

The Foundations: Logic and Proofs

Castor Ye

1 Propositional Logic

1.1 Propositions

A **proposition** is a declarative sentence (that is, a sentence that declares a fact) that is either true or false, but not both.

命题是一个陈述语句（即陈述事实的语句），它可以或真或假，但不能既真又假。

We use letters to denote **propositional variables** (or **sentential variables**), that is, variables that represent propositions, just as letters are used to denote numerical variables.

我们用字母来表示**命题变量**（或称为**语句变量**），即表示命题的变量，就像用字母表示数值变量那样。

The **truth value** of a proposition is true, denoted by T, if it is a true proposition, and the truth of a proposition is false, denoted by F, if it is a false proposition. Propositions that cannot be expressed in terms of simpler propositions are called **atomic propositions**.

如果一个命题是真命题，则它的**真值**为真，用 T 表示；如果它是假命题，则其真值为假，用 F 表示。不能用简单命题表示的命题称为**原子命题**。

New propositions, called **compound propositions**, are formed from existing propositions using **logical operators**.

由已知命题用**逻辑运算符**组合而来的新命题也被称为**复合命题**。

Definition 1 Let p be a proposition. The negation of p , denoted by $\neg p$ (also denoted by \bar{p}), is the statement "It is not the case that p ."

The proposition $\neg p$ is read "not p ." The truth value of the negation of p , $\neg p$, is the opposite of the truth value of p .

定义 1 令 p 为一命题，则 p 的否定记作 $\neg p$ (也可记作 \bar{p})，指“不是 p 所指的情形”。命题 $\neg p$ 读作“非 p ”。 p 的否定 ($\neg p$) 的真值和 p 的真值相反。

Table 1: The Truth Table for the Negation of a Proposition.

p	$\neg p$
T	F
F	T

We will now introduce the logical operators that are used to form new propositions from two or more existing propositions. These logical operators are also called **connectives**.

现在我们将引入从两个或多个已知命题构造新命题的逻辑运算符，这些运算符也称为**联结词**。

Definition 2 Let p and q be propositions. The **conjunction** of p and q , denoted by $p \wedge q$, is the proposition " p and q ." The conjunction $p \wedge q$ is true when both p and q are true and is false otherwise.

定义 2 令 p 和 q 为命题， p, q 的合取即命题“ p 且 q ”，记作 $p \wedge q$ 。当 p 和 q 都是真时， $p \wedge q$ 命题为真，否则为假。

Definition 3 Let p and q be propositions. The **disjunction** of p and q , denoted by $p \vee q$, is the proposition " p or q ." The disjunction $p \vee q$ is false when both p and q are false and is true otherwise.

定义 3 令 p 和 q 为命题， p, q 的析取即命题“ p 或 q ”，记作 $p \vee q$ 。当 p 和 q 都是假时，析取命题 $p \vee q$ 为假，否则为真。

Table 2: The Truth Table for the Con-
junction of Two Propositions

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Table 3: The Truth Table for the Dis-
junction of Two Propositions

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Definition 4 Let p and q be propositions. The **exclusive or** of p and q , denoted by $p \oplus q$ (or p XOR q), is the proposition that is true when exactly one of p and q is true and is false otherwise.

定义 4 令 p 和 q 为命题, p 和 q 的异或 (记作 $p \oplus q$) 是这样命题: 当 p 和 q 中恰好只有一个为真时命题为真, 否则为假。

1.2 Conditional Statement

Definition 5 Let p and q be propositions. The **conditional statement** $p \rightarrow q$ is the proposition "if p , then q ." The conditional statement $p \rightarrow q$ is false when p is true and q is false, and true otherwise. In the conditional statement $p \rightarrow q$, p is called the **hypothesis** (or **antecedent** or **premise**) and q is called the **conclusion** (or **consequence**).

定义 5 令 p 和 q 为命题, 条件语句 $p \rightarrow q$ 是命题 “如果 p , 则 q ”。当 p 为真而 q 为假时, 条件语句 $p \rightarrow q$ 为假, 否则为真。在条件语句 $p \rightarrow q$ 中, p 称为假设 (前件、前提), q 称为结论 (后件)。

The statement $p \rightarrow q$ is called a conditional statement because $p \rightarrow q$ asserts that q is true on the condition that p holds. A conditional statement is also called an **implication**.

语句 $p \rightarrow q$ 称为条件语句, 因为 $p \rightarrow q$ 可以断定条件 p 成立的时候 q 为真, 条件语句也称为**蕴含**。

Table 4: The Truth Table for the Exclusive of Two Propositions

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

Table 5: The Truth Table for the Conditional Statement of Two Propositions

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

CONVERSE, CONTRAPOSITIVE, AND INVERSE: We can form some new conditional statement starting with a conditional statement $p \rightarrow q$. In particular, there are three related conditional statements that occur so often that they have special names. The proposition $q \rightarrow p$ is called the **converse** of $p \rightarrow q$. The **contrapositive** of $p \rightarrow q$ is the proposition $\neg q \rightarrow \neg p$. The proposition $\neg p \rightarrow \neg q$ is called the **inverse** of $p \rightarrow q$. We will see that of these three conditional statements formed from $p \rightarrow q$, only the contrapositive always has the same truth value as $p \rightarrow q$.

逆命题、逆否命题、反命题：由条件语句 $p \rightarrow q$ 可以构成一些新的条件语句，特别是三个常见的相关条件语句还拥有特殊的名称。命题 $q \rightarrow p$ 称为 $p \rightarrow q$ 的**逆命题**，而 $p \rightarrow q$ 的**逆否命题**是命题 $\neg q \rightarrow \neg p$ 。命题 $\neg p \rightarrow \neg q$ 称为 $p \rightarrow q$ 的**反命题**。我们发现，只有逆否命题总是和 $p \rightarrow q$ 有相同的真值。

When two compound propositions always have the same truth values, regardless of the truth values of its propositional variables, we call them **equivalent**. Hence, a conditional statement and its contrapositive are equivalent. The converse and the inverse of a conditional statement are also equivalent.

当两个复合命题总是具有相同真值时，无论其命题变量的真值是什么，我们称它们是**等价的**。因此一个条件语句与它的逆否命题是等价的，条件语句的逆与反也是等价的。

BICONDITIONALS: We now introduce another way to combine propositions that expresses that two propositions have the same truth value.

双条件语句：现在我们介绍另外一种命题复合方式来表达两个命题具有相同真值。

Definition 6 Let p and q be propositions. The **biconditional statement** $p \leftrightarrow q$ is the

proposition "if p and only if q ." The biconditional statement $p \leftrightarrow q$ is true when p and q have the same truth values, and is false otherwise. Biconditional statements are also called **bi-implications**.

定义 6 令 p 和 q 为命题，双条件语句 $p \leftrightarrow q$ 是命题“ p 当且仅当 q ”。当 p 和 q 有同样的真值时，双条件语句为真，否则为假。双条件语句也称为双向蕴含。

Table 6: The Truth Table for the Biconditional $p \leftrightarrow q$

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

1.3 Truth Tables of Compound Propositions

We have now introduce five important logical connectives: conjunction, disjunction, exclusive or, implication, and the biconditional operator, as well as negation. We can use these connectives to build up complicated compound propositions involving any number of propositional variables.

我们已经介绍了五个重要的逻辑联结词：合取、析取、异或、蕴含、双条件。此外，我们还介绍了否定。可以用这些联结词来构造含有一些命题变量的结构复杂的复合命题。

Example 1 Construct the truth table of the compound proposition $(p \vee \neg q) \rightarrow (p \wedge q)$.

Table 7: The Truth Table of $(p \vee \neg q) \rightarrow (p \wedge q)$

p	q	$\neg q$	$p \vee \neg q$	$p \wedge q$	$(p \vee \neg q) \rightarrow (p \wedge q)$
T	T	F	T	T	T
T	F	T	T	F	F
F	T	F	F	F	T
F	F	T	T	F	F

1.4 Precedence of Logical Operators

We can construct compound propositions using the negation operator and the logical operators defined so far. However, to reduce the number of parentheses, we specify an order of precedence for logical operators.

我们可以用所定义的否定运算符和逻辑运算符来构造复合命题，通常使用括号来规定复合命题中的逻辑运算符的作用顺序。然而，为了减少括号的数量，我们规定了逻辑运算符的优先级顺序。

Table 8: Precedence of Logical Operators

Operator	Precedence
\neg	1
\wedge	2
\vee	3
\rightarrow	4
\leftrightarrow	5

1.5 Logic and Bit Operations

Computers represent information using bits. A **bit** is a symbol with two possible values, namely, 0(zero) and 1(one). This meaning of the word bit comes from *binary digit*, because zeros and ones are the digits used in binary representations of numbers. As is customarily done, we will use a 1 bit to represent true and a 0 bit to represent false. That is, 1 represent $T(\text{true})$, 0 represent $F(\text{false})$. A variable is called a **Boolean variable** if its value is either true or false.

计算机用比特表示信息，**比特**是一个具有两个可能值的符号，即 0 和 1。比特一次的含义来自二进制数字，因为 0 和 1 是数的二进制表示中用到的数字。习惯上，我们用 1 表示真，用 0 表示假。如果一个变量的值为真或为假，则此变量就称为**布尔变量**。

Table 9: Table for the Bit Operators OR, AND, and XOR

x	y	$x \vee y$	$x \wedge y$	$x \oplus y$
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

Definition 7 A bit *string* is a sequence of zero or more bits. The *length* of this string is the number of bits in the string.

定义 7 比特串是 0 比特或多比特的序列，比特串的长度就是它所含比特的数目。

1.6 Exercises

常见描述：

- i. p 是 q 的充分条件, q 是 p 的必要条件 $\Rightarrow p \rightarrow q$
- ii. p 是 q 的不充分条件, q 是 p 的不必要条件 $\Rightarrow \neg(p \rightarrow q)$
- iii. p 是 q 的必要不充分条件 $\Rightarrow (q \rightarrow p) \wedge \neg(p \rightarrow q)$
- iv. p 是 q 的充分不必要条件 $\Rightarrow (p \rightarrow q) \wedge \neg(q \rightarrow p)$
- v. p 是 q 的充分必要条件, p 当且仅当 $q \Rightarrow p \leftrightarrow q$

2 Applications of Propositional Logic

2.1 Translating English Sentences

There are many reasons to translate English sentences into expressions involving propositional variables and logical connectives. In particular, English (and every other human language) is often ambiguous. Translating sentences into compound statements removes the ambiguity. Note that may involve making a set of reasonable assumptions based on the intended meaning of the sentence.

有许多理由需要把自然语言翻译成有命题变量和逻辑联结词组成的表达式，特别是，汉语（以及其他各种人类语言）常有二义性，把语句翻译成复合命题可以消除歧义。

Example 2 将该语句翻译为逻辑表达式：如果你身高不足 4 英尺，那么你不能乘坐过山车，除非你已满 16 周岁。

令 q, r, s 分别表示：“你能乘坐过山车”，“你身高不足 4 英尺”，“你已年满 16 周岁”，则可翻译为： $(r \wedge \neg s) \rightarrow \neg q$

3 Propositional Equivalences

3.1 Introduction

An important type of step used in a mathematical argument is the replacement of a statement with another statement with the same truth value. Because of this, methods that produce propositions with the same truth value as a given compound proposition are used extensively in the construction of mathematical arguments. Note that we will use the term “compound proposition” to refer to an expression formed from propositional variables using logical operators, such as $p \wedge q$.

数学证明中使用的一个重要步骤就是用真值相同的一条语句替换另一条语句。因此，从给定复合命题生成具有相同真值命题的方法广泛使用于数学证明的构造。注意，我们用术语“复合命题”来指由命题变量通过逻辑运算形成的一个表达式，如 $p \wedge q$ 。

Definition 8 *A compound proposition that is always true, no matter what the truth values of the propositional variables that occur in it, is called a **tautology**. A compound proposition that is always false is called a **contradiction**. A compound proposition that is neither a tautology nor a contradiction is called a **contingency**.*

定义 8 一个真值永远为真的复合命题称为永真式，一个真值永远为假的复合命题称为矛盾式，既不是永真式又不是矛盾式的复合命题称为可能式。

3.2 Logical Equivalences

Compound propositions that have the same truth values in all possible cases are called **logical equivalent**.

在所有可能的情况下都具有相同真值的两个复合命题称为**逻辑等价的**。

Definition 9 The compound propositions p and q are called **logically equivalent** if $p \leftrightarrow q$ is a tautology. The Notation $p \equiv q$ denotes that p and q are logically equivalent.

定义 9 如果 $p \leftrightarrow q$ 是永真式，则复合命题 p 和 q 称为是逻辑等价的。用记号 $p \equiv q$ 表示 p 和 q 是逻辑等价的。

One way to determine whether two compound propositions are equivalent is to use a truth table. In particular, the compound propositions p and q are equivalent if and only if the columns giving their truth values agree.

判定两个复合命题是否等价的方法之一是使用真值表。特别地，复合命题 p 和 q 是等价的当且仅当对应它们的真值的两列完全一致。

De Morgan laws: $\neg(p \wedge q) \equiv \neg p \vee \neg q$, $\neg(p \vee q) \equiv \neg p \wedge \neg q$

德·摩根律: $\neg(p \wedge q) \equiv \neg p \vee \neg q$, $\neg(p \vee q) \equiv \neg p \wedge \neg q$

Table 10: Logical Equivalences

Equivalence	Name	名称
$p \wedge T \equiv p, p \vee F \equiv p$	Identity laws	恒等律
$p \vee T \equiv T, p \wedge F \equiv F$	Domination laws	支配律
$p \vee p \equiv p, p \wedge p \equiv p$	Idempotent laws	幂等律
$\neg(\neg p) \equiv p$	Double negation law	双重否定律
$p \vee q \equiv q \vee p, p \wedge q \equiv q \wedge p$	Commutative laws	交换律
$(p \vee q) \vee r \equiv p \vee (q \vee r)$	Associative laws	结合律
$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative laws	结合律
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$	Distributive laws	分配律
$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive laws	分配律
$\neg(p \wedge q) \equiv \neg p \vee \neg q, \neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's laws	德·摩根律
$p \vee \neg p \equiv T, p \wedge \neg p \equiv F$	Negation laws	否定律

Table 11: Logical Equivalences Involving Conditional Statements

$p \rightarrow q \equiv \neg p \vee q$	$p \rightarrow q \equiv \neg \neg q \rightarrow \neg p$
$p \vee q \equiv \neg p \rightarrow q$	$p \wedge q \equiv \neg(p \rightarrow \neg q)$
$\neg(p \rightarrow q) \equiv p \wedge \neg q$	
$(p \rightarrow q) \wedge (p \rightarrow r) \equiv p \rightarrow (q \wedge r)$	$(p \rightarrow r) \wedge (q \rightarrow r) \equiv (p \vee q) \rightarrow r$
$(p \rightarrow q) \vee (p \rightarrow r) \equiv p \rightarrow (q \vee r)$	$(p \rightarrow r) \vee (q \rightarrow r) \equiv (p \wedge q) \rightarrow r$
$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$	$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$
$p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$	$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$

The associative law for disjunction shows the expression $p \vee q \vee r$ is well defined, in the sense that it does not matter whether we first take the disjunction of p with q and then the disjunction of $p \vee q$ with r , or if we first take the disjunction of q and r and then take the disjunction of p with $q \vee r$. Similarly, the expression $p \wedge q \wedge r$ is well defined. By extending this reasoning, it follows that $p_1 \vee p_2 \vee \cdots \vee p_n$ and $p_1 \wedge p_2 \wedge \cdots \wedge p_n$ are well defined whenever p_1, p_2, \cdots, p_n are propositions.

析取的结合律表明表达式 $p \vee q \vee r$ 下下面的意义下是良定义的：无论是先做 p 和 q 的析取再做 $p \vee q$ 和 r 的析取，还是先做 q 和 r 的析取再做 p 和 $q \vee r$ 的析取，其结果都是一样的。同样， $p \wedge q \wedge r$ 也是良定义的。扩展这一推理过程可以得到：只要 p_1, p_2, \cdots, p_n 为命题， $p_1 \vee p_2 \vee \cdots \vee p_n$ 和 $p_1 \wedge p_2 \wedge \cdots \wedge p_n$ 均有定义。

Furthermore, note that De Morgan's laws extend to:

$$\neg(p_1 \vee p_2 \vee \cdots \vee p_n) \equiv (\neg p_1 \wedge \neg p_2 \wedge \cdots \wedge \neg p_n)$$

$$\neg(p_1 \wedge p_2 \wedge \cdots \wedge p_n) \equiv (\neg p_1 \vee \neg p_2 \vee \cdots \vee \neg p_n)$$

德·摩根律可以扩展为：

$$\neg(p_1 \vee p_2 \vee \cdots \vee p_n) \equiv (\neg p_1 \wedge \neg p_2 \wedge \cdots \wedge \neg p_n)$$

$$\neg(p_1 \wedge p_2 \wedge \cdots \wedge p_n) \equiv (\neg p_1 \vee \neg p_2 \vee \cdots \vee \neg p_n)$$

We will sometimes use the notation $\bigvee_{j=1}^n p_j$ for $p_1 \vee p_2 \vee \cdots \vee p_n$ and $\bigwedge_{j=1}^n p_j$ for $p_1 \wedge p_2 \wedge \cdots \wedge p_n$. Using this notation, the extended version of De Morgan's laws can be written concisely as $\neg(\bigvee_{j=1}^n p_j) \equiv \bigwedge_{j=1}^n \neg p_j$ and $\neg(\bigwedge_{j=1}^n p_j) \equiv \bigvee_{j=1}^n \neg p_j$.

我们有时用符号 $\bigvee_{j=1}^n p_j$ 来表示 $p_1 \vee p_2 \vee \cdots \vee p_n$, 用 $\bigwedge_{j=1}^n p_j$ 来表示 $p_1 \wedge p_2 \wedge \cdots \wedge p_n$ 。德·摩根律就可以改写为: $\neg(\bigvee_{j=1}^n p_j) \equiv \bigwedge_{j=1}^n \neg p_j$ 和 $\neg(\bigwedge_{j=1}^n p_j) \equiv \bigvee_{j=1}^n \neg p_j$ 。

4 Predicates and Quantifiers

4.1 Predicates

The statement “ x is greater than 3” has two parts. The first part, the variable x , is the subject of the statement. The second part - the **predicate**, “is greater than 3” - refers to a property that the subject of the statement can have. We can denote the statement “ x is greater than 3” by $P(x)$, where P denotes the predicate “is greater than 3” and x is the variable. The statement $P(x)$ is also said to be the value of the **propositional function** P at x . Once a value has been assigned to the variable x , the statement $P(x)$ becomes a proposition and has a truth value.

语句 “ x 大于 3” 有两个部分。第一部分即变量 x 是语句的主语。第二部分 (谓词 “大于 3”) 表明语句的主语具有一个性质。我们可以用 $P(x)$ 表示语句 “ x 大于 3”, 其中 P 表示谓词 “大于 3”, 而 x 是变量。语句 $P(x)$ 也可以说成是命题函数 P 在 x 的值。一旦给变量 x 赋一个值, 语句 $P(x)$ 就成为命题并具有真值。

Example 3 令 $P(x)$ 表示语句 “ $x > 3$ ”, 求 $P(2), P(4)$ 的真值。

$$P(2) \Rightarrow 2 > 3 = F, P(4) \Rightarrow 4 > 3 = T$$

In general, a statement involving the n variables x_1, x_2, \cdots, x_n can be denoted by $P(x_1, x_2, \cdots, x_n)$. A statement of the form $P(x_1, x_2, \cdots, x_n)$ is the value of the **propositional function** P at the n -tuple (x_1, x_2, \cdots, x_n) , and P is also called an **n -place predicate** or an **n -ary predicate**.

一般地, 涉及 n 个变量 x_1, x_2, \cdots, x_n 的语句可以表示成 $P(x_1, x_2, \cdots, x_n)$, 形式为 $P(x_1, x_2, \cdots, x_n)$ 的语句是**命题函数** P 在 n 元组 (x_1, x_2, \cdots, x_n) 中的值, P 也称为 n 位谓词或 n 元谓词。

PRECONDITIONS AND POSTCONDITIONS: Predicates are also used to establish the correctness of computer programs, that is, to show that computer programs always produce the desired output when given valid input. The statements that describe

valid input are known as **preconditions** and the conditions that the output should satisfy when the program has run are known as **postconditions**.

前置条件和后置条件：谓词还可以用来验证计算机程序，也就是证明当给定合法输入时计算机程序总是能产生所期望的输出。描述合法输入的语句叫作**前置条件**，而程序运行的输出应满足的条件称为**后置条件**。

4.2 Quantifiers

We will focus on two types of quantification here: universal quantification, which tells us that a predicate is true for every element under consideration, and existential quantification, which tells us that there is one or more element under consideration for which the predicate is true. The area of logic that deals with predicates and Quantifiers is called the **predicate calculus**.

这里我们集中讨论两类量化：全称量化，它告诉我们一个谓词在所考虑范围内对每一个体都为真；存在量化，它告诉我们一个谓词在所考虑范围内对一个或多个个体为真。处理谓词和量词的逻辑领域称为**谓词演算**。

THE UNIVERSAL QUANTIFIER Many mathematical statements assert that a property is true for all values of a variable in a particular domain, called the **domain of discourse** (or the **universe of discourse**), often just referred to as the **domain**. The domain must always be specified when a universal quantifier is used; without it, the universal quantification of a statement is not defined.

全称量词许多数学命题断言某一性质对于变量在某一特定域内的所有值均为真，这一定域称为变量的**论域**（或**全体域**），时常简称为**域**。在使用全称量词时必须指定论域，否则语句的**全称量化**就是无定义的。

Definition 10 *The **universal quantification** of $P(x)$ is the statement " $P(x)$ for all values of x in the domain."* The notation $\forall xP(x)$ denotes the universal quantification of $P(x)$. Here \forall is called the **universal quantifier**. We read $\forall xP(x)$ as "*for all $xP(x)$* " or "*for every $xP(x)$* ." An element for which $P(x)$ is false is called a counterexample to $\forall xP(x)$.

定义 10 $P(x)$ 的全称量化是语句“ $P(x)$ 对 x 在其论域的所有值为真。”符号 $\forall xP(x)$ 表示 $P(x)$ 的全称量化，其中 \forall 称为全称量词。命题 $\forall xP(x)$ 读作“对所

有 x , $P(x)$ ” 或 “对每个 x , $P(x)$ ”。一个使 $P(x)$ 为假的个体称为 $\forall xP(x)$ 的反例。

Table 12: Quantifiers

Statement	When True?	When False?
$\forall xP(x)$	$P(x)$ is true for every x .	There is an x for which $P(x)$ is false.
$\exists xP(x)$	There is an x for which $P(x)$ is true.	$P(x)$ is false for every x .

THE EXISTENTIAL QUANTIFIER: Many mathematical statements assert that there is an element with a certain property. Such statements are expressed using existential quantification. With existential quantification, we form a proposition that is true if and only if $P(x)$ is true for at least one value of x in the domain.

存在量词: 许多数学定理断言: 有一个个体使得某种性质成立。这类语句可以用存在量化表示。我们可以用存在量化构成这样一个命题: 该命题为真当且仅当论域中至少有一个 x 的值使得 $P(x)$ 为真。

Definition 11 *The **existential quantification** of $P(x)$ is the proposition "There exists an element x in the domain such that $P(x)$." We use the notation $\exists xP(x)$ for the existential quantification of $P(x)$. Here \exists is called the **existential quantifier**.*

定义 11 $P(x)$ 的存在量化是命题 “论域中存在一个个体 x 满足 $P(x)$ ”。我们用符号 $\exists xP(x)$ 表示 $P(x)$ 的存在量化, 其中 \exists 称为存在量词。

A domain must always be specified when a statement $\exists xP(x)$ is used. Furthermore, the meaning of $\exists xP(x)$ changes when the domain changes. Without specifying the domain, the statement $\exists xP(x)$ has no meaning.

当使用语句 $\exists xP(x)$ 时, 必须指定一个论域。而且, 当论域变化时, $\exists xP(x)$ 的意义也随之改变。如果没有指定论域, 那么语句 $\exists xP(x)$ 没有意义。

THE UNIQUENESS QUANTIFIER: Of these other Quantifiers, the one that is most often seen is the **uniqueness quantifier**, denoted by $\exists!$ or \exists_1 .

唯一性量词: 所有其他量词中最常见的是**唯一性量词**, 用符号 $\exists!$ 或 \exists_1 表示。 $\exists!xP(x)$ 这种表示法指 “存在一个唯一的 x 使得 $P(x)$ 为真”。

4.3 QUANTIFIERS OVER FINITE DOMAINS

When the domain of a quantifier is finite, that is, when all its elements can be listed, quantified statements can be expressed using propositional logic. In particular, when the elements of the domain are x_1, x_2, \dots, x_n , where n is a positive integer, the universal quantification $\forall x P(x)$ is the same as the conjunction $P(x_1) \wedge P(x_2) \wedge \dots \wedge P(x_n)$ because this conjunction is true if and only if $P(x_1), P(x_2), \dots, P(x_n)$ are all true.

当一个量词的域是有限的时候，即所有元素可以一一列出时，量化语句就可以用命题逻辑来表达。特别是，当论域中的元素为 x_1, x_2, \dots, x_n ，其中 n 是一个正整数，则全称量化 $\forall x P(x)$ 与合取式 $P(x_1) \wedge P(x_2) \wedge \dots \wedge P(x_n)$ 想同，因为这一合取式为真当且仅当 $P(x_1), P(x_2), \dots, P(x_n)$ 全部为真。

4.4 Negating Quantified Expressions

考虑语句的否定：班上每一个学生都学过一门微积分课。

则全命题为： $\forall x P(x)$ ，其否定为： $\exists x \neg P(x)$ 。

德·摩根律： $\neg \exists x P(x) \equiv \forall x \neg P(x)$, $\neg \forall x P(x) \equiv \exists x \neg P(x)$