中国计算机学会
学术著作丛书

逻辑程序设计语言及其实现技术

王鼎兴 温冬婵 高耀清 黄志毅

清华大学出版社
广西科学技术出版社

中国计算机学会学术著作丛书

逻辑程序语言及其实现技术

王鼎兴　温冬婵
高耀清　黄志毅　编著

清华大学出版社
广西科学技术出版社

(京)新登字 158 号

内 容 提 要

本书详细论述了逻辑程序设计语言的实现技术和面向逻辑程序设计语言的体系结构。其主要内容包括，逻辑程序理论基础，Prolog 语言简介，WAM 编译技术，WAM 的扩充和优化，顺序推理机体系结构，逻辑程序并行处理方法和模型，抽象解释技术，并行逻辑程序设计语言及其实现技术、并行推理机体系结构等。

本书适合从事计算机体系结构、信息处理、并行处理技术研究的科技人员阅读，也可作为计算机自动化信息处理等专业研究生和高年级本科生的教材和教学参考书。

图书在版编目(CIP)数据

逻辑程序语言及其实现技术/王鼎兴等编著．—北京：清华大学出版社，1995
(中国计算机学会学术著作丛书)
ISBN 7-302-01841-3

Ⅰ.逻… Ⅱ.王… Ⅲ.LOGIC 语言-程序语言-程序设计 Ⅳ.TP312L0

中国版本图书馆 CIP 数据核字 (95) 第 05675 号

出 版 者：清华大学出版社(北京清华大学校内，邮编：100084)
广西科学技术出版社(南宁市河堤路 14 号，邮编：530021)
印 刷 者：人民文学印刷厂
发 行 者：新华书店总店北京科技发行所
开 本：787×1092 1/16 印张：23.5 字数：555 千字
版 次：1996 年 1 月第 1 版 1996 年 1 月第 1 次印刷
书 号：ISBN 7-302-01841-3/TP・827
印 数：0001—3000
定 价：29.00 元

出 版 说 明

近年来，随着微电子和计算机技术渗透到各个技术领域，人类正在步入一个技术迅猛发展的新时期。这个新时期的主要标志是计算机和信息处理的广泛应用。计算机在改造传统产业，实现管理自动化，促进新兴产业的发展等方面都起着重要作用，它在现代化建设中的战略地位愈来愈明显。计算机科学与其它学科的交叉又产生了许多新学科，推动着科学技术向更广阔的领域发展，正在对人类社会产生深远的影响。

科学技术是第一生产力。计算机科学技术是我国高科技领域的一个重要方面。为了推动我国计算机科学及产业的发展，促进学术交流，使科研成果尽快转化为生产力，清华大学出版社与广西科学技术出版社联合设立了“计算机学术著作基金”，旨在支持和鼓励科技人员，撰写高水平的学术著作，以反映和推广我国在这一领域的最新成果。

计算机学术著作出版基金资助出版的著作范围包括：有重要理论价值或重要应用价值的学术专著；计算机学科前沿探索的论著；推动计算机技术及产业发展的专著；与计算机有关的交叉学科的论著；有较大应用价值的工具书；世界名著的优秀翻译作品。凡经作者本人申请，计算机学术著作出版基金评审委员会评审通过的著作，将由该基金资助出版，出版社将努力做好出版工作。

基金还支持两社列选的国家高科技重点图书和国家教委重点图书规划中计算机学科领域的学术著作的出版。为了做好选题工作，出版社特邀请“中国计算机学会”、“中国中文信息学会”帮助做好组织有关学术著作丛书的列选工作。

清华大学出版社
广西科学技术出版社　　计算机学术著作出版基金办公室

1992 年 4 月

序　言

计算机是当代发展最为迅猛的科学技术，其应用几乎已深入到人类社会活动和生活的一切领域，大大提高了社会生产力，引起了经济结构、社会结构和生活方式的深刻变化和变革，是最为活跃的生产力之一。计算机本身在国际范围内已成为年产值达2500亿美元的巨大产业，国际竞争异常剧烈，预计到本世纪末将发展为世界第一大产业。计算机科技具有极大的综合性质，与众多科学技术相交叉而反过来又渗入更多的科学技术，促进它们的发展。计算机科技内容十分丰富，学科分支生长尤为迅速，日新月异，层出不穷。因此在我国计算机科技尚比较落后的情况下，加强计算机科技的传播实为当务之急。

中国计算机学会一直把出版图书刊物作为学术活动的重要内容之一。我国计算机专家学者通过科学实践，做出了大量成果，积累了丰富经验与学识。他们有撰写著作的很大积极性，但相当时期以来计算机学术著作由于印数不多，出版往往遇到不少困难，专业性越强越有深度的著作，出版难度越大。最近清华大学出版社与广西科学技术出版社为促进我国计算机科学技术及产业的发展，推动计算机科技著作的出版工作，特设立“计算机学术著作出版基金”，以支持我国计算机科技工作者撰写高水平的学术著作，并将资助出版的著作列为中国计算机学会的学术著作丛书。我们十分重视这件事，并已把它列为学会本届理事会的工作要点之一。我们希望这一系列丛书能对传播学术成果、交流学术思想、促进科技转化为生产力起到良好作用，能对我国计算机科技发展具有有益的导向意义，也希望我国广大学会会员和计算机科技工作者，包括海外工作和学习的神州学人们能积极投稿，出好这一系列丛书。

中国计算机学会

1992年4月20日

清华大学出版社　广西科学技术出版社

计算机学术著作出版基金

评 审 委 员 会

主 任 委 员　张效祥

副主任委员　周远清　汪成为

委　　员　王鼎兴　杨芙清　李三立　施伯乐　徐家福

夏培肃　董韫美　张兴强　徐培忠

About the Book

In this book, the authors discuss in detail the implementation techniques for Logic Programming Languages (LPL) and the LPL-oriented architectures. They cover the fundamental of logic programming, an introduction to Prolog, WAM compiling techniques, extention and optimization techniques for WAM, sequential inference machine architectures, parallel processing approaches and models for logic programming, abstract interpretation techniques, parallel logic programming languages and their implementation techniques, parallel inference machine architectures, etc.

This book is helpful for these people who engage in research and development of computer architecture, information processing, parallel computing, etc. It is also a text book or reference book for the undergraduate and graduate students whose major is computer science, automation, or information processing, etc.

序

将逻辑与程序设计结合在一起是计算机科学领域中的一个创举。它向世人宣告，逻辑不仅仅是计算机科学的理论指导，它已从高高的理论殿堂走向计算机日常的工程实践。

在美国常被人们认为是“人工智能之父”的 John McCarthy 曾就逻辑在计算机科学和人工智能领域中所起的作用问题有如下论述：

“我们有理由期待着下个世纪计算方法与数理逻辑的结合将会结出丰硕的成果，就像上个世纪分析方法和物理学的结合产生出丰硕的成果一样”(1967)。

在今天看来，这一预言已经或正在变成现实。在逻辑程序设计领域中，从理论到实现，都已产生许多令人瞩目的成就。而且，逻辑与计算机应用的结合还产生了许多新领域，如定理机器证明、抽象数据类型说明、程序验证、程序综合、非单调推理等等。这些都充分展示了逻辑对计算机科学所具有的独特的魅力。

自 Colmerauer 和 Kowalski 提出使用谓词逻辑进行程序设计的思想以来，逻辑程序设计已有了飞速的发展。在语言方面，从最早的 Prolog 语言，发展到 PARLOG，Concurrent Prolog，GHC 等并发语言，以及后来提出的各种为开发并行性而设计的并行语言。在执行方式上，从早期的解释执行，发展到编译执行。在执行模型和抽象机方面，从最先的解释模型发展到 WAM 模型，直至现在各种开发 AND、OR 等并行性的并行执行模型。在体系结构的实现上，有顺序处理机，并行处理机，有通用机，也有专用机。在这短短 20 年里，逻辑程序设计能有如此迅速的发展可说是出乎 John McCarthy 意料的。

逻辑程序之所以能受到各方面人士的关注，是因其独有的特点。Kowalski 用一个等式很好地描述了逻辑程序的基本特点：算法＝逻辑＋控制。其思想是，程序员仅描述他希望程序干什么，而将具体如何执行留给机器去做。除此之外，逻辑程序还具有良好的理论基础。它以一阶逻辑的 Horn 子集为基础，其正确性和完备性都得到过严格的证明。逻辑程序的多种非确定性便于程序并行性的开发，其丰富的表达能力对于知识处理、人工智能等应用领域的开发具有很大的促进作用。

逻辑程序实现技术的研究是逻辑程序设计能否具有生命力的关键。虽然逻辑程序已从低效的解释执行，发展到了高效的编译和并行执行，但与传统语言相比，其实现效率仍待进一步提高，其编程环境也需要再完善与改进。80 年代以来，人们对智能技术与系统以及它们的应用前景抱有很高的期望，各国都投入相当的人力、物力开展这一领域的研究开发工作。十多年来实际进展的结果并非如人们预期之高。仔细想来这也很自然，因为技术发展需要一定的过程，而且需求也尚未迫切到非使用它而无法前进的程度。尽管如此，人工智能、知识工程、函数和逻辑程序设计等的研究前景依然是光明的，因而有必要对前一时期有关的工作进行整理、分析与总结。这就是促使作者写作这本书的主要动机。我们希望书中讨论的这些实现方法和技术能为从事这一领域工作的同行们提供参考和借鉴。

全书共分十章。第一章简述逻辑程序理论基础及一些背景知识。第二章着重介绍

Prolog 语言及其编程风格，通过一些实例说明 Prolog 语言的特点及其解释执行机制。第三章介绍编译执行 Prolog 语言的抽象机 WAM。第四章给出了对 WAM 抽象机的各种优化实现技术以及 WAM 的扩充。第五章讨论各种顺序推理机的体系结构和实现技术。第六章介绍逻辑程序并行处理中的一些基本原理和概念。第七章分析全解逻辑程序设计语言的并行处理方法和模型，尤其是通过抽象解释技术开发并行性和提取程序优化的信息。第八章介绍几种流行的并行逻辑程序设计语言并对其进行分析比较。第九章着重讨论 PARLOG 语言的各种实现技术。第十章总结了各种并行推理机的体系结构和实现技术。

理论指导实践，实践又反过来推动理论的发展。近年来实现技术的进步促使了不少新理论的产生，如约束逻辑程序设计(Constraint Logic Programming，简称 CLP)等。由于这些工作正在发展中，本书未予收录，感兴趣的读者可查阅有关文献进行分析研究。

作　者

1994 年 10 月

目　录

CONTENTS

第一章　逻辑程序的基础知识

1.1　逻辑程序的发展历史

逻辑程序设计(Logic Programming)始于70年代早期。它是早期自动定理证明和人工智能发展的自然结果。1930年,Herbrand为定理机器证明奠定了理论基础[1]。在60年代早期,自动定理证明的研究工作十分活跃,如Prawitz[2],Gilmore[3],Davis,Putnam[4]等的工作。1965年,J. A. Robinson提出了归结原理[5],它为计算机提供了机械证明的方法。

1972年Kowalski[6]和Colmerauer[7]提出了逻辑可以作为程序设计语言的基本思想,把逻辑和程序这两个截然不同的概念协调统一为一个概念——逻辑程序设计。这个思想产生了深远而持久的影响。1972年,Colmerauer和他的研究小组在马赛(Marseille)大学用ALGOL-W实现了第一个逻辑程序设计语言Prolog[8]。紧接着又用FORTRAN对原系统进行了改进[9],从而使逻辑程序迈向了可用阶段。早期的逻辑程序系统采用解释的方法,因而执行速度极慢,一度遭到了人们的非议。80年代初期,D. H. D Warren提出Warren抽象机模型WAM[10],采用编译的方法大大提高了逻辑程序的时空效率。与最初的解释方法相比,编译方法使Prolog的执行速度提高了一个数量级之多。从而逻辑程序迈向了实用阶段。

当今计算机技术迅猛发展。它已经历了数值计算,数据处理和信息处理阶段,而今正迈向知识信息处理阶段。未来信息处理发展的基本方向是知识信息处理[11]。基于一阶谓词逻辑的逻辑程序设计语言,将逻辑推理对应于计算,它的丰富表达能力、非确定性等特点,适合于表达人工智能和知识工程中的问题,从而使逻辑程序越来越得到广泛的应用。80年代逻辑程序设计受到了世界许多国家高技术计划的青睐。许多国家提出了研究新一代的计算机系统的计划,以提高推理速度。例如,1981年秋,日本提出了第五代计算机系统研究计划。逻辑程序不仅用作为应用编程语言,而且也用作为系统编程语言。1984年,日本生产出第一台个人顺序推理机,并首次用逻辑程序实现了个人顺序推理机的操作系统SIMPOS。

80年代初期起,逻辑程序并行处理技术的研究成为一个热点。在并行逻辑程序语言方面:Clark和Gregory提出了关系语言和Parlog[12][13]。Shapiro提出了Concurrent Prolog[14],Ueda提出了GHC[15],R. Yang和H. Aiso提出了P-Prolog[16][17],L. M. Pereira和R. Nasr提出了Delta-Prolog[18][19]。Haridi提出了Andorra Prolog[20],Wise提出了EPILOG[21][22]等等。在逻辑程序并行执行模型和体系结构方面,代表性的工作有:Pollard首创的并行执行模型[23]。Conery的AND/OR进程模型[24][25],DeGroot的RAP模型[26][27],Ciepielewski的TOKEN模型[28],Goto的G-R模型[29],孙成政的PSOF模型[30],Tung的并行模型[31],Hermenegildo的RAP-WAM抽象机模型[32][33][34],D. H. D Warren的SRI抽象机模型[35] Hausman等的VV-WAM抽象机模型[36],以及其它一些模型[37][38][39][40][41]。

1.2 逻辑程序的理论基础

1.2.1 一阶逻辑

一阶谓词逻辑提供了一种形式方法来表示某个域上的前提和结论，并处理它们之间的蕴含关系。

1. 语法

一阶逻辑包含两个方面：语法和语义。一阶逻辑的语法涉及到形式语法所认可的合式公式及证明论。它的语义涉及到与合式公式中符号相关的含义。

一阶理论由字母表、一阶语言、一个公理集合(简称一集公理)和一个推理规则集合(简称一集推理规则)组成。

下面我们将定义字母表和一阶语言。

定义 1.1 字母表由下列六类符号组成：变量符号、函数符号、谓词符号、连结词、量词和标点符号。变量符号是以大写字母开头的有限字母数字串。n 元函数符号和 n 元谓词符号是以小写字母开头的有限字母数字串(可带下标)。0 元函数符号称之为常量。连结词由～、∧、∨、→和↔组成。量词包括∃ 和∀ 。标点符号由“(”,“)”和“,”组成。

定义 1.2 一个项(term)归纳定义如下：

(1) 变量是项

(2) 常量是项

(3) 如 f 是一个 n 元函数，$t_1,\cdots,t_n$ 是项，则 $f(t_1,\cdots,t_n)$是项。

定义 1.3 一个合式的公式(或简称公式)归纳定义如下：

(1) 如果 P 是一个 n 元谓词，$t_1,\cdots,t_n$ 是项，那么

$P(t_1,\cdots,t_n)$是一个合式公式(或称原子公式)。

(2) 如 F 和 G 是合式公式，那么(～F)，(F∧G)，(F∨G)，(F→G)和(F↔G)也是合式公式。

(3) 如果 F 是合式公式，X 是一个变量，那么(∀ x F)和(∀ x F)也是合式公式。

定义 1.4 一阶语言是由字母表中的符号构造的合式公式集。

量词和连结词的非形式语义为：～是否定，∧是合取(and)，∨是析取(or)，→是蕴含，↔是等价。∃ 是存在量词，∃ X 意为“存在一个 X”，∀ 是全称量词，∀ X 意为“所有的 X”。

∀x((p(x,g(x)))←q(x)∧～r(x))的非形式语义为“对于所有的 x，如果 q(x)为真且 r(x)是假，那么 p(x,g(x))为真”。

定义 1.5 公式∀ xF 中∀ x(或公式∃ xF 中∃ x)的作用域是 F。如果一个变量在量词的作用域中出现，则称此变量是约束出现；否则称此变量是自由出现。

例如：公式(∃ x)(x≠f(a)↔～r(x,f(a)))中∃ x 的作用域是公式(x≠f(a)↔～r(x,f(a)))。

定义 1.6 无自由变量出现的公式称为闭公式。

定义 1.7 如 F 是一个公式，对于 F 中的每一个自由出现的变量，加一个全称量词得到∀(F)，它称为 F 的全称闭包。类似地对于 F 中的每一个自由出现的变量，加一个存在量

词得到∃(F),它称为 F 的存在闭包。

2. 解释和模型

当我们讨论公式的真假时,必须首先赋予公式中每一个符号某种含义。量词和连结词有固定的含义,但常量、函数和谓词的含义可以变化。

定义 1.8 一阶语言 L 的解释由下列组成:

(1) 一个非空集 D,称为解释域。

(2) 对于 L 中的每一个常量,赋予 D 中的一个元素。

(3) 对于 L 中的每一个 n 元函数,赋予 D^n 到 D 的一个映射。

(4) 对于 L 中的每一个 n 元谓词,赋予 D^n 到{true,false}的一个映射。

定义 1.9 设 L 是一阶语言 L 的解释,将 L 中的每一个变量赋予 I 中的一个元素,这样的赋值称为关于 I 的变量赋值。

定义 1.10 设 I 是一阶语言 D 域上的一个解释。A 是一个变量赋值。L 中关于 I 的项赋值定义如下:

(1) 根据 A 赋值每一个变量。

(2) 根据 I 赋值常数。

(3) 如 $t_1',\cdots,t_n'$ 是 $t_1,\cdots,t_n$ 的项赋值,f′ 是 f 的赋值,则 $f'(t_1',\cdots,t_n')\in D$ 是 $f(t_1,\cdots,t_n)$的项赋值。

定义 1.11 设 I 是一阶语言 D 域上的一个解释,A 是一个变量赋值,那么,L 中公式关于 I 和 A 的真值给定如下:

(1) 原子公式 $P(t_1,\cdots,t_n)$的真值通过求值 $P'(t_1',\cdots,t_n')$得到。其中 P′ 是根据 I 赋予 P 的映射,$t_1',\cdots,t_n'$ 是关于 I 和 A 的项赋值。

(2) 公式~F,F∧G,F∨G,F→G,F↔G 的真值由下表给出。

F	G	~F	F∧G	F∨G	F→G	F↔G
true	true	false	true	true	true	true
true	false	false	false	true	false	false
false	true	true	false	true	true	false
false	false	true	false	false	true	true

(3) 如果存在 d∈D 使得 F 关于 I 和 A(x/d)的真值为真,则公式∃ xF 的真值为真,否则为假。其中 A(x/d)与 A 的不同仅在于 A(x/d)中,x 赋予 d。

(4) 如果对于所有 d∈D,F 关于 I 和 A(x/d)的真值为真,则公式∀ xF 的真值为真,否则为假。

定义 1.12 设 I 是一阶语言 L 的解释,F 是 L 的闭公式,如果 F 关于 I 的真值为真,则 I 称为 F 的模型。

一阶理论的公理是该理论的语言中指定的闭公式子集。

定义 1.13 设 T 是一阶理论,L 是 T 的语言,如 L 的一个解释是 T 中每一个公理的模型,则称此解释为 T 的模型。

类似地,此概念可推广到一集闭公式。

定义 1.14 如果S是一阶语言L的闭公式，I是L的解释，如果I是S中每一个公式的模型，则称I是S的模型。

定义 1.15 设S是一阶语言L的闭公式集，如L有一个解释，它是S的模型，则称S是可满足的。如L的每一个解释是S的模型，称S是有效的。如S没有模型，则称S是不可满足的。

3. 逻辑结论

定义 1.16 设S是一阶语言的闭公式集，F是一个闭公式。如果对于L的每一个解释I，I是S的模型，则意味着I是F的模型，那么F称为S的逻辑结论。

命题 1.1 设S是一阶语言的闭公式集，F是一个闭公式。F是S的逻辑结论当且仅当$S\cup\{\sim F\}$是不可满足的。

定义 1.17 不含变量的项称为基础项(ground term)；不含变量的原子称为基础原子(ground atom)。

定义 1.18 设L是一阶语言，L的Herbrand通域U_L是由L中的常量和函数构造的所有基础项组成。

定义 1.19 设L是一阶语言，L中满足下列条件的解释称为Herbrand解释：

(1) 解释域为Herbrand通域U_L；

(2) L中常量赋予U_L中自身常量；

(3) 如f是L中的n元函数，f赋予$(U_L)^n$到U_L的映射。

定义 1.20 设L是一阶语言，S是L的闭公式集，如L的Herbrand解释是S的一个模型，则称它为S的Herbrand模型。

命题 1.2 设S是一集公式

(1) 如S有一个模型，则S有一个Herbrand模型；

(2) S不可满足，当且仅当S没有Herbrand模型。

4. 合一(unifier)

定义 1.21 一个置换是形如$\{V_1/t_1,\cdots,V_n/t_n\}$的有穷集。其中$V_i$为变量，$t_i$是不同于$V_i$的项，$V_i,\cdots,V_n$是不同的变量。每一个元素$V_i/t_i$称为$V_i$的一个约束。

定义 1.22 设S是形如$\{E_1,\cdots,E_n\}$的有穷简单表达式集，如果置换θ使得$E_1\theta=E_2\theta=\cdots=E_n\theta$，则$\theta$为S的合一。如果对于S的每一个合一$\sigma$，存在一个置换$\gamma$使得$\sigma=\theta\gamma$，则$\theta$是S的最一般的合一(most general unifier，mgu)。

1.2.2 Horn 子句和逻辑程序

1. 逻辑程序的语法

定义 1.23 一个文字或者是一个正原子，或者是一个负原子。

定义 1.24 一个子句是形式为$\forall x_1,\cdots,\forall x_s(L_1\vee,\cdots,\vee L_m)$的公式，其中每一个$L_i$是一个文字，$x_1,\cdots,x_s$是$L_1\vee,\cdots,\vee L_m$中出现的所有变量。

上述子句$\forall x_1,\cdots,\forall x_s(L_1\vee,\cdots,\vee L_m)$可等价变换为：

$\forall x_1,\cdots,\forall x_s(A_1\vee,\cdots,\vee A_m\vee\sim B_1\vee,\cdots,\vee\sim B_n)$

其中，$A_1,\cdots,A_m,B_1,\cdots,B_n$为原子，$x_1,\cdots,x_s$为这些原子中出现的所有变量。我们将上述