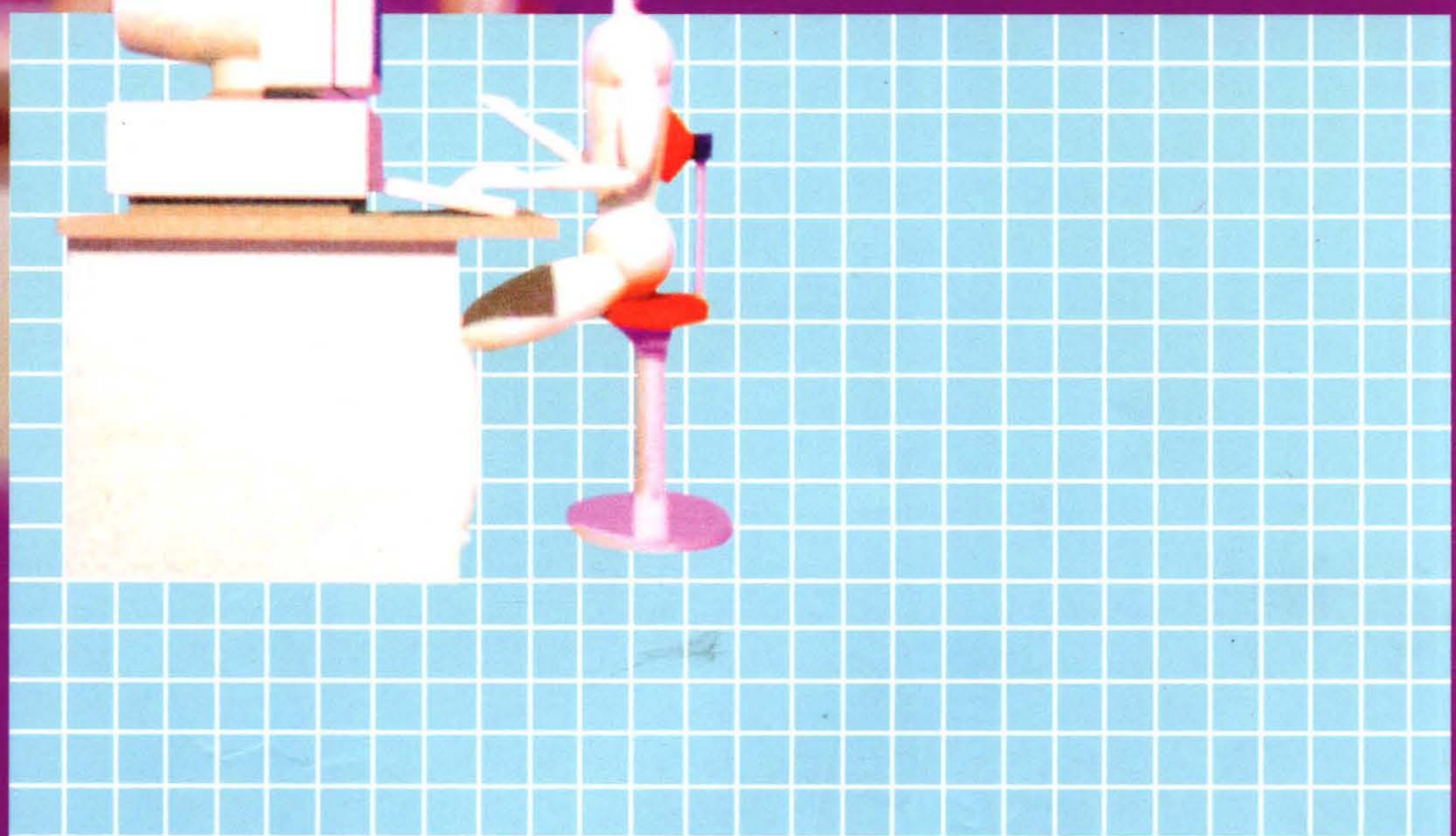
朱保平　编



数理逻辑及其应用

北京理工大学出版社

数理逻辑及其应用

朱保平　编

北京理工大学出版社

内 容 简 介

数理逻辑与计算机科学有着密切的关系，是计算机科学的重要理论基础，本书介绍了数理逻辑的基本概念、方法和理论，并且介绍了命题演算、谓词演算等在计算机中的应用。

全书共九章，前三章介绍了命题演算及非标准逻辑，包括命题、联结词、推理形式、模态逻辑和多值逻辑；第四、第五章介绍了谓词演算，包括谓词、个体、推理理论；后三章介绍了谓词演算的应用，包括事物的结构化表示，归结反演系统、基于规则的演绎系统；最后一章介绍了可计算机理论的一些基本知识。

图书在版编目(CIP)数据

数理逻辑及其应用/朱保平编. —北京：北京理工大学出版社，1998.10

ISBN 7-81013-749-2

Ⅰ. 数… Ⅱ. 朱… Ⅲ. 数理逻辑 Ⅳ. 0141

中国版本图书馆 CIP 数据核字(98)第 21313 号

责任印制：刘季昌 责任校对：陈玉梅

北京理工大学出版社出版发行

(北京市海淀区白石桥路 7 号)

邮政编码 100081 电话(010)68912824

各地新华书店经售

北京地质印刷厂印刷

*

787 毫米×1092 毫米 16 开本 8.75 印张 204 千字

1998 年 10 月第 1 版 1998 年 10 月 1 次印刷

印数:1-4000 册 定价:12.50 元

目　录

绪　论

数理逻辑是形式逻辑与数学相结合的产物，它主要是研究推理的科学，特别是数学中推理的科学，它是采用数学符号化的方法，给出推理规则来建立推理体系，进而讨论推理体系的一致性、可靠性和完备性。具体地讲是研究推理中前提和结论间的形式关系，这种形式关系是由作为前提和结论的命题的逻辑形式决定的，为研究前题和结论间的形式关系需要构造形式系统，这样的形式系统称为逻辑演算。

数理逻辑的研究内容概括地讲是两个演算加上四论，两个演算为命题演算和谓词演算；四论为递归论、证明论、模型论、公理集合论。其中命题演算和谓词演算是四论的共同基础。命题演算、谓词演算和递归函数本书将重点介绍，下面对其中的四论加以简单的说明，以便对数理逻辑有一个总的了解。

(1) 递归论(可计算性理论)：它为能行可计算函数找出各种理论上的、精确化的、严密化的类比物。

递归函数可以采用图灵机来实现，递归函数为Turing可计算函数，反过来Turing可计算函数必为递归函数。

(2) 证明论：研究数学理论系统的相容性。

(3) 模型论：主要对各种数学理论系统建立模型并研究各种模型间的关系及模型和数学系统之间的关系。

(4) 公理集合论：在消除已知集合论悖论的情况下用公理的方法把有关集合的理论发展下去。

所谓悖论是指自相矛盾的语句。著名的如罗素悖论。

如理发师问题，一理发师称“我只给不给自己理发的人理发”。

如果他不给自己理发，则他就应该给自己理发；反之，如果他给自己理发，则他就不应该给自己理发。同样如＂我正在说谎＂、＂黑板上这句话是假的＂均属于自身相关的悖论。

一般地，集合本身不能成为它自己的元素。如$\{a\}\notin\{a\}$，但有些集合本身可以成为它自己的元素，如概念的集合，因为它本身是一个概念，因此这个集合可以是它自己一个元素。因此$A\in A$和$A\notin A$均为谓词，可以用来定义集合。

如下面的罗素悖论：

设论述域是所有集合的集合，并定义S为下述集合

$$S=\{A \mid A\notin A\}$$

这样，S是不以自身为元素的全体集合的集合，现在的问题为“S是不是它自身的元素？”

假设S不是它自己的元素，即$S\notin S$，则S满足谓词$A\notin A$，而谓词定义了S，所以$S\in S$。另一方面，如果$S\in S$，则S必须满足定义S的谓词，所以，$S\notin S$。

如此导出了一个悖论：既非$S\in S$，又非$S\notin S$。

由上所述，在定义集合时，应直接或简接地限制集合成为它自己的元素，以避免悖论的产生。

数理逻辑和计算机科学有着十分密切的关系，无论是数字电子计算机雏形的图灵机，还是数字电路的布尔代数，以及作为程序设计工具的语言、程序设计方法学、关系数据库、知识库、编译方法、人工智能等领域均离不开数理逻辑。同时，由于两者的相互渗透推动了数理逻辑的发展。因此学好数理逻辑对于计算机科学理论的研究有重要的作用。

第一章　命题演算基础

1.1　命题和联结词

1.1.1　命题

定义：凡是可以判断真假的陈述句均称为命题。

命题具有两个特征，首先命题应是一个陈述句，感叹句、疑问句、祁使句等均不是命题；

其次这个陈述句所表达的内容可决定真或假且真假不可兼，即它应有真假性。

如果一命题取为真，则说明该命题的真值为真，用 T表示真；如果一命题取为假，则说明该命题真值为假，用 F表示为假。

下面举例说明命题的概念：

（1）数理逻辑是计算机科学中的一门必修课。

它是一个陈述句，可决定其真值为 T，所以为命题。

（2）请给我一张纸！

它不是陈述句，不是命题。

（3）火星上有生物。

它是陈述句，从本质而言，它具有真假性，不过当今尚不知其是真还是假，随着科学技术的发展一定能够判断其真假，所以是命题。

（4）我正在说谎。

悖论，虽其为陈述句，但其不能判断其真假，非命题。

（5）如果A和B为偶数，则$A+B$亦为偶数。

它是陈述句，可决定其真假，所以为命题。

（6）今天天气很凉快。

它是陈述句，其真假可根据个人的感觉来决定是 T或 F，所以是命题。

（7）$X+Y=Z$。

虽其为陈述句，但不能判断其真假，非命题。

（8）如果2+3=6，则3+7=9。

它是陈述句，其值为 T，是命题。

（9）1+1=3吗？

疑问句，不为陈述句，非命题。

(10) How beautiful flower it is !

感叹句，不为陈述句，非命题。

命题具有两种类型，原子命题和复合命题。

(1) 第一种类型是不可剖开或分解为更简单命题的命题称为原子命题。

如：＂雪是白的＂、＂2为质数＂、＂南京是江苏省的省会＂等就是原子命题。

(2) 第二种类型是由成分命题利用联结词构成的命题称为复合命题。

如：

＂1＋1＝2＂且＂雪是白的＂；

＂如果一个三角形有三条边相等，则该三角形为等边三角形＂等就是复合命题，其中语句中的＂和＂、＂且＂、“如果……则……”等就是联结词。

注意有时有些语句看似复合命题，但实际上为原子命题。

如语句＂TOM和JOHN是兄弟＂就不能分解为＂TOM是兄弟＂和＂JHON是兄弟＂，因为某一个人不能成为兄弟，故应把它理解为原子命题。

在绪论中，介绍了数理逻辑是研究前提和结论间的形式关系，而不研究具体的内容。为此采用数学方法将命题符号化亦称为形式化方法是十分重要的，约定用大写字母表示命题。

例如：P：表示＂1＋1＝2＂；

Q：表示＂南京是江苏省的省会＂；

R：表示“2为质数”。

定义：如果当P表示任意命题时，P就称为命题变元。

注意命题变元和命题是两个不同的概念。

命题指具体的陈述句有确定的真值。

命题变元没有确定的真值，只有当代以具体的命题时才能确定它的真值。换言之，命题变元是以真假为变域的变元，用P，Q，R等表示命题变元。

1.1.2 联结词

下面介绍五个常用的联结词：

$\neg$、$\wedge$、$\vee$、$\rightarrow$、$\leftrightarrow$

一、否定词$\neg$

否定词是一个一元联结词。

利用该联结词可利用成分命题P构成复合命题$\neg P$，读为非P。

日常语言中的＂非＂、＂不＂、＂并非＂等均表示逻辑非。

非P的真假和P的真假关系定义如下：

$\neg P$为真当且仅当P为假

其真值表如图1.1-1

P	$\neg P$
T	F
F	T

图 1.1-1

例 1：P：今天是星期天。

$\neg P$：今天不是星期天。

例 2：P：＂4为质数＂

$\neg P$：＂4不为质数＂。

二、合取词∧

合取词＂∧＂是一个二元联结词。

利用该联结词可将成分命题 P和 Q构成复合命题 $P\wedge Q$，读为 P合取 Q。其中 $P\wedge Q$称为合取式，P和 Q称为合取项。

日常语言中的＂且＂、＂与＂、＂并且＂、＂但＂等均可表示为合取。

P合取 Q的真假和 P，Q的真假关系定义如下：

$P\wedge Q$真当且仅当 P和 Q均真

其真值表如图1.1－2：

例 1：　P：＂1＋1＝2＂；

　　Q：＂雪是白的＂。

则　$P\wedge Q$：＂表示1＋1＝2且雪是白的＂。

例 2：这台计算机质量很好但很贵。

令：　P表示＂这台计算机质量很好＂；

　　Q表示＂这台计算机很贵＂。

则原句译为 $P\wedge Q$。

P	Q	$P\wedge Q$
T	T	T
T	F	F
F	T	F
F	F	F

图 1.1-2

三、析取词∨

析取词 ＂∨＂是一个二元联结词。

利用成分命题 P和 Q可构成复合命题 $P\vee Q$，读为 P析取 Q。其中 $P\vee Q$称为析取式，P和 Q称为析取项。

日常语言中的＂或＂等可用析取词来表示。

P析取 Q的真假和 P，Q的真假关系定义如下：

$P\vee Q$为假当且仅当 P和 Q均假

其真值表如图1.1-3。

P	Q	$P\vee Q$
T	T	T
T	F	T
F	T	T
F	F	F

图 1.1-3

例 1：　P：＂今天下雨＂；

　　Q：＂今天下雪＂。

　$P\vee Q$：表示今天下雨或今天下雪。

例 2：　P表示＂今天我去书店购书＂；

　　Q表示＂今天我在家学电脑＂。

　$P\vee Q$：表示＂今天我去书店购书或在家学电脑＂。

注意语言中＂或＂在现实生活中有可兼性和不可兼性，在数理逻辑中规定只有唯一的一种意思，即可兼的＂或＂。

四、蕴含词→

蕴含词＂→＂是一个二元联结词。

利用成分命题 P和 Q可构成复合命题 $P\rightarrow Q$，读为 P蕴含 Q，其中 $P\rightarrow Q$称为蕴含式，P称为蕴含前件，Q称为蕴含后件，蕴含词亦可以用 " ⊃ " 表示。

日常语言中的 " 如果...则... " 、 " 只要...就... " 等均可表示为蕴含词。

P蕴含 Q的真假和 P，Q的真假关系定义如下：

$P\rightarrow Q$假当且仅当 P真 Q假

其真值表如图1.1-4。

注意在现实生活中，用如果...则...连系起来的蕴含式，其前后件必须是相关的，不能把不相干的两个语句连在一起，如下面例2中语句在生活中是没有意义的，但在数理逻辑中由于只考虑两者之间的形式关系，而不涉及其内容，因此可以把任意的两个命题构成一蕴含式。

P	Q	$P\rightarrow Q$
T	T	T
T	F	F
F	T	T
F	F	T

图 1.1-4

例 1：P：" 天下雪 " ；

Q：" 天下雨 "

R：" 体育比赛照常进行 " 。

$(\neg P\vee\neg Q)\rightarrow R$：表示 " 只要天不下雪或不下雨，体育比赛就照常进行 " 。

例 2：P：" 1+1=3 " ；

Q：" 雪是黑的 " ；

R：" 2+2=6 " ；

$P\rightarrow(Q\wedge R)$表示 " 如果1+1=3，则雪是黑的且2+2=6 " 。

从真值表可看出，当蕴含前件 P取 F值时，不管其后件 Q取 T或 F，蕴含式 $P\rightarrow Q$总取真 T，故复合命题 " 1+1=3，则雪是黑的且2+2=6 " 之值为 T。也就是说，在形式推理中只要前件为假，就可推出任何命题，而此推理过程是正确的。

五、等价词↔

等价词 " ↔ " 是一个二元联结词。

利用成分命题 P和 Q可以构成复合命题 $P\leftrightarrow Q$，读为 P等价于 Q。其中 $P\leftrightarrow Q$称为等价式，P称为等价式左端，Q称为等价式右端。等价词有时亦用 " ≡ " 来表示。

日常语言中的 " 当且仅当 " 等可用等价词来表示。

P等价于 Q的真假关系和 P，Q的真假关系定义如下：

$P\leftrightarrow Q$为真当且仅当 P与 Q均假

其真值表如图1.1-5。

例 1：令 P：表示“三角形 $A\ B\ C$为等腰三角形”；

Q：表示“三角形 $A\ B\ C$有两条边相等”；

则：$P\leftrightarrow Q$：表示“一个三角形 $A\ B\ C$为等腰三角形当且仅当三角形 $A\ B\ C$有两条边相等”。

例 2：若不是他生病或出差，我是不会同意他不去

P	Q	$P\leftrightarrow Q$
T	T	T
T	F	F
F	T	F
F	F	T

图 1.1-5

上课的。

令 P表示 " 他生病 " ；

Q表示 " 他出差 " ；

R表示 " 我同意他不去上课 " ；

则 $\neg(P\vee Q)\leftrightarrow\neg R$就表示为 " 若不是他生病或出差，我是不会同意他不去上课的 " 。

上面介绍了五个常用的真值联结词，其实真值联结词还很多。为了能更好表达其它真值联结词引进真值函项的概念，用真值函项的概念可以定义一元二元甚至n元真值联结词。

定义：以真假为定义域并且以真假为值域的函数称为真值函项。

有了真值函项的概念就可以用它来表达联结词。

一元联结词有一个命题变项 P，它取值只有真和假两种，可定义四个不同的一元联结词 f_0， f_1， f_2， f_3，或称为真值函项 $f_i(P)$。

其真假关系可用图1.1-6表示：

P	$f_0(P)$	$f_1(P)$	$f_2(P)$	$f_3(P)$
T	T	T	F	F
F	T	F	T	F

图 1.1-6

从图可看出：

$f_0(P)$表示永真；

$f_1(P)$表示恒等；

$f_2(P)$表示否定，即$\neg P$；

$f_3(P)$表示永假联结词。

同理，二元真值联结词有16个，如图1.1-7。

P	Q	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
T	T	T	F	T	T	T	F	F	F	T	T	T	T	F	F	F	F
T	F	T	T	F	T	T	F	T	T	F	F	T	F	T	F	F	F
F	T	T	T	T	F	T	T	F	T	F	T	F	F	F	T	F	F
F	F	T	T	T	T	F	T	T	F	T	F	F	F	F	F	T	F

图 1.1-7

从图可看出：

f_4 为析取∨；

f_{11} 为合取∧；

f_2 为蕴含→；

f_8 为等价↔；

f_1为与非： $P\uparrow Q=\neg(P\wedge Q)$

f_{14} 为或非： $P\downarrow Q=\neg(P\vee Q)$

f_7为异或： $P\oplus Q=\neg(P\leftrightarrow Q)$

同样可以利用真值函项的概念来定义三元真值联结词，三元联结词共有256个，图1.1-8为一个三元联结词真值关系的**定义**：

一般地，对于 n 个命题变元 P_1， P_2，…， P_n，

P	Q	R	$f(P, Q, R)$
T	T	T	T
T	T	F	T
T	F	T	T
F	T	T	T
T	F	F	F
F	T	F	F
F	F	T	F
F	F	F	F

图 1.1-8

每个 P有两种取值：真或假，从而 P_1， P_2，… P_n共有2^n种取值，于是相应的真值联结词有2^{2^n}个。不难验证，任何命题公式均可由上面定义的五个联结词来表示。

1.1.3 合式公式

有了命题变元和联结词的概念，就可以利用括号来讨论命题演算的合式公式，其中括号可用来区别联结词的运算优先次序。

合式公式为如下定义的式子，简称公式。

(1) 任何命题变元均是公式。

(2) 如果 P为公式，则$\neg P$为公式。

(3) 如果 P， Q为公式，则$(P\vee Q)$，$(P\wedge Q)$，$(P\rightarrow Q)$，$(P\leftrightarrow Q)$为公式。

(4) 当且仅当经过有限次使用(1)、(2)、(3)所组成的符号串才能是公式，否则不是公式。

例如： P，$(P\vee Q)$，$((P\vee Q)\wedge R)$为公式。

$P\vee Q\wedge$， $P\rightarrow Q)\leftrightarrow R$等就不为公式。

但为了方便起见，我们采用省略一些括号，保留一些括号来描述合式公式。

如：$((P\vee Q)\wedge R)$省写为$(P\vee Q)\wedge R$

$(P\rightarrow(Q\leftrightarrow R))$省写为 $P\rightarrow(Q\leftrightarrow R)$等等。

例：试说明$(P\vee Q)\leftrightarrow((\neg P\rightarrow Q)\wedge R)$为公式。

解：① P为命题公式　　根据(1)

② Q为命题公式　　根据(1)

③ R为命题公式　　根据(1)

④ $(P\vee Q)$为公式　　由①、②根据(3)

⑤$\neg P$为公式　　根据(2)

⑥$(\neg P\rightarrow Q)$　　由⑤、②根据(3)

⑦$((\neg P\rightarrow Q)\wedge R)$　由⑥、③根据(3)

⑧$(P\vee Q)\leftrightarrow((\neg P\rightarrow Q)\wedge R)$为公式　由④、⑦根据(3)

定义：若公式α中有 n 个不同的命题变元，就说α是 n元公式。

如公式$(P\vee Q)\leftrightarrow((\neg P\rightarrow Q)\wedge R)$中含有 P、 Q、 R三个命题变元，因此它就为三元公式。

下面举一些例子说明怎么把命题符号化成公式，注意在语句符号化时，一定要分解至原子命题，而不能把某个复合问题直接用命题变元 P来表示，如此不能完整表达语句的意思，也不便于计算机完整处理语句及其产生的知识。

例 1：如果三角形有两条边相等，则该三角形一定为等腰三角形。

解：

令 P表示＂三角形有两条边相等＂；

Q表示＂三角形为等腰三角形＂；

则原句可表示为： $P\rightarrow Q$。

例 2：只有努力学习、认真复习，方能取得好成绩。

解：

令 P表示 " 努力学习 " ；

Q表示 " 认真复习 " ；

R表示 " 取得好成绩 " ；

则原句可表示为： $R \supset (P \wedge Q)$，

而非译为$(P \wedge Q) \supset R$。

注意该语句符号化时，一定要考虑条件的必要性和充分性。

例 3：今天体育场有球赛，除非天下雨。

解：

令 P表示 " 体育场有球赛 " ；

Q表示 " 天下雨 " ；

则原句可表示为： $\neg Q \supset P$。

例 4：TOM总在计算中心上网，除非计算中心不开门或TOM生病。

解：令 P表示 " TOM在计算中心上网 " ；

Q表示 " 计算中心开门 " ；

R表示 " TOM生病 " ；

则原句可表示为：$(\neg Q \vee R) \rightarrow \neg P$

亦可译为： $P \rightarrow (Q \wedge \neg R)$

例 5： D是 A和 B的公倍数当且仅当 A和 B都能除尽 D。

解：

令 P表示“A能除尽 D”；

Q表示“ B能除尽 D”；

R表示“ D是 A和 B的公倍数”；

则原句可表示为： $R \leftrightarrow (P \wedge Q)$。

例 6：南京不是中国的首都。

解：

令 P表示“南京是中国的首都”；

则原句可译为$\neg P$。

例 7：设 P表示今天很冷；

Q表示正在下雪。

试将下列语句行号化：

如果正在下雪，那么今天很冷。

今天很冷当且仅当正在下雪。

正在下雪的必要条件是今天很冷。

解：上述语句译为： $Q \supset P$

$P \leftrightarrow Q$

$Q \supset P$

例 8：已知三个命题：

P：今晚我在家看电视节目。

Q：今晚我去音乐厅听音乐会。

R：今晚我在家看电视节目或去音乐厅听音乐会。

试问 $P\vee Q$和 R是否表达同一命题？请用真值表说明之。

解： R和 $P\vee Q$不表达同一命题。

可有真值表表示如图1.1-9。

P	Q	R	$P\vee Q$
T	T	F	T
T	F	T	T
F	T	T	T
F	F	F	F

图 1.1-9

由真值表可看出当 P和 Q同取 T时，R取 F，但 $P\vee Q$取 T，也就是说今晚我去音乐厅听音乐会又在家看电视，显然这是不可能的。

故　R应表示为 $R=(\neg P\wedge Q)\vee(P\wedge\neg Q)$

1.1.4　关于公式的表示法

数理逻辑对语句的符号化是为了便于用计算机来处理逻辑推理，但不同的表达方式对计算机处理算法会产生不同的效率。为了提高计算机处理符号的效率，在符号化公式时有时采用波兰表示法、逆波兰表示法等。

一、计算机对括号的处理

在命题演算中合式公式的定义是采用联结词的中缀表示，且用括号来区分联结词的运算次序。计算机处理中缀表示法表达的公式，采用从左向右及从右向左来回扫描公式找出括号的匹配关系才能计算公式的真值。

如计算公式$(P\wedge(Q\rightarrow R))\vee(S\leftrightarrow T)$的真值应采用如下方法：

先从左向右扫描公式，当发现第一个右括号时，便返回至最近的左括号处，计算部分公式$(Q\rightarrow R)$的值，重复此过程直到计算结束。此过程如下：

(1)　$(Q\rightarrow R)$

(2)　$(P\wedge(1))$

(3)　$(R\leftrightarrow T)$

(4)　$(2)\vee(3)$

从以上处理过程来看，要处理一公式需来回扫描公式才能计算一公式的值，从计算机的优化角度来讲这种处理方式增加了存贮空间和运行时间。因此中缀表示法不是计算机处理符号的最好表达方式，为此我们引进波兰表示法或逆波兰表示法，以提高计算机的处理效率。

二、逆波兰表示法

1. 逆波兰表示法的优点

逆波兰表示方法是采用命题变元在前，联结词在后的表示方法。

其有下面两个优点：

(1) 无括号，形式简洁。

(2) 联结词的顺序就是它的运算顺序。

如上述中缀表示的公式$((P\wedge(Q\rightarrow R))\vee(S\leftrightarrow T))$

可以表示为逆波兰表示式如下：

$$P\ Q\ R\rightarrow\wedge\ S\ T\leftrightarrow\vee$$

又如公式 $(P\wedge(Q\vee R))$

逆波兰式为：$P\ Q\ R\vee\wedge$

2. 中缀表达式转换为逆波兰式的算法

首先规定联结词的运算次序：

(1) ¬优先于∨、∧；

(2) ∨、∧优先于→、↔；

(3) 左边的∨、∧优先于右边的∨、∧；

(4) 左边的→、↔优先于右边的→、↔；

(5) 括号内的公式先运算；

(6) 左括号优先于其左边式子中的联结词，小于其右边的联结词；

(7) 左右括号的优先数相同。

(8) 右括号 ") " 的优先级最小，其不进联结词栈，也不进输出区。

据上讨论知运算符间存在三种关系：

(1) 大于关系，用>表示；

(2) 小于关系，用<表示；

(3) 等于关系，用＝表示。

其优先关系可列表如图1.1-10：

	¬	∨	∧	→	↔	(	)
¬	>	>	>	>	>	<	>
∨	<	>	>	>	>	<	>
∧	<	>	>	>	>	<	>
→	<	<	<	>	>	<	>
↔	<	<	<	<	>	<	>
(	<	<	<	<	<	<	=
)	>	>	>	>	>		>

图 1.1-10

把中缀表达公式转换成逆波兰式时，由于联结词间存在优先关系，必须设立一个联结词栈(称为ω栈)存放暂时不能处理的联结词。

根据联结词的优先关系生成逆波兰表达式的算法如下：

(1) 从左到右扫描中缀表示的命题公式，有以下几种情形：

① 当前符号为命题变元，立即输出该命题变元，继续本步；

② 当前符号是联结词，则转(2)；

③ 当前符号是左括号“(”，则转(3)；

④ 当前符号是右括号“)”，则转(4)；

⑤ 当中缀表达式扫描完毕，则转(5)；

(2) 检查联结词栈ω中有无联结词，若ω栈为空，则当前联结词进入ω栈，转(1)。若ω栈栈顶不空，则比较当前联结词与ω栈栈顶联结词的优先关系。