this theory is equivalent to proving that the implication $(\alpha_1 \wedge \alpha_2 \wedge \cdots \wedge \alpha_n) \to \phi$ is a tautology. For example, Group Theory is a set of axioms defining the behaviour of a binary operation. Lagrange's theorem is not a tautology of pure logic (it is not true for just any predicates), but it is a theorem of Group Theory because it is a logical consequence of the group axioms.

A.5 Normal Forms and the Resolution Calculus

While Hilbert-style calculi are rigorous, they rely on intuition to select the correct axioms and rules for a derivation. For the purpose of automated reasoning—such as in computer theorem provers—we require a calculus that relies on a purely mechanical process. The *Resolution Calculus* serves this function. To utilise it, however, we must first standardise the structure of our propositions.

Conjunctive Normal Form

In our discussion of functional completeness, we utilised the Disjunctive Normal Form (DNF), which expresses a formula as a disjunction of conjunctions. Resolution requires the dual structure.

Definition A.5.1. ***Literal and Clause***. A literal is an atomic proposition p or its negation $\neg p$. A clause is a finite set of literals $\{L_1, L_2, \ldots, L_k\}$, which is semantically interpreted as their disjunction $L_1 \vee L_2 \vee \cdots \vee L_k$. The empty clause, denoted $\emptyset$, represents the empty disjunction, which is equivalent to $\bot$.

Definition A.5.2. ***Conjunctive Normal Form (CNF)***. A formula is in Conjunctive Normal Form if it is a conjunction of disjunctions of literals. We represent a formula in CNF as a set of clauses $\mathcal{K} = \{C_1, C_2, \ldots, C_m\}$, interpreted as the conjunction $C_1 \wedge C_2 \wedge \cdots \wedge C_m$.

***Theorem* A.5.1. Existence of CNF**. Every propositional formula is logically equivalent to a formula in CNF.

Proof. Given a formula ϕ, we can construct an equivalent CNF formula via its truth table. For every row where ϕ evaluates to $\bot$, we construct a clause that evaluates to $\bot$ only in that specific row. This clause is the disjunction of literals defined as follows: if the atomic variable p_i is $\top$ in that row, include $\neg p_i$; if p_i is $\bot$, include p_i. The conjunction of all such clauses is false exactly when ϕ is false, and true otherwise. ■

Resolution

The resolution calculus, denoted Res, consists of a single rule of inference designed to prove unsatisfiability. To prove that a formula ϕ follows from a set Γ (i.e., $\Gamma \models \phi$), we demonstrate that the clause set corresponding to $\Gamma \cup \{\neg\phi\}$ is unsatisfiable.

Definition A.5.3. ***The Resolution Rule***. Let K_1 and K_2 be two clauses. If there exists a literal L such that $L \in K_1$ and $\neg L \in K_2$, the **resolvent** R is defined as:

$$R = (K_1 \setminus \{L\}) \cup (K_2 \setminus \{\neg L\})$$

We write $\{K_1, K_2\} \vdash_{\text{res}} R$.

This rule captures the intuition that if $A \vee B$ is true and $\neg A \vee C$ is true, then $B \vee C$ must be true.

Theorem **A.5.2. Soundness of Resolution**. If $\mathcal{K} \vdash_{\text{res}} R$, then $\mathcal{K} \models R$.

Proof. Let $K_1, K_2 \in \mathcal{K}$ be the parent clauses resolving on literal L. Let $\mathcal{A}$ be any model of $\mathcal{K}$. We distinguish two cases:

- If $\mathcal{A}(L) = \top$, then $\mathcal{A}(\neg L) = \bot$. Since $\mathcal{A} \models K_2$ (and K_2 contains $\neg L$), $\mathcal{A}$ must satisfy at least one other literal in $K_2 \setminus \{\neg L\}$. Thus $\mathcal{A} \models R$.
- If $\mathcal{A}(L) = \bot$, then since $\mathcal{A} \models K_1$ (and K_1 contains L), $\mathcal{A}$ must satisfy at least one other literal in $K_1 \setminus \{L\}$. Thus $\mathcal{A} \models R$.

In all cases, the resolvent is a logical consequence of the parents. ■

Theorem **A.5.3. Completeness of Resolution**. A set of clauses $\mathcal{K}$ is unsatisfiable if and only if $\mathcal{K} \vdash_{\text{res}} \emptyset$.

Proof. The "if" direction follows immediately from soundness: $\emptyset$ is $\bot$, so if derived, the set must be unsatisfiable. We prove the "only if" direction by induction on the number of atomic variables n in $\mathcal{K}$.

Base Case ($n = 1$): The only unsatisfiable set of clauses involving a single variable p must contain both $\{p\}$ and $\{\neg p\}$ (possibly as subsets of other clauses, but simplification reduces them to this core conflict). Resolving these yields $\emptyset$.

Inductive Step: Assume the theorem holds for any unsatisfiable set with n variables. Let $\mathcal{K}$ be an unsatisfiable set with $n + 1$ variables, $p_1, \ldots, p_{n+1}$. We construct two reduced sets of clauses on n variables:

- $\mathcal{K}_0$: Formed by setting $p_{n+1} = \bot$. Remove all clauses containing $\neg p_{n+1}$ (they are satisfied) and remove p_{n+1} from the remaining clauses.

- $\mathcal{K}_1$: Formed by setting $p_{n+1} = \top$. Remove all clauses containing p_{n+1} and remove $\neg p_{n+1}$ from the remaining clauses.

Since $\mathcal{K}$ is unsatisfiable, both $\mathcal{K}_0$ and $\mathcal{K}_1$ must be unsatisfiable. By the inductive hypothesis, $\mathcal{K}_0 \vdash_{\text{res}} \emptyset$ and $\mathcal{K}_1 \vdash_{\text{res}} \emptyset$.

We can map the derivation $\mathcal{K}_0 \vdash_{\text{res}} \emptyset$ back to the original set $\mathcal{K}$. If a step in $\mathcal{K}_0$ used a clause where p_{n+1} was removed, applying that same step in $\mathcal{K}$ retains the p_{n+1}. Thus, performing the sequence of steps from the $\mathcal{K}_0$ derivation on $\mathcal{K}$ yields either $\emptyset$ or $\{p_{n+1}\}$. Similarly, the derivation from $\mathcal{K}_1$ applied to $\mathcal{K}$ yields either $\emptyset$ or $\{\neg p_{n+1}\}$.

If either derivation yields $\emptyset$, we are done. If not, we have derived both $\{p_{n+1}\}$ and $\{\neg p_{n+1}\}$. One final resolution step between these two yields $\emptyset$. ■

A.6 Exercises

1. **The Paradoxes of Implication.** The text proves $\vdash \neg p \to (p \to q)$ by using the Principle of Explosion. Provide a formal, step-by-step proof of this theorem, justifying each line with an axiom, a rule of inference, or a previously proven theorem. Your proof should explicitly show the use of the Deduction Rule.

2. **Disjunctive Normal Form (DNF).** The proof of functional completeness relies on constructing a formula in Disjunctive Normal Form. Consider the formula for the biconditional, $p \leftrightarrow q$.
   (a) Construct the full truth table for $p \leftrightarrow q$.
   (b) Identify the minterms corresponding to the rows where $p \leftrightarrow q$ is true.
   (c) Write down the Disjunctive Normal Form for $p \leftrightarrow q$ and verify, using the axioms of logic (such as distributivity), that your DNF is logically equivalent to the standard definition $(p \to q) \wedge (q \to p)$.

3. **Functional Incompleteness.** A truth function $f(p_1, \ldots, p_n)$ is called *truth-preserving* if $f(\top, \ldots, \top) = \top$. In other words, it preserves truth if it evaluates to true when all of its arguments are true.
   (a) Show that any formula constructed using only connectives from the set $\{\wedge, \vee, \to, \leftrightarrow\}$ defines a truth-preserving function.

      Remark. Use an argument by structural induction. Show that the atomic propositions are truth-preserving, and that applying any of the allowed connectives to truth-preserving formulæ results in a truth-preserving formula.

   (b) Using the result from part (a), prove that the set $\{\wedge, \vee, \to, \leftrightarrow\}$ is not functionally complete.

      Remark. Find a common, simple connective that is not truth-preserving.

4. $\star$ **The NOR Operator.** The Peirce arrow (NOR), denoted $\downarrow$, is another connective that is functionally complete on its own. Its truth table is defined by $p \downarrow q \equiv \neg(p \vee q)$. Prove that the set $\{\downarrow\}$ is functionally complete by finding formulæ, using only the $\downarrow$ connective, that are logically equivalent to $\neg p$ and $p \vee q$.

5. **Polish Notation.** Translate the following formulæ between standard infix notation and Polish (prefix) notation.

(a) Convert to Polish Notation: $(\neg p \wedge (q \to r)) \leftrightarrow s$
(b) Convert to Infix Notation: $\to \vee \wedge p \neg q r \neg s$

6. **Soundness and Completeness.** Explain the practical consequences of a logical system being:

   (a) Sound but not complete.
   (b) Complete but not sound.

   Which of these two deficiencies would be more catastrophic for the foundations of mathematics, and why?

7. **The Completeness Lemma in Action.** The proof of Theorem A.4.1 depends on the lemma that for any formula E and any minterm M_i, either $\vdash M_i \to E$ or $\vdash M_i \to \neg E$. Consider the formula $E := p \to q$. The four minterms for variables p, q are $M_1 := p \wedge q$, $M_2 := p \wedge \neg q$, $M_3 := \neg p \wedge q$, and $M_4 := \neg p \wedge \neg q$. For each of these four minterms, determine which of the two conditional statements ($M_i \to E$ or $M_i \to \neg E$) is a theorem, and give a brief justification based on the truth values of the antecedent and consequent.

8. **Automated Reasoning.** Use the Resolution Calculus to prove that hypothetical syllogism is a valid inference. That is, show that the set of clauses derived from $\{\neg((p \to q) \wedge (q \to r) \to (p \to r))\}$ is unsatisfiable.

   Remark. First, negate the formula. Then, convert the result into a set of clauses. Finally, apply the resolution rule until you derive the empty clause $\emptyset$.

A.7 An Axiomatic System for First-Order Logic

To provide a complete rigorous foundation, we present a Hilbert-style axiomatic system for first-order logic. This system extends the propositional calculus with the minimal machinery required to handle quantification, allowing all valid formulæ (tautologies) to be derived through finite syntactic steps.

Syntax and Axioms The system defines a minimal set of primitives from which all other logical objects are constructed.

- **Primitives:** The variables (propositional p, individual x, predicate ϕ), the connectives $\neg$ and $\vee$, and the universal quantifier $\forall$.
- **Definitions:**
  - $\phi \to \psi :\equiv \neg\phi \vee \psi$
  - $\exists x\, \phi(x) :\equiv \neg\forall x\, (\neg\phi(x))$

The system relies on four axiom schemes. Let Φ, Ψ, Ξ be well-formed formulæ.

Ax. 1 $\Phi \to (\Psi \vee \Phi)$
Ax. 2 $(\Phi \vee \Psi) \to (\Psi \vee \Phi)$
Ax. 3 $(\Phi \to \Psi) \to ((\Xi \vee \Phi) \to (\Xi \vee \Psi))$
Ax. 4 $\forall x\, \Phi(x) \to \Phi(t)$ (where t is any term substitutable for x in Φ).

Axioms 1 through 3 generate standard propositional logic, while Axiom 4 governs universal instantiation.

Rules of Inference We utilise two rules of inference to manipulate these axioms.

R. 1 Modus Ponens: From Φ and $\Phi \to \Psi$, infer Ψ.
R. 2 Generalisation: If $\Phi \to \Psi(x)$ is proven and x is not free in Φ, infer $\Phi \to \forall x\, \Psi(x)$.

***Theorem* A.7.1.** $\phi(t) \vdash \exists x\, \phi(x)$

Proof. This theorem justifies Existential Generalisation. By Ax. 4, substituting $\neg\phi$ for Φ, we obtain $\forall x\, \neg\phi(x) \to \neg\phi(t)$.

$$\begin{aligned} \forall x\, \neg\phi(x) \to \neg\phi(t) &\equiv \neg(\neg\phi(t)) \to \neg(\forall x\, \neg\phi(x)) && \text{by Transposition} \\ &\equiv \phi(t) \to \neg(\forall x\, \neg\phi(x)) && \text{by Double Negation} \\ &\equiv \phi(t) \to \exists x\, \phi(x) && \text{by definition of } \exists \end{aligned}$$

Given $\phi(t)$, Modus Ponens yields $\exists x\, \phi(x)$. ■

A.8 Exercises

1. **The Rule of Generalisation.** The Rule of Generalisation (R. 2) includes the condition that x must not appear free in Φ.
   (a) Explain why this restriction is necessary for soundness.
   (b) Provide an example of an invalid conclusion derived from a true premise if this restriction is lifted (e.g., using the predicate "x is even").

2. **From Axiom to Rule.** Ax. 4 codifies Universal Instantiation. Explain how the intuitive inference rule $\forall x\, \Phi(x) \vdash \Phi(t)$ is justified by the combination of Ax. 4 and Modus Ponens.

3. **Quantifier Distribution.** Prove the following theorem using the axioms and rules of inference:
$$\vdash (\forall x\, (\Phi(x) \to \Psi(x))) \to ((\forall x\, \Phi(x)) \to (\forall x\, \Psi(x)))$$

4. **Working with Definitions.** Using the definition $\exists x\, \phi(x) :\equiv \neg\forall x\, (\neg\phi(x))$, prove:
$$\vdash \exists x\, (\neg\phi(x)) \to \neg\forall x\, \phi(x)$$

5. **The Deduction Theorem.** While typically a meta-theorem, prove the simplest case of the Deduction Theorem: If there is a single-line proof of Ψ from Φ, show that $\Phi \to \Psi$ is a theorem.

6. $\star$ **Variable Capture.** Ax. 4 requires t to be "substitutable for x". This prevents *variable capture*, where a variable in t becomes inadvertently bound by a quantifier in Φ. Consider $\Phi(x) := \exists y\, (y > x)$ over the real numbers.
   (a) Determine the truth value of $\forall x\, \Phi(x)$.
   (b) Attempt to apply Ax. 4 using the term $t = y$. What is the resulting formula $\Phi(y)$?
   (c) Evaluate the truth of the implication $\forall x\, \Phi(x) \to \Phi(y)$. Explain how this demonstrates the necessity of the substitutability constraint.