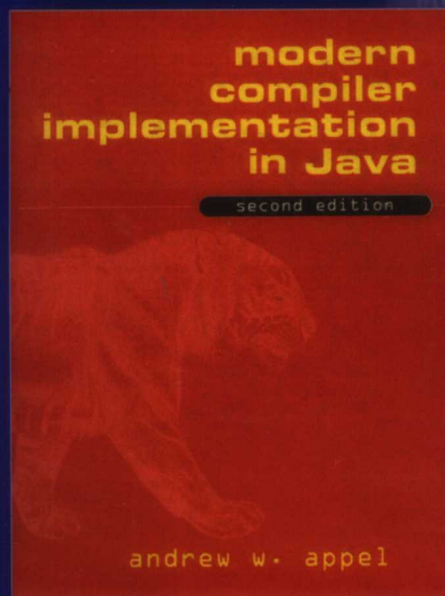


现代编译器的 Java实现

(第二版)

Modern Compiler Implementation in Java
Second Edition



[美] Andrew W. Appel 等著

陈明 等译

程虎 审校



电子工业出版社

Publishing House of Electronics Industry
<http://www.phei.com.cn>

国外计算机科学教材系列

现代编译器的 Java 实现

(第二版)

Modern Compiler Implementation in Java
Second Edition

[美] Andrew W. Appel 等著

陈 明 等译
程 虎 审校

电子工业出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本书是一本编译技术的教程,其特点是注重实现。从学习编译器的结构来掌握理论,并通过编程技术将编译理论融合于实践中。本书主要内容分为两部分,第一部分为编译基础(第1章至第12章),主要包括:词法分析、语法分析、抽象语法、语义分析、活动记录、翻译成中间代码、基本块和轨迹、指令选择、活性分析、寄存器分配、使之成为整体。第二部分为高级课题(第13章至第21章),主要包括:无用信息收集、面向对象语言、函数式编程语言、多态类型、数据流分析、循环优化、静态单赋值表、流水线和调度、分级存储器体系等。

本书可作为高等院校编译技术课程的教材、教师参考书以及编译技术研究人员的参考资料。

Authorized translation from the English language edition published by The Syndicate of the Press of the University of Cambridge, England. Copyright © Cambridge University Press 2003.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

This edition is licensed for distribution and sale in the People's Republic of China only excluding Hong Kong, Taiwan and Macau and may not be distributed and sold elsewhere.

Simplified Chinese language edition published by Publishing House of Electronics Industry. Copyright © 2004.

本书中文简体专有翻译出版权由Cambridge University Press 授予电子工业出版社。其原文版权及中文翻译出版权受法律保护。未经许可,不得以任何形式或手段复制或抄袭本书内容。

本书中文简体字版仅限于在中华人民共和国境内(不包括香港、澳门特别行政区以及台湾地区)发行与销售,并不得在其他地区发行与销售。

版权贸易合同登记号 图字:01-2002-6420

图书在版编目(CIP)数据

现代编译器的Java实现(第二版)/(美)阿佩尔(Appel, A.W.)等著;陈明等译.

-北京:电子工业出版社,2004.9

(国外计算机科学教材系列)

书名原文:Modern Compiler Implementation in Java, Second Edition

ISBN 7-121-00270-1

I. 现... II. ①阿... ②陈... III. JAVA语言-程序设计-教材 IV. TP312

中国版本图书馆CIP数据核字(2004)第085772号

责任编辑:熊 健

印 刷:北京兴华印刷厂

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编:100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:23 字数:589千字

印 次:2004年9月第1次印刷

定 价:35.00元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换;若书店售缺,请与本社发行部联系。联系电话:(010)68279077。质量投诉请发邮件至zlt@phei.com.cn,盗版侵权举报请发邮件至dbqq@phei.com.cn。

出版说明

21 世纪初的 5 至 10 年是我国国民经济和社会发展的关键时期,也是信息产业快速发展的关键时期。在我国加入 WTO 后的今天,培养一支适应国际化竞争的一流 IT 人才队伍是我国高等教育的重要任务之一。信息科学和技术方面人才的优劣与多寡,是我国面对国际竞争时成败的关键因素。

当前,正值我国高等教育特别是信息科学领域的教育调整、变革的重大时期,为使我国教育体制与国际化接轨,有条件的高等院校正在为某些信息学科和技术课程使用国外优秀教材和优秀原版教材,以使我国在计算机教学上尽快赶上国际先进水平。

电子工业出版社秉承多年来引进国外优秀图书的经验,翻译出版了“国外计算机科学教材系列”丛书,这套教材覆盖学科范围广、领域宽、层次多,既有本科专业课程教材,也有研究生课程教材,以适应不同院系、不同专业、不同层次的师生对教材的需求,广大师生可自由选择和自由组合使用。这些教材涉及的学科方向包括网络与通信、操作系统、计算机组织与结构、算法与数据结构、数据库与信息处理、编程语言、图形图像与多媒体、软件工程等。同时,我们也适当引进了一些优秀英文原版教材,本着翻译版本和英文原版并重的原则,对重点图书既提供英文原版又提供相应的翻译版本。

在图书选题上,我们大都选择国外著名出版公司出版的高校教材,如 Pearson Education 培生教育出版集团、麦格劳-希尔教育出版集团、麻省理工学院出版社、剑桥大学出版社等。撰写教材的许多作者都是蜚声世界的教授、学者,如道格拉斯·科默(Douglas E. Comer)、威廉·斯托林斯(William Stallings)、哈维·戴特尔(Harvey M. Deitel)、尤利斯·布莱克(Uyless Black)等。

为确保教材的选题质量和翻译质量,我们约请了清华大学、北京大学、北京航空航天大学、复旦大学、上海交通大学、南京大学、浙江大学、哈尔滨工业大学、华中科技大学、西安交通大学、国防科学技术大学、解放军理工大学等著名高校的教授和骨干教师参与了本系列教材的选题、翻译和审校工作。他们中既有讲授同类教材的骨干教师、博士,也有积累了几十年教学经验的老教授和博士生导师。

在该系列教材的选题、翻译和编辑加工过程中,为提高教材质量,我们做了大量细致的工作,包括对所选教材进行全面论证;选择编辑时力求达到专业对口;对排版、印制质量进行严格把关。对于英文教材中出现的错误,我们通过与作者联络和网上下载勘误表等方式,逐一进行了修订。

此外,我们还将与国外著名出版公司合作,提供一些教材的教学支持资料,希望能为授课老师提供帮助。今后,我们将继续加强与各高校教师的密切联系,为广大师生引进更多的国外优秀教材和参考书,为我国计算机科学教学体系与国际教学体系的接轨做出努力。

电子工业出版社

教材出版委员会

- | | | |
|----|-----|---|
| 主任 | 杨芙清 | 北京大学教授
中国科学院院士
北京大学信息与工程学部主任
北京大学软件工程研究所所长 |
| 委员 | 王 珊 | 中国人民大学信息学院院长、教授 |
| | 胡道元 | 清华大学计算机科学与技术系教授
国际信息处理联合会通信系统中国代表 |
| | 钟玉琢 | 清华大学计算机科学与技术系教授
中国计算机学会多媒体专业委员会主任 |
| | 谢希仁 | 中国人民解放军理工大学教授
全军网络技术研究中心主任、博士生导师 |
| | 尤晋元 | 上海交通大学计算机科学与工程系教授
上海分布计算技术中心主任 |
| | 施伯乐 | 上海国际数据库研究中心主任、复旦大学教授
中国计算机学会常务理事、上海市计算机学会理事长 |
| | 邹 鹏 | 国防科学技术大学计算机学院教授、博士生导师
教育部计算机基础课程教学指导委员会副主任委员 |
| | 张昆藏 | 青岛大学信息工程学院教授 |

译 者 序

编译技术已出现 40 余年，一直是计算机科学中十分重要的研究领域。编译技术的产生与发展推动了计算机科学的发展。

编译原理与技术是计算机科学与技术专业中十分重要的课程，对于一名计算机科学与技术的学者和工作者来说，编译原理与技术将长期被使用。但是，由于这门课程涉及到形式语言与自动机理论、程序设计语言和计算机算法设计与分析等多门课程的知识，因此，编译原理与技术也成为大学阶段中最难学习的课程之一。

本书是一本编译技术的教材，主要介绍编译器构造的一般原理、基本设计方法和基于 Java 语言的实现技术。本书的特点是注重实现，从学习编译器的结构来掌握理论，并通过编程技术将编译理论融合于实践中。本书主要内容分为两部分，第一部分为编译基础，主要包括：词法分析、文法分析、抽象语法、语义分析、活动记录、翻译成中间代码、基本块和轨迹、指令选择、活性分析、寄存器分配、使之成为整体。第二部分为高级课题，主要包括：无用信息收集、面向对象语言、函数式编程语言、多态类型、数据流分析、循环优化、静态单赋值表、流水线和调度、存储器层次等。本书可作为高等院校编译技术课程的教材、教师参考书以及编译技术研究人员的参考资料。

为使接口和编程实例更清晰正确，本书采用 Java 语言来编写。Java 语言是当前流行的跨平台语言，其精髓是复用性强。同时，每一章都包含了相关的习题，并给出了难度说明。

本书的两位作者分别是普林斯顿大学计算机科学系教授 Andrew W.Appel 和普度大学计算机科学系副教授 Jens Palsberg，他们都从事编译器的研究和教学工作并颇有建树。

本书由陈明、马晓丽、王振武、王秀文、高燕、孙成云、孙修虎、鹿扬、张世博、徐东燕、陈清爽、田登山、白强、宋立军、王玉亭、陈琳、张琳、刘志德、亓雪冬、苏钢等人翻译并校对。

中国科学院软件研究所程虎研究员十分认真地审阅了本书的全部译稿，并提出了许多宝贵意见，在此表示衷心的感谢。

尽管我们尽了很大的努力，但限于专业和英语水平，书中难免出现不妥之处，敬请读者批评指正。

前言

通过本书的学习，学生将了解到编译器不同组成部分之间所蕴涵的原理、编程技巧以及用于模块化编译器的接口。为了使接口和编程实例更加清晰，本书采用了 Java 语言加以编写。

项目实施：我们所强调的“学生项目编译器”是相当简单的，其目的是为了示范一些重要的、常用的技术，其中包括：用来避免语法和语义混乱的抽象语法树、寄存器分配时指令选择的分离、增加编译器早期阶段灵活度的复制传播，以及目标机器依赖的“约束”原则。与其他教材中的学生编译器不同的是，本编译器具有简单而灵活的后端，并允许在指令选择后实现寄存器的优化分配。

本书的第二版包括一个重新设计的编译器。它使用了 Java 的子集——MiniJava 作为编译器的源语言，解释了语法分析生成器 JavaCC 和 SableCC 的用途，并且支持带有访问模式的编程技术。因此，学生可以利用他们所熟悉的语言及标准工具，并以一种面向对象的风格实现一个编译器。

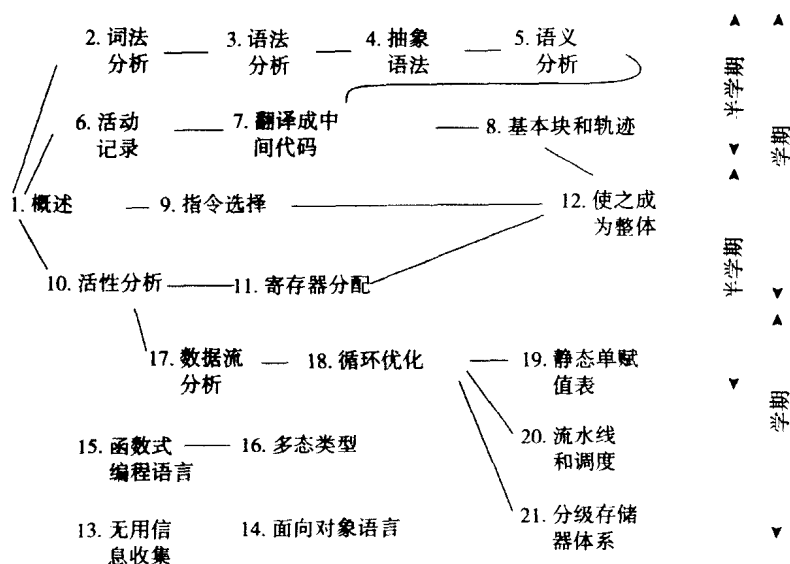
第一部分中的每一章都包含了相应的编程习题，它们对应着编译器的一个模块。习题中所使用的软件可在下列地址中找到：

<http://uk.cambridge.org/resources/052182060X>（北美以外地区）

<http://us.cambridge.org/titles/052182060X.html>（北美地区）

习题：标有一个星号的习题表示该习题包含一些挑战性；标有两个星号的习题表示该习题存在相应的难度但仍可以加以解决；偶尔出现的、标有三个星号的习题是一些尚没有解决方法的问题。

课程顺序：下图表示了各章之间的联系。



- 第一部分（第 1 章至第 12 章）包含了一个学期的课程，以便学生实现一个项目编译器（或者以项目组的方式）。另外，课程将包括第二部分中某些精选的主题。
- 第二部分（第 13 章至第 21 章）包含了高级课题或研究生课程，以及当前文献中一些附加的主题。第二部分中的许多章节与第一部分保持独立，以便在最初的课程中使用不同教材的学生能够学习更高级的课程。
- 在两个半学期的学习中，前半学期包括第 1 章至第 8 章，后半学期包括第 9 章至第 12 章，以及第二部分中的某些章节。

致谢：在本书的出版过程中，许多人给我们提供了建设性的意见及相应的帮助。Vidyut Samanta 对于本书及其相关软件给予了非常大的帮助。同时，我们也要感谢 Leonor Abraido-Fandino, Scott Ananian, Nils Andersen, Stephen Bailey, Joao Cangussu, Maia Ginsburg, Max Hailperin, David Hanson, Jeffrey Hsu, David MacQueen, Torben Mogensen, Doug Morgan, Robert Netzer, Elma Lee Noah, Mikael Petterson, Benjamin Pierce, Todd Proebsting, Anne Rogers, Barbara Ryder, Amr Sabry, Mooly Sagiv, Zhong Shao, Mary Lou Soffa, Andrew Tolmach, Kwangkeun Yi, Kenneth Zadeck。

目 录

第一部分 编译基础

第1章 概述	1
1.1 模块及接口	1
1.1.1 阶段的描述	2
1.2 工具和软件	3
1.3 树型语言的数据结构	3
程序设计：直线程序解释器	6
习题	8
第2章 词法分析	9
2.1 词法记号	9
2.2 正则表达式	10
2.3 有限自动机	12
2.3.1 识别最长的匹配	14
2.4 非确定有限自动机	14
2.4.1 正规文法转换为 NFA	15
2.4.2 NFA 转换为 DFA	16
2.5 词法分析生成器	19
2.5.1 JAVACC	19
2.5.2 SableCC	20
程序设计：词法分析	20
进一步阅读	21
习题	21
第3章 语法分析	24
3.1 上下文无关文法	25
3.1.1 推导	26
3.1.2 分析树	27
3.1.3 二义性文法	27
3.1.4 文件结束符	29
3.2 预测分析	29
3.2.1 FIRST 和 FOLLOW 集	30
3.2.2 构造一个预测分析器	33
3.2.3 消除左递归	33

3.2.4 左因子.....	35
3.2.5 出错恢复.....	35
3.3 LR 分析.....	36
3.3.1 LR 分析器.....	38
3.3.2 LR(0)分析器生成器.....	38
3.3.3 SLR 分析器生成器.....	41
3.3.4 LR(1)项目和 LR(1)分析表.....	42
3.3.5 LALR(1)分析表.....	44
3.3.6 文法类型的层次结构.....	44
3.3.7 二义性文法的 LR 分析.....	45
3.4 使用分析器生成器.....	45
3.4.1 JAVACC	46
3.4.2 SableCC	47
3.4.3 算符优先分析法.....	49
3.4.4 语法和语义.....	51
3.5 出错恢复.....	52
3.5.1 用 error 符号恢复	52
3.5.2 全局出错修复.....	53
程序设计: 实现分析器.....	55
进一步阅读.....	55
习题.....	55
第 4 章 抽象语法	59
4.1 语义分析.....	59
4.1.1 递归下降.....	59
4.1.2 自动生成分析器.....	60
4.2 抽象分析树.....	61
4.2.1 位置.....	64
4.3 访问者.....	64
4.3.1 MiniJava 的抽象语法	67
程序设计: 抽象语法.....	70
进一步阅读.....	70
习题.....	70
第 5 章 语义分析	71
5.1 符号表.....	71
5.1.1 多重符号表.....	72
5.1.2 高效率的命令符号表.....	73
5.1.3 高效率的功能符号表.....	74
5.1.4 符号.....	75
5.2 MiniJava 的类型检查	77

5.2.1 错误处理.....	79
程序设计：类型检查.....	79
习题.....	79
第6章 活动纪录.....	81
6.1 堆栈帧.....	82
6.1.1 帧指针.....	84
6.1.2 寄存器.....	84
6.1.3 参数传递.....	85
6.1.4 返回地址.....	86
6.1.5 常驻帧变量.....	86
6.1.6 静态连接.....	87
6.2 MiniJava 语言编译器中的帧.....	88
6.2.1 帧的表示.....	90
6.2.2 局部变量.....	91
6.2.3 临时（局部）变量和标号.....	92
6.2.4 静态连接的管理.....	92
程序设计：帧.....	92
进一步阅读.....	93
习题.....	93
第7章 翻译成中间代码.....	96
7.1 中间树.....	96
7.2 树的翻译.....	98
7.2.1 表达式的类型.....	98
7.2.2 简单变量.....	101
7.2.3 数组变量.....	102
7.2.4 结构化的 L-值.....	103
7.2.5 下标和域选择.....	103
7.2.6 关于安全性.....	104
7.2.7 算术运算.....	105
7.2.8 条件.....	105
7.2.9 字符串.....	106
7.2.10 记录和数组的创建.....	106
7.2.11 while 循环.....	108
7.2.12 for 循环.....	108
7.2.13 函数调用.....	109
7.2.14 静态连接.....	109
7.3 声明.....	110
7.3.1 变量定义.....	110
7.3.2 函数定义.....	110

7.3.3 段	111
7.3.4 类和对象	112
程序设计: 翻译为树	113
习题	113
第 8 章 基本块和轨迹	115
8.1 规范树	116
8.1.1 ESEQ 的翻译	116
8.1.2 常用重写规则	117
8.1.3 将 CALL 移至顶部	119
8.1.4 语句的线性表	119
8.2 时间条件分支	120
8.2.1 基本块	120
8.2.2 轨迹	121
8.2.3 完成	121
8.2.4 最佳轨迹	122
进一步阅读	122
习题	122
第 9 章 指令选择	124
9.1 指令选择的算法	126
9.1.1 maximal munch 算法	127
9.1.2 动态编程	128
9.1.3 树的文法规则	129
9.1.4 快速匹配	131
9.1.5 表示算法的效率	131
9.2 CISC 机	132
9.3 MiniJava 编译器中的指令选择	134
9.3.1 抽象的汇编语言指令	135
9.3.2 生成汇编指令	137
9.3.3 过程调用	139
9.3.4 如果不存在帧指针	139
程序设计: 指令选择	140
进一步阅读	142
习题	143
第 10 章 活性分析	144
10.1 数据流的解	145
10.1.1 活性的计算	146
10.1.2 集合的表示	147
10.1.3 时间复杂度	148

10.1.4 最少的确定点	148
10.1.5 静态与动态活性	149
10.1.6 干扰图	150
10.2 MiniJava 编译器中的活性分析	151
10.2.1 图	151
10.2.2 控制流图	152
10.2.3 活性分析	153
程序设计: 构造流图	154
程序设计: 活性	154
习题	154
第 11 章 寄存器分配	156
11.1 简化着色	156
11.1.1 举例	157
11.2 结合	158
11.2.1 溢出	160
11.3 预着色节点	161
11.3.1 机器寄存器临时变量的复制	161
11.3.2 调用保存寄存器和被调用保存寄存器	161
11.3.3 预着色节点的例子	162
11.4 图着色实现	165
11.4.1 数据结构	166
11.4.2 不变量	167
11.4.3 程序代码	167
11.5 树的寄存器分配	172
程序设计: 图着色	175
进一步阅读	176
习题	176
第 12 章 使之成为整体	179
程序设计 进入/退出过程	179
程序设计: 使程序运行	181

第二部分 高级课题

第 13 章 无用信息收集	182
13.1 标记—清除收集机制	183
13.2 引用计数	186
13.3 复制收集	187
13.4 世代收集	190
13.5 增量收集	192
13.6 Baker 算法	193

13.7 编译器接口	194
13.7.1 快速分配	194
13.7.2 数据分布描述	195
13.7.3 派生指针	195
程序设计: 描述符	196
程序设计: 无用信息收集	197
进一步阅读	197
习题	198
第 14 章 面向对象语言	200
14.1 类扩展	200
14.2 数据字段的单继承	201
14.2.1 方法	201
14.3 多继承	202
14.4 类成员测试	204
14.5 私有字段成员和方法	207
14.6 无类语言	207
14.7 优化面向对象程序	207
程序设计: 带类扩展的 MiniJava	208
进一步阅读	208
习题	209
第 15 章 函数式编程语言	211
15.1 一种简单的函数式语言	211
15.2 闭包	213
15.2.1 堆分配激活记录	213
15.3 恒变量	214
15.3.1 基于连续的 I/O	216
15.3.2 语言变换	216
15.3.3 纯函数式语言的优化	217
15.4 内部扩展	218
15.5 闭包转换	224
15.6 有效尾部递归	226
15.7 惰性评估	227
15.7.1 按名调用评估	228
15.7.2 按需调用	229
15.7.3 一个惰性程序的计算	230
15.7.4 推高不变量	230
15.7.5 惰性函数式程序的优化	231
15.7.6 严格性分析	233
进一步阅读	235

程序设计: 编译函数式语言	236
习题	236
第 16 章 多态类型	237
16.1 参数多态	237
16.2 多态类型检查	240
16.3 多态程序的翻译	244
16.3.1 指针、整型和包装	245
16.4 静态重载的解决方法	246
进一步阅读	246
习题	247
第 17 章 数据流分析	248
17.1 流分析的中间表示	249
17.1.1 四元组	249
17.2 多种的数据流分析	250
17.2.1 到达定义	251
17.2.2 可用表达式	252
17.2.3 到达表达式	253
17.2.4 活性分析	253
17.3 使用数据流分析的变换	254
17.3.1 公用子表达式消除	254
17.3.2 常量传播	254
17.3.3 复制传播	254
17.3.4 死代码消除	255
17.4 加快数据流分析	255
17.4.1 位向量	255
17.4.2 基本块	255
17.4.3 节点排序	256
17.4.4 use-def 和 def-use 链	257
17.4.5 work-list 算法	257
17.4.6 增量式数据流分析	258
17.5 别名分析	261
17.5.1 基于类型的别名分析	262
17.5.2 基于流的别名分析	262
17.5.3 使用 may-alias 信息	264
17.5.4 严格纯函数式语言中的别名分析	265
进一步阅读	265
习题	265
第 18 章 循环优化	267
18.1 必经节点	269

18.1.1 寻找必经节点的算法	269
18.1.2 直接必经节点	270
18.1.3 循环.....	271
18.1.4 循环前置首部	272
18.2 循环不变量的计算.....	272
18.2.1 提升.....	273
18.3 归纳变量.....	274
18.3.1 归纳变量检查	275
18.3.2 强度削减.....	276
18.3.3 消除.....	277
18.3.4 重写比较.....	277
18.4 数组边界检查.....	278
18.5 循环展开.....	281
进一步阅读.....	282
习题.....	282
第 19 章 静态单赋值表	284
19.1 转化为 SSA 表.....	286
19.1.1 插入 ϕ -function 的准则.....	286
19.1.2 必经前端.....	287
19.1.3 插入 ϕ -function.....	289
19.1.4 变量重命名	290
19.1.5 边分离.....	291
19.2 必经节点树的有效计算.....	291
19.2.1 深度优先生成 (Spanning) 树	291
19.2.2 半必经节点	292
19.2.3 Lengauer-Tarjan 算法	293
19.3 采用 SSA 优化算法.....	296
19.3.1 消除死代码	296
19.3.2 简单常量传播	297
19.3.3 条件常量复制	298
19.3.4 保存必经性质	300
19.4 数组、指针和存储.....	301
19.4.1 存储相关.....	301
19.5 控制相关图.....	302
19.5.1 积极的死代码消除	304
19.6 从 SSA 表后的转换.....	305
19.6.1 关于 SSA 的活性的分析	305
19.7 函数式中介表.....	306
进一步阅读.....	309

习题.....	310
第 20 章 流水线和调度	313
20.1 不受资源限制的循环调度.....	315
20.2 资源限制循环流水线.....	318
20.2.1 模调度.....	319
20.2.2 发现最小启动间隔	320
20.2.3 其他控制流	322
20.2.4 编译器应该调度指令吗	323
20.3 分支预测.....	323
20.3.1 静态转移预测	324
20.3.2 编译器应该预测分支转移吗?	324
进一步阅读.....	325
习题.....	326
第 21 章 分级存储器体系	328
21.1 高速缓冲存储器结构.....	328
21.2 cache 块的排列.....	330
21.2.1 指令 cache 的对齐.....	331
21.3 预取指令.....	332
21.4 循环交换.....	335
21.5 分块.....	336
21.6 无用信息收集和分级存储器体系.....	338
进一步阅读.....	339
习题.....	340
附录 MiniJava 语言参考手册	341
参考文献	343