1 基础：逻辑与证明

逻辑规则给出数学语句的准确含义。例如，这些规则有助于我们理解下列语句机器推理："存在一个整数，它不是两个整数的平方和"，以及"对每个正整数 n ，小于等与 n 的正整数之和是 $n(n+1)/2$"。逻辑是所有数学推理的基础，也是所有自动推理的基础。对计算机的设计、系统规范说明、人工智能、计算机程序设计、程序设计语言以及计算机科学的其他许多研究领域，逻辑都有实际应用。

为理解数学，我们必须理解正确的数学论证是由什么组成的。只要证明一个数学语句是真的，我们就称之为一个定理。关于一个主题的定理的集合就组成我们对这个主题的认知。为了学习一个数学主题，同我们需要积极地构关于此主题的数学论证，而不仅仅是阅读论证。此外，了解一个定理的证明通常就有可能通过细小的改动来获得适应新情况的结论。

每个人都知道证明在数学中的重要性，大许多人对于证明在计算机科学中的重要程度感到惊讶。事实上，证明常常用于验证计算机陈翔对所有可能的输入值产生正确输出值，用于解释算法总是产生正确结果，用于建立一个系统的安全性，以及用于创建人工智能系统。而且，自动推理系统已经被创造出来，让计算机自己来构造证明。

2 命题逻辑

2.1 引言

逻辑规则给出数学语句的准确含义，这些规则用来区分有效和无效的数学论证。由于本书的一个主要目的是教会读者如何理解和构造正确的数学论证，所以我们从介绍逻辑开始离散数学的学习。

逻辑不仅对理解数数学推理十分重要，而且在计算机科学中有许多应用。这些逻辑规则用于计算机电路设计、计算机程序的构造、程序正确性证明以及许多其他方面。而且，已经开发了一些软件系统用于自动构造某些类型的证明。在随后的几章中将逐一讨论这些应用。

2.2 命题

逻辑的基本组件——命题。命题是一个陈述语句，它或真或假，但不能即真又假。

我们用字母来表示命题变元，它是表示命题的变量。习惯上用字母 $p, q, r, s, \cdots$ 表示命题。如果一个命题是真命题，它的真值为真，用 T 表示；如果它是假命题，其真值为假，用 F 来表示。

涉及命题的逻辑领域称为命题演算或命题逻辑。它最初是 2300 多年前由古希腊哲学家亚里士多德系统地创建的。

现在我们转而关注从已有命题产生新命题的方法。这些方法由英国数学家布尔在他的题为《The Laws of Thought》的书中讨论过。去多数学陈述都是由一个或多个命题组合而来。称为复合命题的新命题是由已知命题用逻辑运算符组合而来。

定义 1 令 p 为一命题，则 p 的否定记作 $\neg p$，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。

例 3 找出命题“Michael 的 PC 运行 Linux。”的否定，并用中文表示。

下表是命题 p 及其否定的真值表。此表列出命题 p 的两种可能真值。每一行显示对应于 p 的真值时 $\neg p$ 的真值。

Table 1: 命题之否定的真值表

p	$\neg p$
T	F
F	T

命题的否定也可以看做否定运算符作用在命题上的结果。否定运算符从一个已知命题构造出一个新命题。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。这些逻辑运算符也称为联结词。

定义 2 令 p 和 q 为命题。p、q 的合取即命题“p并且 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。

注意在逻辑合取中，有时候用“但是”一词。比如，语句“阳光灿烂，但是在下雨”是“阳光灿烂并且下雨”一句的另一种说法。

例 5 找出命题 p 和 q 的合取，其中 p 命题“Rebecca 的 PC 至少有 16 GB 空闲磁盘空间”，q 为命题“Rebecca 的 PC 处理器的速度大于 1 GHz”。

定义 3 令 p 和 q 为命题。p q 的析取式即命题“p或q”，记作 $p \vee q$ 。当 p和q 均为假时，析取式 $p \vee q$ 才为假，否则为真。

在析取中使用的联结词或（or ）对应于或在自然语言中所使用的两种情况之一, 即兼或 (inclusive or)。另一方面, 当我们说“学过微积分或学过计算机科学, 但不是两者都学过的学生，可以选修本课程”的时候使用的是异或（exclusive or）。这里的意思是既学过微积分，又学过计算机科学的学生不能选修本课程。只有那些恰好在这两门课中修过一门的学生可以选修本课程。

定义 4 令 p q 为命题。p和q 的异或是这样一个命题：当 p和q 中恰好只有一个为真时命题为真，否则为假。

2.3 条件语句

下面讨论其他几个重要的命题合成方式。

定义 5 令 p q 为命题. 条件语句 $p \to q$ 是命题“如果 p, 则 q”。当 p 为真而 q 为假时, 条件语句 $p \to q$ 为假，否则为真。在条件语句 $p \to q$ 中，p 称为假设, q 称为结论.

由于条件语句在数学推理中具有重要的作用，所以表达 $p \to q$ 的术语也很多，即使不是全部，你也会碰到下面几个常见的条件语句的表述方式:

”如果p 则q”

”p蕴含q”

”如果p, q”

”p仅当q”

”p是q的充分条件”

"q的充分条件是p"

"q如果p"

"q每当q"

"q当p"

"q是p的必要条件"

"p的必要条件是q"

"q由p得出"

"q除非$\neg p$"

条件语句 $p \to q$ 的众多表达方式中有两个最容易引起混淆的是："p仅当q "和"q除非p "。因此，这里提供一些消除混淆的建议。

请记住"p仅当q "表达了与"如果p 则q" 同样的以，注意"p仅当q "说的是当 q 不为真时 p 不能为真。要小心不用用"q仅当p"来表达 $p \to q$，因为这是错误的。要明白这一点，请注意当 p 和 q 取不同的真值时，"q仅当p "和 $p \to q$ 的真值是不同的。

例 7 令 p 为语句"Maria 学习离散数学", q 为语句"Maria 会找到好工作"。用中文表达语句 $p \to q$ 。

" 如果 Maria 学习离散数学，那么她会找到好工作"

" 当 Maria 学习了离散数学，她会找到月份好工作"

"Maria 会找到一份好工作，她只要学习了离散数学就足够了"

"Maria 会找到一份好工作，除非她不学习离散数学"

许多程序设计语言总使用的 if-then 结构与逻辑中使用的不同。大部分程序设计语言中都有 if p then S 这样的语句，其中 p 是命题而 S 是一段程序段。当程序的运行遇到这样一条语句时，如果 p 为真，就执行 S；如果 p 为假，则 S 不执行。

逆命题、逆否命题与反命题由条件语句 $p \to q$ 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 $q \to p$ 称为 $p \to q$ 的逆命题，而 $p \to q$ 的逆否命题是命题 $\neg q \to \neg p$ 。命题 $\neg p \to \neg q$ 称为反命题。我们发现，三个由 $p \to q$ 衍生的条件语句中，只有逆否命题总是和 $p \to q$ 具有相同的真值。

当两个复合命题总是具有相同的真值，我们称它们是等价的。

双条件语句下面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的真值。

定义 6 令 p 和 q 为命题。双条件语句 $p \to q$ 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。

"p是q 的充分必要条件"

" 如果 p 那么 q，反之依然"

"p 当且仅当 q"。

例 10 令 p 语句"你可以搭乘该航班"，令 q 为语句"你买票了"。则 $p \leftrightarrow q$ 为语句"你可以搭乘该航班当且仅当你买机票了"

此语句为真，如果 p 和 q 均为真或均为假。

双条件的隐式使用你应该意识到在自然语言中双条件并不总是显式地使用。特别是在自然语言中很少使用双条件中的"当且仅当"结构。通常用"如果，那么"或"仅当"结构来表示蕴含。"当且仅当"的另一部分是隐含的。也就是逆命题是蕴含的而没有说明出来。例如，考虑这个自然语言的语句"如果你吃晚饭了，则可以吃餐后甜点"。其真正含义是"你可以吃餐后甜点当且仅当你吃完饭"。后而这个语句在逻辑上等价于两个语句"如果你吃晚饭了，那么你可以吃

运算符	优先级
$\neg$	1
$\wedge$ $\vee$	2 3
$\rightarrow$ $\leftrightarrow$	4 5

甜点”和“仅当你吃完了饭，你才能吃甜点”。由于自然语言的这种不精确性，所以我们需要对自然语言中的条件语句是否蕴含它的逆做出假设。因为数学和逻辑注重精确，所以我们总是区分条件语句 $p \rightarrow q$ 和双条件语句 $p \leftrightarrow q$。

2.4 复合命题的真值表

我们已经介绍了否定以及 4 个重要的逻辑联结词——合取、析取、条件、双条件。我们可以用这些联结词来构造含有一些命题变元的季候复杂的复合命题。我们可以用真值表来决定这些复合命题的真值。

2.5 逻辑运算符的优先级

现在，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。我们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。例如，$(p \vee q) \wedge (\neg r)$ 是 $p \vee q$ 和 $\neg r$ 的合取。然而，为了减少括号的数量，我们规定否定运算符先于所有其他运算符。这意味着 $\neg p \wedge q$ 是 $\neg p$ 和 q 的合取，即 $(\neg p) \wedge q$ ，而不是 p 和 q 的合取的否定，即 $\neg(p \wedge q)$。另一个长用的优先级规定则是合取运算符优先于析取运算符，因此 $p \wedge q \vee r$ 意味着 $(p \wedge q) \vee r$ ，而不是 $p \wedge (q \vee r)$ 。因为这个规则不太好记，所以我们将继续使用括号是析取运算符和合取运算符的作用顺序看起来很明了。

最后，一个已被接受的规则是条件运算符和双条件运算符的优先级低于合取和析取运算符。因此，$p \vee q \rightarrow r$ 等同于 $(p \vee q) \rightarrow r$ 。当设计条件运算符和双条件运算符的作用顺序时，我们也将使用括号，尽管条件运算的优先级高于双条件运算。

2.6 逻辑运算和位运算

计算机用位表示信息。位是一个具有两个可能值的符号，即 0 和 1。位一词的含义来自二进制数字（binary digit），因为 0 和 1 是数的二进制表示中用到的数字。1946 年著名的统计学家约翰-图基引入了这一术语。一位可以用于表示真值，因为只有两个真值，即真和假。习惯上，我们用 1 表示真，用 0 表示假。即，1 表示 T ，0 表示 F 。如果一个变量的值或为真或为假，则此变量称为布尔变量。于是一个布尔变量可以用一位表示。

2.7 习题

1. 下列那些语句是命题？这些是命题的语句的真值是什么？

   (a) 南宁是广西的首府

   (b) 2+3=5

(c) $5+7=10$

(d) $x+2=11$

(e) 回答这个问题

2. 下列个命题的否命题是什么?

   (a) Mei 有一台 MP3 播放器

   (b) 右江区没有污染

   (c) $2+1=3$

   (d) 右江区的夏天又热又晒

3. 下列各命题的否定是什么?

   (a) Steve 的笔记本电脑有大于 100GB 的空闲磁盘空间。

   (b) Zach 阻止来自 Jennifer 的邮件和短信。

   (c) $7\cdot 11\cdot 13=999$

   (d) Diane 周日骑自行车骑了 200 公里。

4. 假设在最近的财年期间，Acme 计算机公司的年收入是 1380 亿美元且其净利润是 80 亿美元，Nadir 软件公司的年收入是 870 亿美元且将利润是 50 亿美元，Quixote 媒体的年收入是 1110 亿美元且净利润是 130 亿美元。试判断有关最近财年的每个命题的真值。

   (a) Quixote 媒体的年收入最多。

   (b) Nadir 软件公司的净利润最少且 Acme 计算机公司的年收入最多。

   (c) Acme 计算机公司的净利润最多或者 Quixote 媒体的净利润最多。

   (d) 如果 Quixote 媒体的净利润最少，则 Acme 计算机公司的年收入最多。

   (e) Nadir 软件公司的净利润最少当且仅当 Acme 计算机公司的年收入最多。

5. 令 p 和 q 分布表示命题“在新泽西海岸游泳是允许的”和“在海岸附近发现过鲨鱼”。试用汉语表达下列每个复合命题。

   (a) $\neg q$

   (b) $p\wedge q$

   (c) $\neg p\vee q$

   (d) $p\to\neg q$

   (e) $\neg q\to p$

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

   (g) $p\neg q$

   (h) $\neg p\wedge(p\vee\neg q)$

6. 令 p 、q 为如下命题：p：气温在零度以下 q：正在下雪。用 p 、q 和逻辑联结词写出下列各命题：

   (a) 气温在零度以下且正在下雪。
   (b) 气温在零度以下，但没有下雪。
   (c) 气温不在零度以下，清切没有下雪。
   (d) 也许正在下着雪，也许在零度以下。
   (e) 如果气温在零度以下，则下着雪。
   (f) 也许气温字零度以下，也许下着雪；但如果在零度以下，就没有下雪。
   (g) 气温在零度以下是下雪的充分必要条件。

7. 令 p 、q 为如下命题：

   p：你的车速超过每小时 104 公里。

   q：你街道一张超速罚单。

   用 p、q 和逻辑联结词写出下列命题：

   (a) 你的车速没有超过每小时 104 公里。
   (b) 你的车速超过每小时 104 公里，但没有街道超速罚单。
   (c) 如果你的车速超过每小时 104 公里，你将接到一张超速罚单。
   (d) 如果你的车速不超过每小时 104 公里，你就不会街道超速罚单。
   (e) 车速超过每小时 104 公里足以街道超速罚单。
   (f) 只要你接到一张超速罚单，你的车速就超过每小时 104 公里。

8. 令 p 、q 、r 为如下命题：p : 在这个地区发现过灰熊。

   q : 在详见小路上徒步旅行是安全的。

   r : 详见小路两旁的草莓成熟了。

   用 pqr 和逻辑联结词写出下列命题：

9. 判断下列个条件语句是真是假：

   (a) 如果 $1+1=2$，则 $2+2=5$。
   (b) 如果 $1+1=3$，则 $2+2=4$。
   (c) 如果 $1+1=3$，则 $2+2=5$。
   (d) 如果猴子会飞，那么 $1+1=3$。

10. 下列各语句，判断其中想表达的是兼或还是异或，说明理由。

    (a) 晚餐有咖啡或者茶。
    (b) 口令必须至少包含 3 个数字或至少 8 个字符长。
    (c) 这门课程的先修课程是数论课程或者密码课程。
    (d) 你可以用没有或者欧元支付。

11. 对下列各语句，说一说如果其中的联结词是兼或与异或时的含义。你认为语句想表达的是哪个或?

    (a) 要选修离散数学课，你必须已经选修了微积分或一门计算机的课程。
    (b) 当你从 Acme 汽车公司购买一部新车时，你就能得到 2000 美元先进折扣或 2% 的汽车贷款。
    (c) 两人套餐包括 A 栏中的两道菜或 B 栏中的三道菜。
    (d) 如果下雪超过 0.6 米或寒风指数低于-100，学校就停课。

12. 把下列语句写成“如果 p，那么 q”的形式。

    (a) 只要吹东北风，就会下雪。
    (b) 苹果树会开发，如果天暖持续一周。

3 命题逻辑的应用

3.1 引言

逻辑在数学、计算机科学和其他许多学科有着许多重要的应用。数学、自然科学以及自然语言中的语句通常不太准确，甚至有歧义。为了使用精确表达，可以将他们防疫成逻辑语言。例如，逻辑可用于软件和硬件规范（specification）描述，应为在开发前这些规范必须要准确。另外，命题逻辑及其规则可用于设计计算机电路、构造计算机程序、验证程序的正确性高一级构造专家系统。逻辑可用于分析和求解许多熟悉的谜题。基于逻辑规则的软件系统也已经开发出来，它能够自动构造每种类型证明。在后续章节中，我们将讨论命题逻辑的部分应用。

3.2 语句翻译

有许多理由要把语句翻译成由命题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）常用二义性。把语句翻译成复合命题可以消除二义性。注意，这样翻译时也许需要根据语句的含义做一些合理的假设。此外，一旦完成了从语句到逻辑表达式的翻译，我们就可以分析这些逻辑表达式以确定他们的真值，对他们进行操作，并用推理规则对它们进行推理。

为了解释把语句翻译成逻辑表达式的过程，考虑下面两个例子。例 1 怎样把下面的语句翻译成逻辑表达式?“你可以在校园访问因特网，仅当你主修计算机科学或者你不是新生”

解我们的办法是用命题变元表示语句中的每个成分，并找出他们之间合适的逻辑联结词。上述语句可以翻译为

$$a \to (c \vee \neg f)$$

3.3 系统规范说明

在描述样件系统和软件系统时，将自然语言语句翻译成逻辑表达式是很重要的一部分。系统和软件工程师根据子软语言描述的需求，生成精确而无二义性的系统规范说明，这些规范说明可作为系统开发的基础。例 3 说明了如何在这

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

一过程中使用复合命题。例 3 使用逻辑联结词表示规范说明“当文件系统已满时，不能够发送自动应答”。解系统规范说明应该是一致的，也就是说，系统规范说明不应该包含可能导致矛盾的相互冲突的需求。当规范说明不一致时，就无法开发出一个满足说有规范说明的系统。

3.4 布尔搜索

3.5 逻辑谜题

3.6 逻辑电路

4 命题等价式

4.1 引言

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”来指由命题变元通过逻辑运算形成的一个表达式，比如 $p \wedge q$ 。

我们就从根据可能的真值对复合命题进行分类开始讨论。

定义 1 一个真值永远为真的复合命题，称为永真式（tautology），也称为重言式。一个真值永远为假的复合命题称为矛盾式（contradiction）。既不是永真式又不是矛盾式的复合命题称为可能式（contingency）。

在数学推理中永真式和矛盾式往往很重要。下面的例 1 解释了这两类复合命题。

4.2 逻辑等价式

在所有可能的情况下都有相同真值的两个复合命题称为逻辑等价的。我们也可以如下定义这一概念。

定义 2 如果 $p \leftrightarrow q$ 是永真式，则复合命题 p 和 q 称为逻辑等价的。用记号 $p \equiv q$ 表示 p 和 q 是逻辑等价的。

判断两个复合命题是否等价的方式之一是使用真值表。特别地，复合命题 p 和 q 是等价的当且仅当他们真值表的两列完全一致。例 2 说明了用这个方法建立了一个非常重要且很有用的逻辑等价式，即 $\neg(p \vee q)$ 和 $\neg p \wedge \neg q$ 等价。这个逻辑等价式是德-摩根律之一。

例 2 证明 $\neg(p \vee q)$ 和 $\neg p \wedge \neg q$ 是等价的。

表给出了若干重要的等价式。在这些等价关系中，T 表示永远为真的复合命题，F 表示永远为假的复合命题。

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

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

4.3 德-摩根律的运用

4.4 构造新的逻辑等价式

4.5 命题的可满足性

一个复合命题称为可满足的，如果存在一个对其变元的真值赋值使其为真。当不存在这样的赋值时，即当复合命题对所有变元的真值赋值都是假的，则复合命题是不可满足的。

当我们找到一个特定的使得复合命题为真的真值赋值时，就证明了它是可满足的。这样的一个赋值称为这个特定的可满足性问题的一个解。可是，要证明一个复合命题是不可满足的，我们需要证明每一组变元的真值赋值都使其为假。尽管我们总是可以用真值表来确定一个复合命题是否是可满足的，但通常有更有效的方法，如例 9 所示。

例 9 试确定下列复合命题是否可满足：$(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p)$，$(p \vee q \vee r) \wedge (\neg p \vee \neg q \neg r)$，以及 $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p) \wedge (p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$。

4.6 可满足性的应用

在不同领域（如机器人学、软件测试、计算机辅助设计、机器视觉、集成电路设计、计算机网络以及遗传学）中的许多问题都可以用命题可满足性建立模型。

4.7 可满足性问题求解

真值表可以用于判断复合命题是否为可满足的，或者等价地，其否定是否为永真式。这个问题对于只含有少量变量的复合命题而言可以通过手动来完成，但当变量数目增多时，就变得不切实际了。

5 谓词和量词

5.1 引言

本节我们将介绍一种表达能力更强的逻辑，即谓词逻辑。我们将看到谓词逻辑如何用来表达数学和计算机科学中各种语句的意义，并允许我们推理和探索对象之间的关系。为了理解为此逻辑，我们首先需要介绍为此的概念。之后，我们将介绍量词的概念，它可以让我们对这样的语句进行推理：某一性质对于某一类型的所有对象均成立，存在一个对象使得某一特性成立。

5.2 谓词

在数学断言，计算机程序以及系统规格说明中经常可以看到含有变量的语句，比如

$$“x > 3”, “x = y + 3”, “x + y = z”$$

和

$$“\quad x \qquad ”$$

以及

$$“\quad x \quad ”$$

当这些变量值为指定时，这些语句即不为真也不为假。本节我们将讨论从这种语句中生成命题的方式。

语句”x 大于 3” 有两部分。第一部分即变量 x 是语句的主语。第二部分表明语句的主语具有的一个性质。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 $P(x)$ 也可以说成命题函数 P 在 x 的取值。一旦给变量 x 赋一个值，语句 $P(x)$ 就称为命题并具有真值。

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 元祖 $(x_1, x_2, \cdots, x_n)$ 的值，P 也称为 n 位谓词或 n 元谓词。

5.3 量词

当命题函数中的变量均被赋值时，所得到语句就变成举要真值的命题。可是，还有领完一种称为量化的重要方式也可以熊命题函数生成一个命题。量化表示在何种程度上谓词对于一定范围的个体成立。在自然语言中，所有、某些、许多、没有，以及少量这些词都可以用在量化上。这里我们集中讨论两类量化：全程量化，它告诉我们一个谓词在所考虑范围内对每一个个体都为真；存在量化，它告诉我们一个谓词对所考虑范围内的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。

定义 1 $P(x)$ 的全称量化是语句

$$“P(x)\ x \qquad ”$$

用 $\forall x P(x)$ 表示 $P(x)$ 的全程量化，其中 $\forall$ 称为全称量词。

定义 2 $P(x)$ 的存在量化是命题

$$\text{“} \qquad x \ \ P(x)\text{”}$$

我们用符号 $\exists x P(x)$ 表示 $P(x)$ 的存在量化，其中 $\exists$ 称为存在量词。
另一个重要的量词是唯一性量词。

5.4 约束论域的量词

在要限定一个量词的论域时经常采用简写的表示法。在这个表示法里，办理必须满足的条件直接放在量词的后面。

5.5 量词的优先级

量词 $\forall$ 和 $\exists$ 比命题演算中的所有逻辑运算符都具有更高的优先级。

5.6 变量绑定

当量词作用于变量 x 时，我们所此变量的这次出现为约束的。一个变量的出现被称为自由的，如果没有被量词约束或设置为等于某一特定值。

5.7 涉及量词的逻辑等价式

5.8 量化表达式的否定

5.9 语句到逻辑表达式的翻译

5.10 系统规范说明中量词的使用

5.11 逻辑程序设计

6 嵌套量词

6.1 引言

嵌套量词，即一个量词出现在另一个量词的作用域内。

$$\forall x \forall y (x + y = y + x)$$

$$\forall x \forall y ((x > 0) \land (y < 0) \to (xy < 0))$$

6.2 理解涉及嵌套量词的语句

6.3 量词的顺序

6.4 数学语句到嵌套量词语句的翻译

6.5 嵌套量词到自然语言的翻译

6.6 汉语到逻辑表达式的翻译

6.7 嵌套量词的否定

7 推理规则

7.1 引言

本章后一部分我们将学习证明。数学中的证明是建立数学命题真实性的有效论证。所谓的论证（argument），是指一连串的命题并以结论为最后的命题。所谓的有效性（valid），是指结论或论证的最后一个命题必须根据论证过程前面的命题或前提（premise）的真实性推出。也就是说，一个论证是有效的当且仅当不可能出现所有前提为真而结论为假的情况。为从已知命题中推出新的命题，我们应用推理规则，这是构造有效论证的模板。推理规则是建立命题真实性的基本工具。

7.2 命题逻辑的有效论证

定义 1 命题逻辑中的一个论证是一连串的命题。除了论证中最后一个命题外都叫作前提，最后那个命题叫结论。一个论证是有效的，如果它的所有前提为真蕴涵着结论为真。

命题逻辑中的论证形式是一连串涉及命题变元的复合命题。无论用什么特定命题替换其中的命题变元，如果前提均为真，则称为该论证形式是有效的。

7.3 命题逻辑的推理规则

我们可以先建立一些相对简单的论证形式（称为推理规则）的有效性。