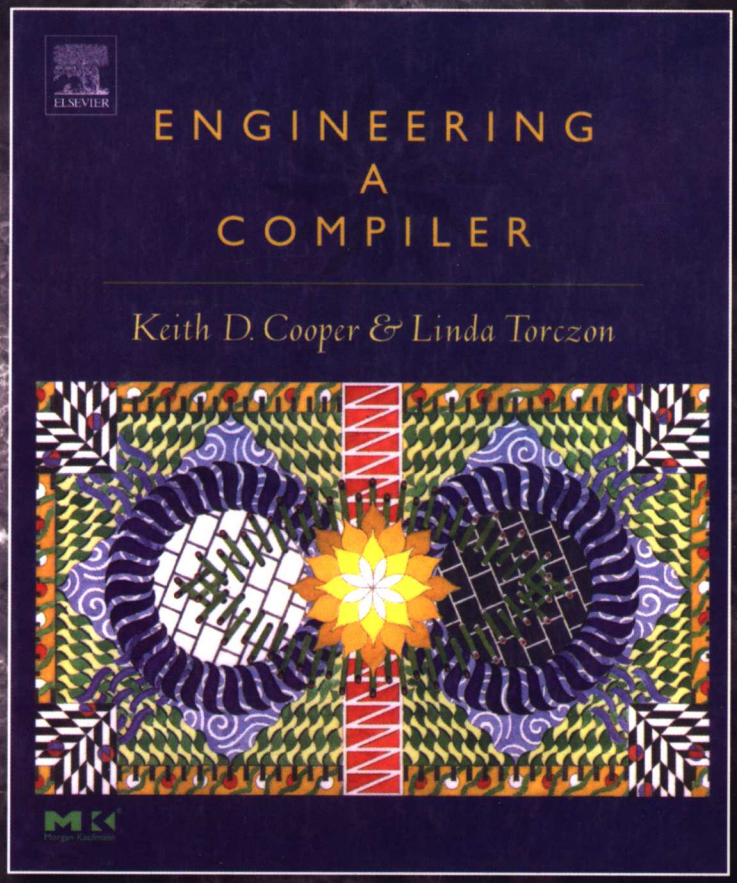
计 算 机 科 学 丛 书

编译器工程

(美) Keith D. Cooper Linda Torczon 著 冯速 译



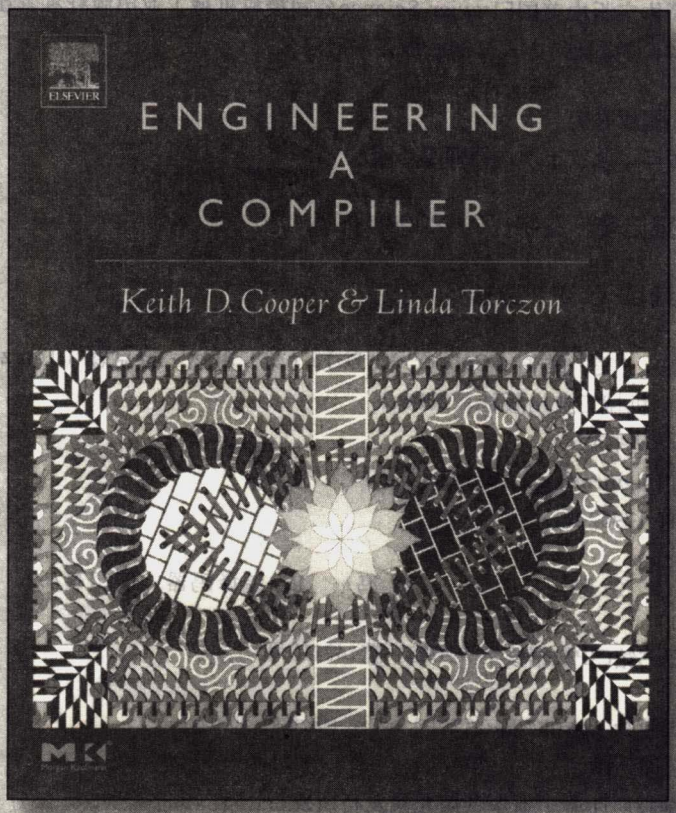
Engineering a Compiler

机械工业出版社
China Machine Press

计 算 机 科 学 丛

编译器工程

(美) Keith D. Cooper Linda Torczon 著 冯速 译



Engineering a Compiler

机械工业出版社
China Machine Press

本书旨在介绍编译器构造法中的艺术和科学。用大量素材向读者展示现实权衡的存在，展示这些选择的影响可能是微妙且深远的。省略由于商业、语言和编译器技术以及可用工具的变迁而变得不太重要的技术、C语言对优化和代码生成提供更深层次的处理。本书内容分为四部分。前端部分介绍扫描、语法分析、上下文相关分析的内容；基础结构部分阐述中间表示、过程抽象、代码形态为主线的知识；优化部分阐述构建编译器的中间部分——优化器所出现的问题；代码生成部分着眼于代码生成中的三个主要问题。

本书内容翔实，文笔流畅，适合作为高等院校计算机专业本科生和研究生编译课程的教材和参考书。

本书版权登记号：图字：01-2004-5175

图书在版编目（CIP）数据

编译器工程 /（美）酷伯（Cooper, K. D.），（美）琳达·特克森（Torczon, L.）著；冯速译. －北京：机械工业出版社，2006.2
（计算机科学丛书）
书名原文：Engineering a Compiler
ISBN 7-111-17962-5

Ⅰ. 编…　Ⅱ. ① 酷…　② 琳…　③ 冯…　Ⅲ. 编译码器　Ⅳ. TN762

中国版本图书馆CIP数据核字（2005）第142110号

机械工业出版社（北京市西城区百万庄大街22号　邮政编码　100037）
责任编辑：隋　曦
北京京北制版厂印刷·新华书店北京发行所发行
2006年2月第1版第1次印刷
787mm × 1092mm 1/16 · 32印张
印数：0 001 - 4 000册
定价：68.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
本社购书热线：（010）68326294

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭橥了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall，Addison-Wesley，McGraw-Hill，Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum，Stroustrup，Kernighan，Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及庋藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师们服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业的教学度身订造的。其中许多教材均已为M. I. T.，Stanford，U.C. Berkeley，C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设

计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

电子邮件：hzjsj@hzbook.com
联系电话：（010）68995264
联系地址：北京市西城区百万庄南街1号
邮政编码：100037

专家指导委员会

（按姓氏笔画顺序）

尤晋元	王　珊	冯博琴	史忠植	史美林
石教英	吕　建	孙玉芳	吴世忠	吴时霖
张立昂	李伟琴	李师贤	李建中	杨冬青
邵维忠	陆丽娜	陆鑫达	陈向群	周伯生
周克定	周傲英	孟小峰	岳丽华	范　明
郑国梁	施伯乐	钟玉琢	唐世渭	袁崇义
高传善	梅　宏	程　旭	程时端	谢希仁
裘宗燕	戴　葵			

对本书的赞誉

“Keith Cooper和Linda Torczon是顶级编译器研究人员，他们还创建了若干艺术品般的编译器。这本书贯穿编译器理论和实践两个领域，解释久经考验的技术和算法，并为编译器的工程化和编译器的构建提出很多切实可行的建议。《编译器工程》是对构建现代编译器必不可少的重要技术的概括和展示。”

——Jim Larus，微软研究院

“本书是对现代编译器的理论、实践和知识的完美介绍。Cooper和Torczon通过编译和计算机科学其他领域之间的优美的相互关联展示这一课题的朴素乐趣。如果你正在寻找由具有广博实践经验诠释而成的编译器构造法的完整之旅，那么这本书就是你的首选。”

——Michael D. Smith，哈佛大学

“我很高兴看到这本关于现代编译器设计的内容全面的书。作者涵盖了经典的素材，也涵盖了近15年间发展起来的重要技术，包括面向对象语言的编译、静态单一赋值、基于区域的寄存器分配以及代码调度。他们的方法把现代编译器依据的形式结构与编译器的优秀工程化所必需的实际观察完美地协调起来。”

——John Hennessy，斯坦福大学

“Cooper和Torczon做了一项完美的工作，他们集成了编译器构建理论和编译器实现的实际问题。本书所涵盖的这一领域的最新进展使得它成为向当今的本科生讲授编译器课程的理想教材或教学参考书。”

——Ken Kennedy，Rice大学

译 者 序

经过数十年的科学研究和经验积累，计算机科学已经取得了长足的发展，计算机已经普及并在各个方面得到广泛的应用。计算机技术已不再是极少数人的专利，它被众多的科学家、开发人员以及相关专业的学生和教师掌握，成为人类的有力工具。

在这样的环境下，计算机相关专业的教学也应该做相应的变革，以适应计算机科学和技术的发展。一方面，一些曾经是关键性的课程已不再是必须掌握的，一些曾经高深的技术已经唾手可得。另一方面，学生需要掌握许多新知识、新技术，而如何取舍这些新知识和新技术成为一个重要的课题。在这一问题上，本书的作者给我们提供了一个非常重要的启示。

经过20多年的研究和实践（从第一个编译器的诞生到今天已近半个世纪），编译器的前端技术已经基本成熟。特别是词法分析和语法分析的技术已经形成固定的模式，有一系列自动生成相关模块的软件，不再需要手工编写词法分析器和语法分析器。相反，代码生成和代码优化日益成为我们关心的技术，许多老技术需要我们去运用、去完善，更多的新技术需要我们去掌握、去开拓。因此，为了顺应时代的要求，同时为了学生和专业人士的各种需要，当代的编译技术的教学应该把重点放在优化和代码生成上，应该简化对编译器前端的论述。这也正是作者完成《编译器工程》这本书的首要动机。

本书一改把编译工作分解成前端和后端两个主要部分的惯例做法，把优化器作为介于前端和后端之间的第三个阶段，这样构成了典型优化编译器的工作流程。虽然计算机的性能越来越高，但人们对它的期待也越来越高，期待的提高远远超出了计算机性能的提高。因此，优化也越来越重要，并且已经成为一项必不可少的技术。

除对讲授内容做了各方面的权衡外，作者对本书的组织也非常讲究。现代编译器技术错综复杂，不存在完美的线性排列。本书通过四个部分把编译技术尽量排列成线性的顺序，以便读者轻松学习。在对编译技术进行总览之后，本书第一部分介绍编译器的前端，讲解扫描、语法分析和上下文相关分析。这里强调自动化技术和形式方法。第二部分讲解编译器中的基础结构，包括各种中间表示、过程抽象和代码形态。把贯穿这一课程的一些分散的内容组织在一起，使得本书其他部分得以依照编译器在编译时的执行顺序进行阐述。第三部分包括优化阶段，展示各种优化技术。最后的部分阐述代码生成技术，包括指令筛选、指令调度和寄存器分配的各种技术。

书中类似算法的比对是一大特色。通过比对，读者可以对类似算法有更加直观且充分的认识。另一个特点是作者把许多算法归类为不动点算法。这一方面使这些算法更加清晰，同时也使读者了解不动点算法的重要性，对算法的学习、研究有很大的帮助。

总之，顶级编译器研究人员Keith Cooper和Linda Torczon给我们带来了一本新颖、独到的编译技术参考教材。本书不仅可作为大学本科和研究生的编译技术或编译原理课程的教材，也应是计算机软件相关人员必备的一本参考书。

最后，作为讲授编译原理的教师，译者经常被学生问道：为什么要学编译原理和编译技术？我希望读者通过本书能够理解编译技术中蕴涵的思维方式，理解程序设计与开发的思维方式，理解本书的风格主题：功能、结构和精美。

译者水平有限，恳请读者不吝指正。

前　言

过去的20年间，编译器构造的实践活动发生了翻天覆地的变化。前端已变成商品组件，它们可以从可靠的供应商那里买到，或从众多自由软件的系统改写而来。与此同时，处理器变得对性能更加敏感；编译代码的实际性能很大程度上依赖于编译器针对特定处理器和系统特征进行优化的能力。这些变化影响着我们构建编译器的方式；它们也应该影响我们教授编译器构造法的方式。

今天，编译器的发展主要致力于优化和代码生成。编译器团体的新任务很可能是把代码生成器移植到新的处理器上，或修改优化遍，而不是工作于扫描器或语法分析器。为学生进入这一环境做好准备是一个真正的挑战。成功的编译器设计者必须熟知优化和代码生成中当前最实用的技术。他们还必须具有理解将出现的新技术的背景知识和洞察力。我们撰写本书（EAC, Engineering a Compiler）的目标是创作一本教科书并创建一门课程，向学生展示现代编译中的重要问题，并为他们提供解决这些问题的背景知识。

学习编译器构造法的动机

编译器构造法结合了来自计算机科学各个领域的技术。最简单地说，编译器就是一个大型计算机程序。编译器读取一个源语言的程序，并为在某个目标体系结构上执行而翻译它。作为翻译的一部分，编译器必须执行语法分析以确定输入程序是否合法。为了把输入程序映射到目标计算机的有限资源上，编译器必须操纵若干不同的名字空间，分配若干不同种类的资源，并协调多个运行时数据结构的行为。为了使输出程序有合理的性能，它必须管理功能单元中的硬件等待时间，预测执行流及内存需求，并推断出程序中的不同机器级操作的独立性和相关性。

打开一个现代优化编译器，你将发现揭示巨大求解空间的贪婪启发式搜索，识别输入中的字的确定性有穷自动机，帮助推断程序行为的不动点算法，试图推测表达式值的简单定理证明器和代数化简器，把抽象计算映射到机器级操作上的模式匹配器，用于分析数组下标的丢番图（diophantine）方程和Pressburger算术的解算器，以及诸如散列表、图算法和稀疏集合实现等经典算法和数据结构。

权衡

我们撰写EAC这本书的主要目标是创建一本用于编译器设计与实现的入门课程的教科书。EAC展示了编译器设计者需要面对的诸多问题，揭示出编译器设计者解决这些问题所用到的一些技术。EAC还提供了一系列构建现代编译器所用的实用技术的有效选择。

在EAC的选材中，为了满足学生在就业市场的需要，我们有意重新权衡了编译器构造法的入门课程的教材内容。我们减少了前端的内容而增加了优化和代码生成的内容。在后者的领域，EAC集中讨论诸如静态单一赋值形式、列表调度以及图着色寄存器分配等最实用的技术。这些课题为学生们准备了他们将来在现代商业编译器和科研编译器中所能遇到的算法。

本书还包含高年级学生或专业人士所需的内容。大多数章节都有*高级话题*一节，这一节讨论超出一般低年级课程内容所涉及的问题和技术。另外，第9章和第10章介绍比一般低年级课程所涉及的内容更深一些的数据流分析和标量优化。把这些内容包含在EAC中，使得它能够满足高年级学生或充满好奇心的学生的需要；而且专业人士也可能在他们实现某些技术时发现这些章节很有用。

途径

编译器构造法是工程设计中的一项实践。编译器设计者必须在充满不同选择，而且每一种选择都有其各自的代价、优势和复杂度的设计空间中选择一条路径。每一个决策都会对结果编译器产生影响。最终产品的质量依赖于在这一路径上的每一步明智的决策。

因此，编译器中的诸多设计决策没有惟一正确的答案。即使对于那些“著名的”和“已解决的”问题，设计和实现的细微差异都将对编译器的行为和编译器生成的代码质量产生影响。对每个决策都有很多需要考虑的问题。作为一个例子，编译器的中间表示的选择对编译器的其余部分，无论是对时间和空间的需求，还是可以运用的算法的难易程度都将产生影响。然而，这些决策通常都有不尽人意之处。第5章分析了中间表示的空间和在做出选择时应该考虑的其他问题。我们将在本书的若干地方，或是直接或在习题中间接地提及这一问题。

EAC尝试探讨设计空间问题，并讨论问题的深度以及可能的解决方案的广度。它给出解决这些问题的一些方法，以及使这些解决方案有吸引力的某些限制。学生需要理解问题和解决方案的参数，需要理解他们的决策对编译器设计其他方方面面的影响。只有这样，编译器设计者才能做出明智的决策。

哲学思想

这本教科书揭示20多年来我们通过研究、教学和实践而发展起来的关于如何构建编译器的哲学思想。例如，中间表示应该展示对最终代码至关重要的细节；这些想法导致低级中间表示的采用。值应该在寄存器中，直到分配器发现它不能把这些值保存在那里为止；这种实践引发了使用虚拟寄存器并只在无法回避时才把值存储于内存中的例子。它还增加了编译器后端中高效算法的重要性。每一个编译器都应该包含优化；它简化编译器的其他部分。

EAC不同于编译器构造法的教科书普遍接受的约定。例如，在例子中我们使用若干种不同的程序设计语言。用C语言描述名字调用参数传递没有意义，所以我们使用Algol-60。用FORTRAN语言描述尾递归没有意义，所以我们使用Scheme。这种使用多语言的方法是现实的；贯穿于读者的整个职业生涯，“未来的语言”将发生若干变化。[⊖]我们用抽象级较高的形式表示EAC中的算法。我们假设读者可以填充这些细节，并处理这些细节以适应代码运行的特定环境。

本书内容的组织

在撰写EAC时，我们的首要目标是为学生工作于真实编译器而编写一本教科书。我们已教授了10年以上本书中的内容，试验了材料的选择、深度和顺序的安排。从网站上可获得的课程内容展示了我们如何改编并在Rice大学教授学生EAC的内容。

教授现代代码生成技术的愿望使编排内容顺序的问题变得复杂。现代代码生成强烈依赖于诸如数据流分析和静态单一赋值形式等优化中的思想。这种依赖关系暗示应在覆盖后端算法之前教授优化。在讨论代码生成的内容之前覆盖优化的内容，意味着学生在尝试着改进选择语句、循环以及数组引用的代码之前是看不到这些代码的。

因为没有这些内容的完美的线性排列，所以EAC尽可能按编译器在编译时的执行顺序给出这些内容。因此，优化的内容处于前端之后却在后端之前，即使代码形态的讨论是后端的内容。章节开篇的图示起到提示这一顺序的作用。教学实践时也可以不完全按照这一顺序。

⊖ 在过去的30年间，Algol-68、APL、PL/I、Smalltalk、C、Modula-3、C++、Java，甚至是ADA都作为未来的语言一个接一个地涌现出来。

本书内容

在概述（第1章）之后，本书内容分为四个部分：

前端

第一部分的各章给出扫描、语法分析和上下文相关分析的内容。第2章介绍识别器、有穷自动机、正则表达式以及从正则表达式出发自动构建扫描器的算法。第3章使用上下文无关文法、自顶向下递归下降分析器和自底向上表驱动LR(1)分析器描述分析。第4章介绍类型系统，这是现实问题中因太复杂而无法使用上下文无关文法表示的一个例子。接着，我们给出解决这样的上下文相关问题的形式技术和专门的技术。

这些章节展示一个进展。在扫描中，自动化已经取代了手工编码。在分析中，自动化大幅度减少了程序员的工作量。在上下文相关分析中，自动化没有取代专门的手工编码方法。然而，这些专门的技术模拟一种形式技术背后的某些思想，这就是属性文法的运用。

基础结构

第二部分把贯穿这一课程的一些分散的内容组织起来。它给出在前端生成中间代码、优化这一代码以及把这一代码转换成目标机器代码所需的背景知识。

第5章描述编译器使用的各种中间表示，包括树、图、线性代码和符号表。第6章介绍编译器必须在它所生成的代码中实现的运行时抽象，包括过程、名字空间、链接约定和内存管理。第7章给出代码生成的前奏，集中讨论编译器为不同的语言结构所生成的代码形态，而不讨论生成这一代码的算法。

优化

第三部分包括构建编译器的中间部分——优化器所出现的问题。第8章通过讨论在不同作用域中出现的一个问题，引出优化的问题和技术。第9章介绍迭代数据流分析，并展示静态单一赋值形式的结构。第10章为标量优化给出一种基于效应的分类法，然后使用精选的例子探讨这一分类法。

这一内容上的划分说明，分析和优化的全面讨论可能超出了一个学期的课程，但是这样处理使本书的内容能够满足高年级和好奇心强的学生的需要。在教授本书内容时，我们在讲解第8章之后讲解代码生成。在讲解代码生成时，有必要的话我们返回第9章和第10章的特定部分。我们还使用本书的这一部分，附加若干论文节选来教授关于标量优化的进阶课程。

代码生成

最后一部分着眼于代码生成中的三个主要问题。第11章涵盖指令筛选的内容；它开始于树模式匹配，然后集中讨论窥孔风格的匹配器。第12章研究指令调度；它集中讨论列表调度及其变形。第13章描述寄存器分配；它给出局部分配和全局分配算法的较深入的论述。EAC给出的算法是学生可能发现的用于现代编译器的技术。

对于某些学生，这些章节是他们第一次要对NP完全问题给出近似解，而不是证明它等价于三可满足性问题。这些章节强调最实用的近似算法。练习为学生们提供在容易处理的例子中实践的机会。

裁剪的思想

编译器构造法是一个复杂、多层面的学科。由于编译器中信息的持续流动，对于一个问题的解决方案的选择决定后期各阶段的输入以及这些阶段改进代码的机会。对前端所做的微小变动可能隐藏优化的

机会；优化的结果对代码生成器有直接的影响（例如，改变对寄存器的需求）。编译器设计决策的复杂、相互作用的属性是这些内容通常被作为本科生顶级课程的理由之一。

这些复杂关系也出现在编译器构造法的课程中。求解技术反复出现在这一课程中。不动点算法在扫描器和语法分析器的构造中起着至关重要的作用。它们是支持优化和代码生成分析的主要工具。在扫描中出现了有穷自动机。它们在LR(1) 表驱动构造法以及指令筛选的模式匹配器中起着关键性的作用。通过强调这些常用的技术，EAC使学生熟悉它们。因此，当学生遇到第8章中的迭代数据流算法时，它只是另外一个不动点算法，因为它已是学生熟悉的内容。同样地，我们把对第12章和第13章中的局部算法到区域算法或到全局算法转换的讨论作为第8章关于优化作用域的补充。

组织课程

编译器构造法课程在具体应用的环境中为教师和学生提供揭示所有这些问题的机会，所有有编译器构造法课程知识背景的学生都能很好地理解它的基本功能。在某些课程计划中，编译器构造法课程把面向实践项目的其他课程的概念结合到一起，成为高年级学生的顶级课程。这一课程的学生可以为简单语言编写完整的编译器，或者把对一个语言的新特性的支持加到现存的编译器中，例如加到GCC或IA-64的ORC编译器中。这一课程可以严格按本书的组织以线性顺序展开教学。

如果课程计划中其他课程为学生提供大项目的实践，那么教师可以把编译器构造法课程局限在算法及其实现上。在这样的课程中，实验可以集中于真正难题的抽象实例，诸如寄存器分配和调度。这一课程可以为了满足实验的需要在内容上进行筛选，并调整授课的顺序。例如，所有学过汇编语言程序设计的学生都能够写出直线代码的寄存器分配器。我们经常把简单的寄存器分配器作为第一个实验。

无论是哪种情况，这一课程都应该从其他课程中汲取素材。和它明显相关的课程包括计算机组成原理和汇编语言程序设计、操作系统、计算机体系结构、算法以及形式语言。尽管编译器构造法与其他课程的联系并不明显，但是它们并非不重要。例如，第7章所讨论的字符拷贝在包括网络协议、文件服务器以及网络服务器在内的应用性能中起着重要的作用。第2章所开发的扫描技术在从文本编辑到URL过滤等方面均有应用。第13章中的自底向上局部寄存器分配器作为最优离线页替换算法的兄弟而得到公认。

编译器构造法中的艺术和科学

编译器构造法这一门学问中既有理论应用于实践的令人惊叹的成功故事，又有让我们束手无策的无奈。就其成功的一面，我们可以通过把正则语言的理论运用到识别器的自动构建中来构建现代扫描器。LR分析器使用相同的技术执行操纵移入归约分析器的句柄识别。数据流分析（及其兄弟）使用既实用又聪明的方式把格论运用于程序的分析。用于代码生成的近似算法对许多真正难题给出优秀的解决方案。

另一方面，编译器构造法也揭示了不存在优秀解决方案的某些复杂问题。现代超标量计算机的编译器后端必须对两个或多个相互作用的NP完全问题（指令调度、寄存器分配以及指令和数据放置）近似求解。然而，这些NP完全问题接近于诸如表达式的代数重组的问题（例如，参见图7-1）。表达式的代数重组问题有上百种解决方案；使情况变得更糟的是，理想的解决方案依赖于编译器运用的其他转换。尽管编译器尝试去解决这些问题（或近似的解决方案），但是它必须在合理的时间内运行并消耗不太多的空间。因此，现代超标量计算机的优秀编译器必须把理论、实践知识、工程学和经验艺术性地结合起来。

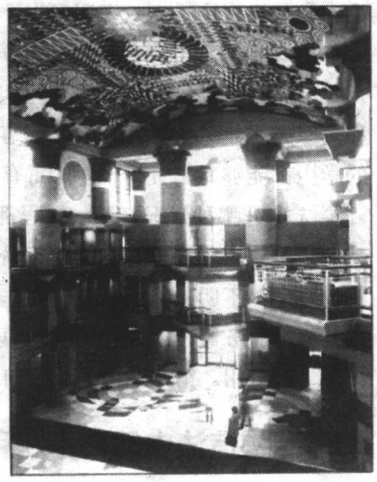
在本书中，我们尝试着传播编译器构造法中的艺术和科学。EAC包含大量的素材向读者展示现实权衡的存在，以及展示这些选择的影响可能是微妙且深远的。EAC省略那些由于商业、语言和编译器技术，或者可用工具的变迁而变得不太重要的技术。另外，EAC对优化和代码生成提供更深层次的处理。

关于原书封面

本书的封面是名画“方舟的登陆（The Landing of the Ark）”的一部分，它装饰着Rice大学的邓肯礼堂（Duncan Hall）的天花板（参看下面这幅画）。邓肯礼堂和它的天花板都是由英国设计师John Outram设计的。邓肯礼堂是Outram作为一名设计者在其职业生涯中所开拓的建筑学、装潢学以及哲学主题的外在表现。这一礼堂的天花板装潢对这一建筑物的设计方案起着重要的作用。Outram把一组寓意深刻的造物神话铭刻在天花板上。通过用巨大、色彩鲜明的寓言画面表达他的思想，Outram创造了一个标志，它告知所有进入这一礼堂的参观者这一建筑物的与众不同。

把这一相同的标志用于我们的《编译器工程》一书的封面，旨在表示这一著作所包含的重要思想是作者的学科的核心。如同Outram的建筑物一样，本书是作者职业生涯中所发展起来的知识主题的巅峰；如同Outram的装潢设计方案一样，本书是交流思想的手段；如同Outram的天花板一样，本书以新方式呈现重要的思想。

通过把编译器的设计与结构与这一建筑物的设计和结构结合在一起，我们意图展示这两个不同活动中的很多相似之处。与Outram的长期讨论把我们引入建筑的维特鲁威（Vitruvian）风格上来：实用、稳固和喜庆。这些思想被运用于很多建筑物。编译器结构也是一样，而这正是本书的一贯主题：功能、结构和精美。功能方面：生成不正确代码的编译器是无用的。结构方面：工程的细节决定了编译器的效率和稳健性。精美方面：设计良好的编译器可以是一件优美的艺术品，其中的算法和数据结构流畅地从一个遍流向另一个遍。



我们很荣幸能用John Outram的作品装饰本书的封面。

邓肯礼堂的天花板是一个有趣的技术艺术品。Outram把原始设计画在一张纸上。这幅画被拍摄下来，并用1200dpi扫描成大约750MB的数据。照片被放大成234个2 × 8英尺[⊖]的板块，制作成一张52 × 72英尺的图片。使用12dpi的丙烯酸喷墨打印机把这些板块印刷到一片片穿孔乙烯基上。这些穿孔乙烯基被裱到2 × 8英尺的吸声瓦上，而吸声瓦被它悬挂到拱顶的框架上。

致谢

很多人为EAC的形式、内容及说明提出了很有价值的反馈。他们包括L. Almagor、Saman Amarasinghe、Thomas Ball、Preston Briggs、Corky Cartwright、Carolyn Cooper、Christine Cooper、Anshuman Das Gupta、Jason Eckhardt、Stephan Ellner、Mike Fagan、Matthias Felleisen、Alex Grosul、John Greiner、Dan Grossman、Timothy Harvey、James Larus、Ken Kennedy、Shriram Krishnamurthy、Ursula Kuterbach、Robert Morgan、Guilherme Ottoni、Vishal Patel、Norm Ramsey、Steve Reeves、Martin Rinard、L.Taylor Simpson、Reid Tatge、Dan Wallach、Todd Waterman和Christian Westbrook。

Steve Carr担当练习的编辑；他与一个小组共同编写了本书的练习。他的小组成员包括 Chen Ding、Rodolfo Jardim de Azevedo、Zhiyuan Li、Guilherme Ottoni和 Sandra Rigo。Aaron Smith、Ben Hardekopf和Paul A. Navratil检查了练习的答案。本书的手稿经过了很多遍的检查和修改；Saman Amarasinghe、Guido Araujo、Preston Briggs、Steve Carr、James Larus、Gloria Melara、Kathryn McKinley、Robert Morgan、Thomas Murtagh、Gordon Novak、Santosh Pande、Allan Porterfield、

⊖ 1英尺＝0.025米。

Martin Rinard、Mark Roberts、Michael Smith和Hongwei Xi担当了审稿人。Wilson Hsieh、Jurek Jaromczyk、Tevfik Bultan、ChauWen Tseng、Mahmut Kandemir和Zhiyuan Li作为练习小组和检查小组的成员在课堂上试用了这本书。他们的工作改进了这本书，使它的风格和内容得到改善，从而更精确。

Michael Scott和Steve Muchnick反复检查了整个手稿。他们付出的细心和耐心以及他们提出的建议都使本书得到极大的改进。

Morgan Kaufmann为本书所组建的编辑、出版小组的工作也极其完美。由Denise Penrose和Yonie Overton所领导的这个小组成员包括John Hammet、Carol Leyba、Rebecca Evans、Steve Rath、Emilia Thiuri和Lauren Wheelock。Dartmouth出版公司的小组负责重新绘制本书的插图；这些图的布局是由Integra的小组完成的。作为作者，我们无法找到比他们更灵活、更富于技术及更耐心的人。他们的建议和观点大大改进了本书，他们的敬业精神使本书的出版得以顺利进行。

最后，在历时五年的时间里，很多人向我们提供了很多充满智慧和热情的支持。首先，我们的家庭和Rice大学的同事一路上给予我们鼓励。另外，Regina Brooks和Janice Bordeaux鼓励我们开始这一艰难工作。Denise Penrose和Yonie Overton使得我们能够完成这一艰巨的工作。我们由衷感谢Kathryn O'Brien以他一贯的幽默风格承担这一项目实施中的所有行政工作，并保证这一项目顺利进行。最后，Ken Kennedy和Scott Warren促使我们形成了关于程序设计、语言问题和编译的思想。

目　录