计算中的基本理论与方法

周培德　编著

北京理工大学出版社

计算中的基本理论与方法

周培德　编著

北京理工大学出版社

内 容 提 要

本书系统地介绍了计算机科学中的基本理论与方法。全书分10章，包括：预备知识、有穷自动机与正规语言、图灵机、递归函数、计算复杂性、程序正确性证明、定理的机器证明、非标准逻辑、推理方法、按自然法则计算等。该书概括了计算机专业研究生必需的绝大部分基础理论知识，内容丰富，由浅入深，可读性较好。

本书是高等院校计算机专业研究生的教材，也可作为相应专业科技人员的参考书。

图书在版编目（CIP）数据

计算中的基本理论与方法/周培德编著．—北京：北京理工大学出版社，1997.9
ISBN 7-81045-301-7

Ⅰ．计… Ⅱ．周… Ⅲ．①电子计算机-算法理论②电子计算机-计算方法 Ⅳ．TP301.6

中国版本图书馆CIP数据核字（97）第12935号

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
*
787×1092毫米 16开本 18.25印张 443千字
1997年9月第一版 1997年9月第一次印刷
印数：1—2000册 定价：21.00元

前　言

计算中的理论（即计算理论）是指计算机科学中的基础理论。它研究的范围涉及到计算机科学的许多分支，为这些分支提供必要的基本理论和方法。

本书是在多次给北京理工大学计算机系研究生授课讲稿的基础上，参照 IEEE－83 教程与 ACM/IEEE－CS〈计算 1991 教程〉大纲编著而成。

众所周知，利用建立在图灵模型基础上的冯·诺依曼体系结构计算机求解实际问题，一般需要经过下列几个阶段：表述实际问题；研制计算模型并论证该模型的可计算性；设计算法；证明算法的正确性并分析其复杂性；实现算法并验证结果。本书将涉及上述第一、第二与第四阶段。

从计算机模型的角度来看已经存在非冯模型，但依据非冯模型所研制的计算机至今并未得到广泛应用。依据冯模型与依据非冯模型设计的计算机在求解问题的过程、效果等诸多方面截然不同，这是值得注意的。但由于后者的研究还不成熟，因此本书只是在第十章作简要的介绍。

标准逻辑（即经典逻辑）及其扩充的非标准逻辑是表述某些知识（问题）的较好形式，在此形式下，依据一定的规则可以推出一些新的知识。这是知识表述及获得新知识的一种途径，我们将在第八、九章中给予介绍。当然，并非所有知识（问题）都适于用逻辑形式来表述。因此人们还要利用其它的数学工具来描述知识，比如图、集合、代数等，从而得到相应的计算模型并研究该计算模型的可计算性。当然本书并不研究某个具体模型的可计算性，而是以图灵机、递归函数等模型为工具研究问题类的可计算性。也就是说，某个问题类如果能够被图灵机计算，或表示为递归函数，那么该问题类就是可计算的。如果某个问题类不能被图灵机计算，或不能表示为递归函数，那么该问题类是否可计算？这是非经典可计算性理论研究的内容。本书第二、三、四章介绍有穷自动机、图灵机与递归函数等模型及它们之间的关系，这属于经典可计算性范畴。算法设计方面的知识本书不作介绍，因为它已独立成为一个分支。第六章介绍与算法正确性密切相关的程序正确性的证明方法。另外，有些问题的解决可以转变成定理证明的问题，因此在第七章将介绍定理机器证明的几种方法。计算复杂性是计算理论中重要的内容之一，本书第五章着重介绍图灵机的计算复杂性及加速定理，至于计算复杂性的其它方面内容，例如 NP 完全性理论，下界理论等在此不作介绍。总之，本书是围绕解决问题的几个阶段与几种计算机可实现的途径及相关理论、方法展开论述的，重点是可计算性、计算复杂性、定理机器证明与非标准逻辑。

由于本书涉及的内容广泛，故而某些符号难以统一，但各章节所用符号都给出了说明。

北京航空航天大学何自强副教授仔细地审阅了书稿（一至九章），提出了许多宝贵意见；余荣老师为书稿做了大量的工作；许多研究生在学习本教材时进行了详细地推敲，提出了很好的建议。编者在此表示衷心的感谢。

由于编者水平有限，书中一定存在许多缺点和错误，恳请同行与读者批评指正。

编者　1997. 2

目　录

第一章　预备知识

本章介绍后继章节所需要的基本知识，其中包括字符串、字母表、语言、图、集合、关系、经典逻辑等。

§1-1　字符串、字母表和语言

“符号”是一个抽象的实体，比如几何学中的“点”和“线”以及数学中的“集合”等。一般来说，人们不给出这些抽象实体的形式定义，字母、数字是经常使用的符号。字符串（或字、串）是并置起来的有穷符号序列，如 a，b，c，5 是符号，而 $abc5$ 是字符串，一般用 W，X，Y，Z 等表示字符串。W 的长度（用 $|W|$ 表示）是该字符串中所含符号的个数，如字符串 $abc5$ 的长度为 4。若一个字符串是由零个字符组成，则称该字符串为空字符串，记作 ε，显然有 $|\varepsilon|=0$。

处在字符串顶前（或顶后）部的若干个符号称为该字符串的前（或后）缀，如在上例中，ε，a，ab，abc 和 $abc5$ 是字符串的前缀，而 ε，5，$c5$，$bc5$ 和 $abc5$ 是它的后缀。如果字符串的前缀（或后缀）不是字符串本身，则称该串为真前缀（或真后缀），上例中，除 $abc5$ 外，都是真前（或后）缀。

字符串 X 和 Y 的连接是 XY，即先写第一个字符串，接着写第二个字符串，当中没有空格。空字符串是连接操作的单位元素，即对每个字符串 X，有 $\varepsilon X=X\varepsilon=X$。

字母表是符号的一个有穷集合，语言 L 是符号选自某个有限字母表的字符串集合。空集 ϕ 没有元素，而由空字符串组成的集合 $\{\varepsilon\}$ 是有元素的，空集 ϕ 和 $\{\varepsilon\}$ 都是语言。由符号 0 和 1 构成的字母表 $\{0,1\}$ 上的回文（顺读和倒读都一样的字符串）的集合是一个无穷语言。注意，一个无穷符号集合上所有的回文不是一个语言，因为它的字符串不能由一个有限字母表建立起来。

用 Σ 表示字母表，Σ^* 表示的语言 L 是字母表 Σ 上所有字符串的集合。例如，若 $\Sigma=\{a\}$，则有 $\Sigma^*=\{\varepsilon,a,aa,aaa,\cdots\}$；若 $\Sigma=\{0,1\}$，则 $\Sigma^*=\{\varepsilon,0,1,00,01,10,11,000,\cdots\}$。

如果字符串 X，Y，Z，$W\in\Sigma^*$，满足 $XYZ=W$，则称 Y 是 W 的子字符串。如果 $Y\neq\varepsilon$ 且 $Y\neq W$，则称 Y 是 W 的真子字符串。设 X 是 Σ 上的字符串，且 $X=a_1a_2\cdots a_{n-1}a_n(a_i\in\Sigma,i=\overline{1,n})$，则称 $a_na_{n-1}\cdots a_2a_1$ 为 X 的反字，记为 X^R。

设 L_1、L_2 是 Σ 上的语言，定义语言 L 为 L_1 与 L_2 的连接：$L=L_1L_2=\{XY\mid X\in L_1,Y\in L_2\}$。

设 X 是 Σ 上的字符串，n 为非负整数，定义

$$\begin{cases}X^0=\varepsilon\\X^{n+1}=X^nX\end{cases}$$

即 X^n 是 n 个 X 的连接。

显然有

$$X^nX^m = X^{n+m} \quad (X \in \Sigma^*, n、m \text{ 为非负整数})$$
$$(X^n)^m = X^{nm} \quad (X \in \Sigma^*, n、m \text{ 为非负整数})$$

另外还有：

(1) $(L_1L_2)L_3 = L_1(L_2L_3)$

(2) $\{\varepsilon\}L = L\{\varepsilon\} = L$

(3) $\phi L = L\phi = \phi$

(4) $(L_1 \cup L_2)L_3 = L_1L_3 \cup L_2L_3$

$L_1(L_2 \cup L_3) = L_1L_2 \cup L_1L_3$

设 L 是 Σ 上的语言，n 为非负整数，定义

$$\begin{cases} L^0 = \{\varepsilon\} \\ L^{n+1} = L^nL \end{cases}$$

设 L 是字母表 Σ 上的语言，定义

$$L^* = \bigcup_{n=0}^{\infty} L^n$$

称 L^* 为 L 的 Kleene 闭包；定义 $L^+ = \bigcup_{n=1}^{\infty} L^n$ 为 L 的正闭包。

这就是说，L^* 是用 L 中的字符串（包括长度为 0 的字符串）连接成的字符串的全体，而 L^+ 是用 L 中的字符串（不包括长度为 0 的字符串）连接成的字符串的全体。

显然有

$$(L^*)^* = L^*$$
$$L^+ = LL^* = L^*L$$

例 1-1 设 $L=\{10,11\}$，$L_1=\{10,1\}$，$L_2=\{011,11\}$，则 $L_1L_2=\{10011,1011,111\}$；$L^*=\{10,11\}^*=\{\varepsilon,10,11,1010,1011,1110,1111,\cdots\}$；$L^+=\{10,11\}^*=\{10,11,1010,1011,1110,1111,\cdots\}$；$L_1 \cup L_2=\{10 \cup 011,1 \cup 011,10 \cup 11,1 \cup 11\}=\{10,011,1,11\}$。

§1-2 图 和 树

图是由有穷顶点集合 V 和边的集合 E 组成，记为 $G=(V,E)$。图中的一条路径是一个顶点序列 v_1，v_2，…，v_k，$k \geqslant 1$，使得对于每个 $i(1 \leqslant i < k)$，都有一条边 (v_i,v_{i+1}) 属于 E，这条路径的长度为 $k-1$（即路径上所含边或弧的条数）。如果 $v_1=v_k$，则称此路径为圈。长为 0 的路径叫空路，于是对于每个顶点 v_i，都存在一条从 v_i 到 v_i 的空路。

有向图是由有穷顶点集合 V 和弧的有序顶点对集合 E 组成，记为 $G=(V,E)$。用 $v_{i-1} \rightarrow v_i$（或 e_i）表示由 v_{i-1} 到 v_i 的弧，其中 v_{i-1}，v_i 分别是 e_i 的始点和终点。

有向图中的一条路径是一个顶点序列 v_1、v_2、…、v_k，$k \geqslant 1$，使得对于每个 $i(1 \leqslant i < k)$，$v_i \rightarrow v_{i+1}$（或 e_i）都是 G 中的一条弧，称该路径由 v_1 到 v_k，称 v_i 是 v_{i+1} 的前趋，v_{i+1} 是 v_i 的后继。v_1，v_k 分别称为该路径的始点和终点。

有向图中，以顶点 v 为始点的边的条数叫做 v 的出度，记为 $d^+(v)$；以顶点 v 为终点的边的条数称为 v 的入度，记为 $d^-(v)$。

设 $G=(V,E)$ 是有向图，$V' \subseteq V$，$E' \subseteq E$，E' 中边的端点全在 V' 中，则称 $G'=(V',E')$ 为 G 的一个子图。

设 $V'\subseteq V$，定义 $G(V')=(V',E')$，其中 $E'=\{e\in E|e$ 的两个端点属于 $V'\}$，称 $G(V')$ 为 V' 生成的子图。设 $E'\subseteq E$，定义 $G(E')=(V',E')$，其中 $V'=\{v\in V|v$ 与 E' 中的某条边关联$\}$，称 $G(E')$ 为 E' 生成的子图。

如果一条边的始点和终点相同，则称此边为一个环。如果两条边的始点与终点分别相同，则称它们是平行边。没有环也没有平行边的有向图叫简单有向图。

用关联矩阵可以表示图。设 $G=(V,E)$，其中 $V=\{v_1,v_2,\cdots,v_n\}$，$E=\{e_1,e_2,\cdots,e_m\}$，则 G 的关联矩阵为

$$D=\begin{pmatrix} d_{11} & d_{12} & \cdots & d_{1m} \\ \cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots \\ d_{n1} & d_{n2} & \cdots & d_{nm} \end{pmatrix}$$

其中

$$d_{ij}=\begin{cases} 1 & v_i \text{ 是 } e_j \text{ 的始点时} \\ -1 & v_i \text{ 是 } e_j \text{ 的终点时} \\ 0 & \text{其它} \end{cases}$$

关联矩阵可以唯一地决定有向图，当有向图无平行边时，则可表示如下：

(1) 邻接表：对于每个顶点 v，集合 $L(v)=\{u|(v,u)\in E\}$称为顶点 v 的邻接表，而全体集合$\{L(v)|v\in V\}$称为图 G 的邻接表。

(2) 邻接矩阵：给定 $G=(V,E)$，$V=\{v_1,v_2,\cdots,v_n\}$，则 G 的邻接矩阵为

$$A=\begin{pmatrix} a_{11}, & \cdots, & a_{1n} \\ \cdots\cdots & \cdots\cdots & \cdots\cdots \\ a_{n1}, & \cdots, & a_{nn} \end{pmatrix}$$

其中

$$a_{ij}=\begin{cases} 1 & (v_i,v_j)\in E \\ 0 & (v_i,v_j)\notin E \end{cases}$$

设 $G(V,E)$是有向图，I 为符号集合，f 是 E 到 I 的映射，则称(G,f)是边赋值有向图，f 称为赋值映射，对于 $e\in E$，$f(e)$称为边 e 的赋值。

边赋值有向图的各边赋值可标在各边的旁边，图 1-1 表示 $\Sigma=\{a,b\}$赋值有向图，其中 e_i 是边的名称，a，b 是边的赋值。在边赋值有向图(G,f)中，路径 $P=e_1e_2\cdots e_n$ 的赋值定义为 $f(P)=f(e_1)f(e_2)\cdots f(e_n)$，它是 Σ 上的一个字符串。

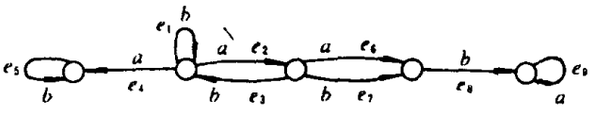

图 1-1　边赋值有向图

有向图的各边反向后得到的有向图，称为原图的反图，记为 G^R，G 和 G^R 的相应边有相同的赋值。P 是 G 的路径等价于 P^R 是 G^R 的路径。

无向图中，顶点没有出度与入度的概念，而有度的概念，顶点 v 的度数记为 $d(v)$，计算度数时，环（具有同一端点的边）应计算两次。无向图中的圈要求相邻的两条边不相同，因此无向图中不是环的圈至少有三条边。无向图的路径不再分始点与终点，只是说路径的两个端点。

无向图 G 中，若从 v_1 到 v_2 有路径，则称 v_1 和 v_2 是连通的。若对于 G 中每一对不同的顶点 v_i 和 v_j 都是连通的，则称 G 是连通图。无向图中的最大连通子图称为图 G 的连通分支。

忽略有向图中边的方向便得到无向图，此无向图称为原图的基础图。

基础图连通的有向图叫连通有向图。

如果有向图 G 中，任意两个顶点 u 和 v，都有从 u 到 v 的路径，也有从 v 到 u 的路径，则称 G 为强连通的。

具有下列性质的有向图称为树：

(1) 有一个称为根的顶点，它没有前趋，由它出发到每一个顶点都有一条路径；

(2) 除根以外，每个顶点都恰有一个前趋；

(3) 每个顶点的后继都被“自左”排序。

从无向图观点来看，连通的无圈的无向图叫树。

在树中，顶点的后继称为后裔，前趋称为先辈。既有先辈又有后裔的顶点，称为树的内部顶点，只有先辈而无后裔的顶点，称为树的叶。

§1-3 集合表示法和关系

集合是一群无重复的客体（它们称为集合的元素）。有穷集合可以通过在括号中列出其元素的方法加以规定，比如用 {0，1} 表示含两个元素 0 和 1 的集合。另外还可以表示成下述形式

$$\{x \mid p(x)\}$$

该形式表示使得 $p(x)$ 为真的那些 x 的集合，其中 $p(x)$ 是关于 x 的某个命题；形式 $\{A$ 中的 $x \mid p(x)\}$，表示集合 A 中使 $p(x)$ 为真的那些 x 的集合，这种表示法等价于形式 $\{x \mid p(x)$ 且 $x \in A\}$，例如，偶数可以表示为

$$\{i \mid i \text{ 是一个整数且存在整数 } j, \text{使 } i = 2j\}$$

集合通常有下列运算：设 A、B 为集合，则有：

(1) A 和 B 的并　$A \cup B = \{x \mid x$ 在 A 中或 x 在 B 中$\}$

(2) A 和 B 的交　$A \cap B = \{x \mid x$ 在 A 中且 x 在 B 中$\}$

(3) A 和 B 的差　$A - B = \{x \mid x$ 在 A 中但 x 不在 B 中$\}$

(4) A 和 B 的笛卡尔积　$A \times B = \{(a,b) \mid a$ 在 A 中，b 在 B 中$\}$，$A^n = \underbrace{A \times A \times \cdots \times A}_{n \uparrow A}$

(5) A 的幂集　$2^A = \{B \mid B \subseteq A\}$

例如，设 $A = \{1,2\}, B = \{2,3\}$。则有 $A \cup B = \{1,2,3\}, A \cap B = \{2\}, A - B = \{1\}, A \times B = \{(1,2),(1,3),(2,2),(2,3)\}, 2^A = \{\phi, \{1\}, \{2\}, \{1,2\}\}$。

如果 A 和 B 分别有 n 个和 m 个元素，则 $A \times B$ 有 $n \times m$ 个元素，2^A 有 2^n 个元素。

用 A^B 表示集合 B 到 A 的全体映射（或称函数）的集合。

若 $\sigma \in A^A$，满足 $\sigma(a) = a (\forall\, a \in A)$，则称 σ 为 A 上的恒等映射。

集合 A 上的关系 R 是 $A \times A$ 的子集合。换句话说，关系是一个序偶集合，每个序偶的第 1 个分量选自定义域，而第 2 个分量选自值域（定义域集合和值域集合可能不同，也可能相同）。这里主要使用定义域和值域是同一集合 A 的关系，在此情况下，称关系 R 是建立在 A 上的关系。若 R 是一个关系，(a,b) 是 R 中的序偶，则记为 aRb。

关系的性质　称集合 A 上的关系 R 是：

(1) 自反的　如果对于 A 中一切 a，有 aRa

（2）非自反的　如果对于 A 中的一切 a，$a\overline{\mathrm{R}}a$，即$(a,a)\notin \mathrm{R}$

（3）传递的　如果 $a\mathrm{R}b$ 和 $b\mathrm{R}c$，则 $a\mathrm{R}c$

（4）对称的　如果 $a\mathrm{R}b$，则 $b\mathrm{R}a$

（5）反对称的　若 $a\mathrm{R}b$ 和 $b\mathrm{R}a$，必有 $a=b$

例如，整数集合上的关系＜是传递的，因为由 $a<b$ 和 $b<c$ 可以推出 $a<c$。它还是反对称的，因为由 $a<b$ 和 $b<a$ 可推出 $a=b$。

一个自反的、对称的和传递的关系 R 称为等价关系。

集合 A 上的关系 R 的正闭包 R^+ 是一个关系。$a\mathrm{R}^+b$ 的充分必要条件为存在 $a=a_1,a_2,\cdots,a_{n-1},a_n=b(n\geqslant 2)$，使得 $a_i\mathrm{R}a_{i+1}(i=1,2,\cdots,n-1)$。关系 R 的闭包定义为：$a\mathrm{R}^*b$ 的充分必要条件为 $a=b$ 或 $a\mathrm{R}^+b$。

§1－4　经典逻辑

利用符号及逻辑运算符∧、→等可以把某些句子表示为逻辑公式，例如：

F_1　$P\wedge Q\rightarrow R$ 表示句子“如果张三身高 1.8m 并且体重 80kg，则张三是大个子。”

F_2　$Q\rightarrow P$ 表示“如果张三体重 80kg，则张三身高 1.8m。”

F_3　Q 表示“张三体重 80kg。”

F_4　R 表示“张三是大个子。”

当逻辑公式 F_1、F_2 和 F_3 为真时，公式 F_4 为真，称 F_4 是逻辑上由 F_1、F_2 和 F_3 推得。

在上面的例子中，必须证明一个公式逻辑上由另一个公式推得，“一个公式逻辑上由另一个公式推得”的陈述称为定理，定理为真的证明就是指一个公式逻辑上由另一个公式推出。数学定理证明的问题是考虑寻找定理证明的数学方法。有许多问题可以转变成定理证明的问题，比如，在程序正确性证明中，用公式 A 描述一个程序的执行，公式 B 表示程序终止的条件，那么证明程序终止等价于证明公式 B 由公式 A 推出。

本节简要介绍命题逻辑与一阶逻辑。

一、命题逻辑

命题逻辑研究的对象是命题。命题是一个陈述语句，它可以为真或假但不能两者都成立，比如，“长江是世界上第三大河流”是命题，而且它是真的；“长江是世界上第一大河流”是命题，但它是假的。

一般用大写英文字母或大写字符串表示命题，T 与 F 分别表示命题是真与假，T 和 F 称为命题的真值。用于表示命题的符号(如上面的 P、Q 与 R)称为原子，原子通过逻辑运算～、∨、∧、→与↔可以构成复合命题，利用复合命题可以表示一个更复杂的概念。

设 P、Q 是两个命题，P 与 Q 经逻辑运算后，其真值列如表 1－1。

命题逻辑中，表示命题的表达式(比如 P、$P\wedge Q\rightarrow R$)称为合式公式，简称公式。公式可递归地定义如下：

1. 原子是公式；
2. 如果 A、B 是公式，则～A、$A\vee B$、$A\wedge B$、$A\rightarrow B$、$A\leftrightarrow B$ 是公式；
3. 重复使用 1 与 2 可产生所有公式。

表 1-1 P 与 Q 经逻辑运算后的真值

P	Q	$\sim P$	$P \vee Q$	$P \wedge Q$	$P \rightarrow Q$	$P \leftrightarrow Q$
T	T	F	T	T	T	T
T	F	F	T	F	F	F
F	T	T	T	F	T	F
F	F	T	F	F	T	T

公式是由原子、逻辑联结词、括号组成的字符串，公式的真值是由对原子的真值计算得到的。

设 G 是公式，A_1，…，A_n 是 G 中的全部原子（或称命题变元），规定 A_1，…，A_n 的一组真值，则这组真值称为 G 的一个赋值（或解释），用 v 表示 G 的赋值，显然 G 有 2^n 个赋值。G 在其所有可能赋值下所取真值的表，称为 G 的真值表。通常用集合 $\{m_1, \cdots, m_n\}$ 表示 G 的一个赋值，其中

$$m_i = \begin{cases} A_i & \text{当 } A_i \text{ 在 } v \text{ 下为 } T \text{ 时} \\ \sim A_i & \text{当 } A_i \text{ 在 } v \text{ 下为 } F \text{ 时} \end{cases} \quad i = 1,2,\cdots,n$$

例如集合$\{P,\sim Q,\sim R,S\}$表示 G 的一个赋值，在此赋值下 P、Q、R 和 S 分别赋给 T、F、F 和 T。

如果 G 在它的所有赋值下都是真的，则称公式 G 为永真的（重言式）；如果 G 在其所有赋值下都是假的，则称 G 为永假的（不可满足的）；如果 G 不是不可满足的，则 G 称为可满足的。由此显然有：

G 永真当且仅当$\sim G$ 是永假的（不可满足的）；

G 是可满足的当且仅当至少有一个赋值 v，使 G 在 v 下为真；

G 是非永真的当且仅当至少有一个赋值 v，使 G 在 v 下为假；

如果 G 是永真的，则 G 是可满足的，但逆不成立；

如果 G 是永假的，则它是非永真的，但逆不成立。

如果 G 在 v 下是真的，则称 v 满足 G；如果 G 在 v 下是假的，则称 v 弄假 G。例如，公式 $P \wedge \sim Q$ 在赋值$\{P,\sim Q\}$下是真的，所以$\{P,\sim Q\}$满足 $P \wedge \sim Q$，但此公式被赋值$\{P,Q\}$弄假。

命题逻辑中，由于一个公式的赋值数目是有限的，所以至少可以用穷举法决定命题逻辑中公式的永真或永假性，即命题逻辑的判定问题是可解的。

常常需要把一个公式由一种形式变换到另一种形式（特别是范式），这是在给定公式中用与它等价的一个公式来代替而得到的，这个代替过程直到所要求的形式被得到为止。

公式 F、G 称是逻辑等价的（表示为 $F=G$），当且仅当 F 和 G 在其每个赋值下有相同的真值。

例如，$P \rightarrow Q$ 等价于$(\sim P \vee Q)$，用列真值表的方法可以验证。

表 1-2 中给出了常用的等价公式（又称“规则”）。

表 1-2　常用的等价公式

(1-1) $F\leftrightarrow G=(F\rightarrow G)\wedge(G\rightarrow F)$	
(1-2) $F\rightarrow G=\sim F\vee G$	
(1-3) (a)$F\vee G=G\vee F$　(b)$F\wedge G=G\wedge F$	交换律
(1-4) (a)$(F\vee G)\vee H=F\vee(G\vee H)$　(b)$(F\wedge G)\wedge H=F\wedge(G\wedge H)$	结合律
(1-5) (a)$F\vee(G\wedge H)=(F\vee G)\wedge(F\vee H)$　(b)$F\wedge(G\vee H)=(F\wedge G)\vee(F\wedge H)$	分配律
(1-6) (a)$F\vee\square=F$　(b)$F\wedge\blacksquare=F$ (1-7) (a)$F\vee\blacksquare=\blacksquare$　(b)$F\wedge\square=\square$	泛界律
(1-8) (a)$F\vee\sim F=\blacksquare$　(b)$F\wedge\sim F=\square$	互余律
(1-9) $\sim(\sim F)=F$	
(1-10) (a)$\sim(F\vee G)=\sim F\wedge\sim G$　(b)$\sim(F\wedge G)=\sim F\vee\sim G$	De Morgan 律

其中□表示真值 F，■表示真值 T。

若 F_1，F_2，…，F_n 是公式，则 $F_1\vee F_2\vee\cdots\vee F_n$ 称为 F_1，F_2，…，F_n 的析取；$F_1\wedge F_2\wedge\cdots\wedge F_n$ 称为 F_1，…，F_n 的合取。

原子或原子的非称为文字。

公式 G 称为合取范式，当且仅当 G 有下列形式

$$G\triangleq G_1\wedge\cdots\wedge G_n\qquad n\geqslant 1$$

其中 G_i 是一些文字的析取，$i=1$，…，n

公式 G 称为析取范式，当且仅当 G 有下列形式

$$G\triangleq G_1\vee\cdots\vee G_n\qquad n\geqslant 1$$

其中 G_i 是一些文字的合取，$i=1$，…，n

合取范式（析取范式）并不唯一。利用表 1-2 中给出的规则，可以把任意公式 G 变换成与 G 等价的合取或析取范式，步骤如下：

步 1　使用规则(1-1)和(1-2)消去逻辑连接词↔和→。

步 2　重复使用规则(1-9)和(1-10)，将 G 中所有的～都放在原子之前。

步 3　反复使用规则(1-5)，即可得到范式。

例 1-2　化 $(P\vee\sim Q)\rightarrow R$ 为析取范式。

$$\begin{aligned}(P\vee\sim Q)\rightarrow R&=\sim(P\vee\sim Q)\vee R && \text{用(1-2)}\\&=(\sim P\wedge\sim(\sim Q))\vee R && \text{用(1-10)}\\&=(\sim P\wedge Q)\vee R && \text{用(1-9)}\end{aligned}$$

这表明只需使用三个联结词{～、∨、∧}（或两个联结词{～、→}）就可以表示命题逻辑公式。如果一切使命题公式集 Γ 中每一公式均取真值 T 的赋值 v，亦使公式 A 取真值 T，则称 A 是 Γ 的逻辑结果（或 Γ 逻辑蕴涵 A），记为 $\Gamma\models A$，例如，$\{A,A\rightarrow B\}\models B$，$\{\sim A,A\vee B\}\models B$。

给定公式集 Γ 和公式 G，称 G 为 Γ 的演绎结果（或 G 逻辑上由 Γ 推出，记为 $\Gamma\vdash G$）当且仅当存在有穷公式序列 G_1，G_2，…，$G_k(=G)$，使得 $G_i(i=1,2,\cdots,k)$ 或是公理，或是 Γ 中公式，或是由已证公式使用推理规则得出的。G_i 称为 G 的由 Γ 出发的演绎，Γ 称为演绎的前提。

下述三个公式可以作为命题演算形式系统 FSPC 的公理：

A_1. $(A\to(B\to A))$

A_2. $((A(B\to C))\to((A\to B)\to(A\to C)))$

A_3. $(((\sim A)\to(\sim B))\to(B\to A))$

$\dfrac{A,\ A\to B}{B}$是 FSPC 中唯一的推理规则，称为分离规则，或称三段论。

可以证明，$\Gamma\vdash A$(A 可由 Γ 推出)当且仅当 $F\models A$(A 是 Γ 的逻辑结果)。

证明定理或推演某个演绎结果，是从公理或前提出发逐步构造证明序列和演绎序列，这种证明方式称为公理化推演。

例 1-3 证明$\sim B\to(B\to A)$是 FSPC 的定理，即 $\vdash_{FSPC}\sim B\to(B\to A)$。

从$\sim B$ 出发，利用 A_1 得 $\sim B\to(\sim A\to\sim B)$ (1)

从 (1) 的后件出发，利用 A_3 得 $\underbrace{(\sim A\to\sim B)}_{C_1}\to\underbrace{(B\to A)}_{C_2}$，记为 C (2)

从 (2) 出发，利用 A_1 得 $C\to(\sim B\to C)$ (3)

对 (2)、(3) 使用分离规则 $\sim B\to C$，即$\sim B\to(C_1\to C_2)$ (4)

对 (4) 使用 A_2 $(\sim B\to(C_1\to C_2))\to((\sim B\to C_1)\to(\sim B\to C_2))$ (5)

对 (4)、(5) 使用分离规则 $(\sim B\to C_1)\to(\sim B\to C_2)$ (6)

对 (1)、(6) 使用分离规则 $\sim B\to C_2=\sim B\to(B\to A)$ (7)

FSPC 是一致的 (或相容的)，如果不存在 FSPC 的公式 A，使得 $\vdash_{FSPC}A$ 与 $\vdash_{FSPC}\sim A$ 同时成立。

FSPC 不是完全的，即存在 FSPC 的公式 A，使得 $\vdash_{FSPC}A$ 与 $\vdash_{FSPC}\sim A$ 都不成立。

FSPC 是可判定的，如果存在一个算法对 FSPC 中任一公式 A，可确定 $\vdash_{FSPC}A$ 是否成立；否则称为不可判定的。如果存在一过程，对系统中的定理可作出肯定的判断，但对非定理的公式不能作出判断，则称该系统为半可判定的。

形式系统 FS 是一个五元组(Σ,T,F,A,R)，其中 Σ 是 FS 的符号表；T 是 Σ^* 的一个子集，其元素称为 FS 的项，T 有子集 VA，其元素称为变元；F 也是 Σ^* 的一个子集，它的元素是 FS 的公式。$T\cap F=\phi$。A 称为 FS 的公理集，它也是 F 的一个子集。最后，R 是 FS 的推理规则集。

形式系统 FS 中全体定理的集合称为 FS 的理论，记为 Th(FS)。FS 是可判定的，当且仅当 Th(FS)为递归集；FS 是半可判定的，当且仅当 Th(FS)为递归可枚举集。

可以证明，FSPC 是一致的，而且还是可判定的，但不是完全的(即存在公式 A，使 $\vdash A$ 与 $\vdash\sim A$ 都不成立)。赋给 FS 的论域及解释称为 FS 的语义结构，语义结构及该结构下公式真值的规定，称为 FS 的语义。有所谓指称语义 (上述定义的语义)、操作语义、公理化语义等。

形式系统 FS 的结构 $\mathscr{U}=\langle U,I\rangle$，其中 U 为非空集合，称为论域或个体域；I 是一组规则，称为解释，它规定项如何指称 U 中个体，原子公式如何指称 U 中个体的性质(或 U 的子集)、关系(或 U^n 的子集)。

给定形式系统 FS，映射 s：变元$\to U$ 称为指派。指派 s 的扩展 $\bar{s}$：项$\to U$，使得

$$\bar{s}(t)=\begin{cases}s(t) & t\text{ 为变元}\\ s\text{ 指派 }t\text{ 中变元后由解释 }I\text{ 确定} & t\text{ 为非变元}\end{cases}$$

设 FS 的结构为 $\mathscr{U}$，则 FS 的每一个公式 A 对应于 U 的一个命题，记为 $A_{\mathscr{U}}[s]$。$v(A_{\mathscr{U}}[s])$ 表示原子公式 A 在结构 $\mathscr{U}$ 中指派 s 下的值，$\bar{v}(A_{\mathscr{U}}[s])$表示公式 A 在结构 $\mathscr{U}$ 中指派 s 下的值。$\models_{\mathscr{U}}A[s]$表示 $\bar{v}(A_{\mathscr{U}}[s])=1$，即公式 A 在结构 $\mathscr{U}$ 中指派 s 下真。如果对一切 s 有$\models_{\mathscr{U}}A[s]$，则称 A 在 $\mathscr{U}$ 中真(或 $\mathscr{U}$ 为 A 的模型)，记为$\models_{\mathscr{U}}A$。如果对一类结构 M 中的每一个结构 $\mathscr{U}$，均有$\models_{\mathscr{U}}A$，则称 A 永真，记为$\models_{M}A$。

A 为 Γ（公式集）的逻辑结果，可表示成 $\Gamma\models_{M}A$，即弄真 Γ 中每一公式的结构 $\mathscr{U}$ 指派 s 亦必弄真 A。

如果 A 是 FS 中任一公式，$\vdash_{FS}A$，那么$\models_{M}A$，则称 FS 是合理的。如果$\models_{M}A$，那么 $\vdash_{FS}A$，则称 FS 是完备的。可以证明，FSPC 既是合理的，又是完备的。

如果包含初等数论的形式系统 FS 是一致的，那么 FS 是不完备的，即存在表示初等数论真命题的公式 A，使得 A 与$\sim A$ 都不是 FS 的定理。

二、一阶逻辑

命题逻辑的研究对象是命题，命题是有真假意义的陈述语句。命题逻辑中不考虑陈述语句的结构和成分，因此许多思维过程不能用命题逻辑中的命题来表达，为此需要引入谓词、函词、量词、变元和常元等概念，这些是一阶逻辑的逻辑成分。

谓词是表示客体性质和关系的语言成分，它带有若干个空位，当空位填入对象后谓词才表示关于所填对象的语句，空位的数目称为谓词的元数。常用大写字母或字符串表示谓词，e_i 表示空位，例如，一元谓词 $H(x)$表示“x 是人”；二元谓词 Le_1e_2 表示“e_1 小于 e_2”。

函词是表示某种操作的语言成分，它也带有若干个空位，当空位填入对象后函词才产生出一个对象。空位数目称为函词的元数。函词用小写拉丁字母或字符串表示，e_i 表示空位，例如，一元函词 fe_1 表示“e_1 的小数部分”；二元函词 $adde_1e_2$ 表示“e_1 与 e_2 的和”。

变元表示论域 U 上的任一对象，变元分为自由变元和约束变元。对自由变元可作代入，比如 $x+y=0$ 是二元谓词填式，其中 x，y 是自由变元，对 x、y 可代入实数。自由变元不能改名，例如 x 和 y 代表不同的意义，故不能把 y 改为 x。与此相反，对约束变元不能作代入，但可以改名，例如 $\sum_{i=1}^{n}f(i)$，是用变元 i（约束变元）简化和式的一种记号，对 i 不能作代入，因为 $f(8)$不再是和式，但 i 可以改名为 j，$\sum_{j=1}^{n}f(j)$ 与 $\sum_{i=1}^{n}f(i)$ 的意义相同。

常元是表示确定对象的符号。

量词分全称量词(“对任意 x”，即$\forall x$，)和存在量词(“存在 x”，即$\exists x$)两种。$\forall xPx$ 表示“所有 x 均满足 $P^{(1)}(Px$ 真)”，$\exists xPx$ 表示“存在 x 满足 $P^{(1)}(Px$ 真)”；$\sim\forall xPx$ 表示“并非所有 x 满足 $P^{(1)}$”，$\sim\exists xPx$ 表示“不存在 x 满足 $P^{(1)}$”。

$\forall xPx$ 中的 Px 称为量词$\forall$ 的辖域，x 为约束变元。$\exists xPx$ 同样。

变元和常元是项；如果 $f^{(n)}$是 n 元函词符号，t_1，…，t_n 是项，则 $f^{(n)}t_1\cdots t_n$ 是项。

谓词填式是公式(即原子公式 $P^{(n)}t_1\cdots t_n$)；如果 x 是变元，A、B 是公式，则$\sim A$，$A\vee B$，$A\wedge B$，$A\rightarrow B$，$A\leftrightarrow B$，$\forall xA$，$\exists xA$ 是公式。

一阶谓词演算形式系统 FSFC，只使用逻辑联结词$\sim$、$\rightarrow$和量词$\forall$。FSFC 的语言（即一阶语言）由个体变元 v_i、个体常元 a_i、函词 $f^{(n)}$、谓词 $P^{(n)}$、$\sim$、$\rightarrow$、$\forall$ 及括号组成，项和公

式的定义与上相同。

如果公式 A 中自由变元 v 的任何自由出现都不在$\forall u(\exists u)$的辖域内，则称项 t(u 是 t 中任一自由变元)对 A 中 v 可代入。

对公式 A 中变元 v 的所有自由出现都代换为项 t 的过程称为代入，代换后所得公式称为 A 的代入实例，记为 A_t^v。

命题逻辑中的 A_1、A_2 和 A_3 再加上下述三个公式可构成一阶逻辑的公理组：

A_4. $\forall vA\to A_t^v$(t 对 A 中变元 v 可代入)

A_5. $\forall v(A\to B)\to(\forall vA\to\forall vB)$

A_6. $A\to\forall vA$(v 在 A 中无自由出现)

一阶逻辑中的推理规则仍是分离规则。

显然，FSPC 中的定理均可看作 FSFC 中的定理，只是语法变元（即系统中的公式）A、B 等表示 FSFC 的公式。

设 Γ 是 FSFC 中公式集，A、B 是 FSFC 的任意公式，则 $\Gamma;A\vdash B$ 当且仅当 $\Gamma\vdash A\to B$。

设 Γ 是 ΓSFC 的公式集，A 为公式，变元 v 不在 Γ 的任一公式里自由出现，如果 $\Gamma\vdash A$，则有 $\Gamma\vdash\forall vA$。

例 1-4 设 Γ 是 FSFC 的公式集，A、B 为任意公式，变元 v 在 Γ 的任一公式及 B 中无自由出现，则由 $\Gamma\vdash\exists vA$ 及 $\Gamma;A\vdash B$ 可推出 $\Gamma\vdash B$。($\exists$ 消除规则)

证明　由 $\Gamma;A\vdash B$ 可得 $\Gamma\vdash A\to B$。又由重言式$(A\to B)\to(\sim B\to\sim A)$，可推得 $\Gamma\vdash\sim B\to\sim A$。根据题给条件，$\Gamma\vdash\forall v(\sim B\to\sim A)$，进而有 $\Gamma\vdash\forall v\sim B\to\forall v\sim A$ 及 $\Gamma\vdash\sim\forall v\sim A\to\sim\forall v\sim B$，即 $\Gamma\vdash\exists vA\to\exists vB$。题已给 $\Gamma\vdash\exists vA$，因此可得 $\Gamma\vdash\exists vB$。因为 v 在 B 中无自由出现，$\sim B\to\forall v\sim B$ 即公理 A_6，所以有 $\Gamma\vdash\sim\forall v\sim B\to B$，即 $\Gamma\vdash\exists vB\to B$。上面已得到 $\Gamma\vdash\exists vB$，因此有 $\Gamma\vdash B$。

设 x，y 是变元，A 是公式，则有关系式：

$$\forall xA\text{ 真当且仅当 }\sim(\exists x\sim A)\text{ 真}$$
$$\exists xA\text{ 真当且仅当 }\sim(\forall x(\sim A))\text{ 真}$$
$$\sim(\forall xA)\text{ 真当且仅当 }\exists x(\sim A)\text{ 真}$$
$$\sim(\exists xA)\text{ 真当且仅当 }\forall x(\sim A)\text{ 真}$$
$$\forall x\forall yA\text{ 真当且仅当 }\forall y\forall xA\text{ 真}$$
$$\exists x\exists yA\text{ 真当且仅当 }\exists y\exists xA\text{ 真}$$

一阶谓词演算形式系统 FSFC 的塔斯基语义结构类 T，T 中结构以$\{0,1\}$为其真值集。结构 $\mathscr{U}\in T$，由非空集合 U 和解释 I 组成，规定 I 如下：

设 a 为常元，则 $\bar{a}=I(a)\in U$。

设 $f^{(n)}$ 是 n 元函词，则 $\bar{f}^{(n)}=I(f^{(n)})$，$U^n\to U$ 是 U 上的 n 元函数。

设 $P^{(n)}$ 是 n 元谓词，则 $\bar{P}^{(n)}=I(P^{(n)})\subseteq U^n$ 是 U 上的 n 元关系。

指派 s 是变元集合$\{v_1,v_2,\cdots\}$到 U 上的映射。s 可扩展为映射 $\bar{s}$：项 $t\to U$：

$$\bar{s}(t')=\begin{cases}\bar{a} & t'\text{ 为常元}\\ s(t') & t'\text{ 为变元}\\ \bar{f}^n\bar{s}(t_1)\cdots\bar{s}(t_n) & t'\text{ 为 }f^{(n)}t_1\cdots t_n;f^{(n)}\text{ 是 }n\text{ 元函词},t_i\text{ 是项}\end{cases}$$

s 与 $\mathscr{U}$ 无关，而 $\bar{s}$ 与 $\mathscr{U}$ 有关，$\mathscr{U}$ 与指派共同决定了 FSFC 中项和原子公式的指称。

在结构 $\mathscr{U}$、指派 s 下规定赋值映射 v：

（1）定义 v　原子公式$_{\mathscr{U},s}\rightarrow\{0,1\}$ 为：对 n 元谓词 $P^{(n)}$ 及项 $t_1,\cdots,t_n$

$$v(\bar{P}^{(n)}\bar{s}(t_1)\cdots\bar{s}(t_n))=1 \text{ 当且仅当} \langle\bar{s}(t_1),\cdots,\bar{s}(t_n)\rangle\in\bar{P}^{(n)}$$

（2）v 扩展为赋值函数 $\bar{v}$　公式$_{\mathscr{U},s}\rightarrow\{0,1\}$

A 是原子公式，$\bar{v}(A_{\mathscr{U}}[s])=v(A_{\mathscr{U}}[s])$

对于 $\sim B$，$\bar{v}(\sim B)_{\mathscr{U}}[s])=1$ 当且仅当 $\bar{v}(B_{\mathscr{U}}[s])=0$

对于 $B\rightarrow C$，$\bar{v}((B\rightarrow C)_{\mathscr{U}}[s])=1$，当且仅当 $\bar{v}(B_{\mathscr{U}}[s])=0$ 或者 $\bar{v}(C_{\mathscr{U}}[s])=1$

对于 $\forall vA$，$\bar{v}((\forall vA)_{\mathscr{U}}[s])=1$ 当且仅当对每个 $d\in U$，$\bar{v}(A_{\mathscr{U}}[s(v|d)])=1$

其中 $s(v|d)$ 表示限定 s 对 v 指派 d，即对任一变元 u，

$$s(v|d)(u)=\begin{cases}s(u) & \text{当 } u\neq v\\ d & \text{当 } u=v\end{cases}$$

公式 $\forall vA$ 在结构 $\mathscr{U}$ 中指派 s 下真表示为 $\models_{\mathscr{U}}\forall vA[s]$，当且仅当 $\bar{v}((\forall vA)_{\mathscr{U}}[s])=1$，因此 $\models_{\mathscr{U}}\forall vA[s]$ 当且仅当对任一 $d\in U$ 均有 $\models_{\mathscr{U}}A[s(v|d)]$。

Γ 逻辑蕴涵 A（记为 $\Gamma\models_T A$）定义为：设 B 是 Γ 中任一公式，使 $\models_{\mathscr{U}}B[s]$ 成立的 T 中结构 $\mathscr{U}$ 和指派 s，恒使 $\models_{\mathscr{U}}A[s]$ 成立。

可以证明 FSFC 中的每个公理 A_i 在 T 中所有语义结构 $\mathscr{U}$ 里都为真，即 A_i 永真或 $\models_T A_i$。

设 A 是 FSFC 中任一公式，如果 $\vdash_{\text{FSFC}}A$，那么 $\models_T A$。因此，FSFC 是合理的。

FSFC 是一致的，即不存在公式 A（FSFC 中），使 $\vdash A$ 及 $\vdash\sim A$ 同时成立。

FSFC 是不完全的，即存在 FSFC 的公式 A，使 $\vdash A$ 与 $\vdash\sim A$ 都不成立。

FSFC 是完备的，即所有永真公式均为 FSFC 的定理。

FSFC 是半可判定的，即存在一过程，可对 FSFC 的定理作出肯定判断，但对非定理的 FSFC 的公式却未必能作出否定判定。