

原书第3版

计 算 机 科 学 丛 书

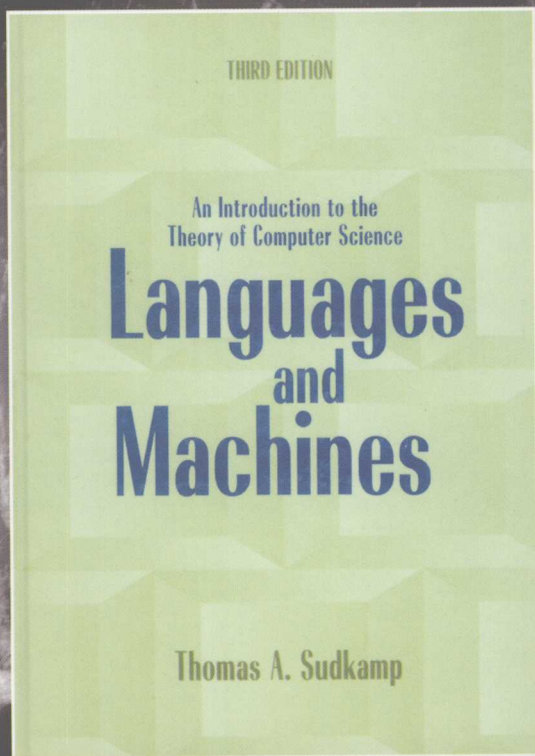
HZ BOOKS
华章教育

PEARSON
Addison
Wesley

语言与机器

计算机科学理论导论

(美) Thomas A. Sudkamp 著 孙家骥 等译



Languages and Machines
An Introduction to the Theory of Computer Science
Third Edition



机械工业出版社
China Machine Press

计 算 机

TP3/560

2008

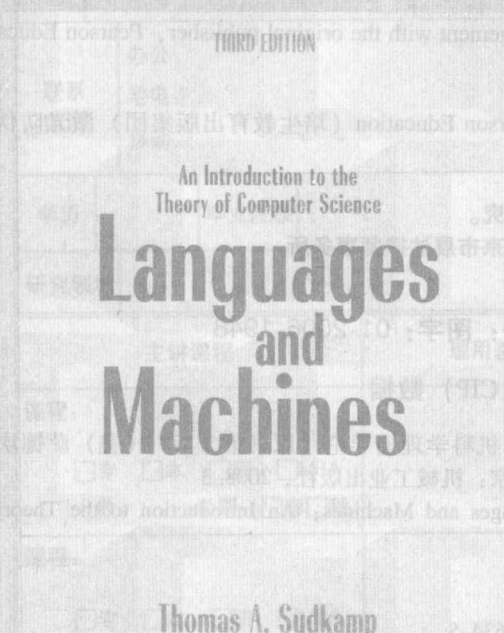
书

原书第3版

语言与机器

计算机科学理论导论

(美) Thomas A. Sudkamp 著 孙家骊 等译



Languages and Machines
An Introduction to the Theory
of Computer Science
Third Edition



机械工业出版社
China Machine Press

本书是计算理论方面的优秀教材之一,包括上下文无关文法、上下文无关文法范式、有限自动机、正则语言的性质、下推自动机和上下文无关语言、图灵机、图灵可计算函数、乔姆斯基层次、判定问题与丘奇图灵机、不可判定性、 μ -递归函数、时间复杂性、库克定理、NP-完全问题、LL(k)文法以及LR(k)文法等问题。本书不仅介绍了计算机科学的基础,而且通过概念的严格表述,以及使用通俗的例子来阐释定理,从而帮助学生提高数学论证能力以及对计算理论知识的全面深入的理解。书中每章后面都有附有大量习题,通过完成这些习题,学生可以加深对本章内容的理解。

本书可以用作计算机科学、计算机工程及其相关专业的教材,也可以作为从事计算理论、形式语言以及计算机系统研发的研究人员和工程技术人员的参考书。

Simplified Chinese edition copyright © 2008 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *Languages and Machines: An Introduction to the Theory of Computer Science, Third Edition* (ISBN 0-321-32221-5) by Thomas A. Sudkamp, Copyright © 2006 by Pearson Education, Inc.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Addison Wesley.

本书封面贴有 Pearson Education (培生教育出版集团) 激光防伪标签,无标签者不得销售。

版权所有,侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号:图字:01-2006-1948

图书在版编目(CIP)数据

语言与机器:计算机科学理论导论(原书第3版)/(美)萨德坎普(Sudkamp, T. A.)著;孙家骥等译. —北京:机械工业出版社,2008.3

书名原文:Languages and Machines: An Introduction to the Theory of Computer Science, Third Edition

(计算机科学丛书)

ISBN 978-7-111-22634-5

I. 语… II. ①萨… ②孙… III. 计算机科学 IV. TP3

中国版本图书馆CIP数据核字(2007)第164490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:李南丰

北京诚信伟业印刷有限公司印刷·新华书店北京发行所发行

2008年3月第1版第1次印刷

184mm×260mm·25.75印张

标准书号:ISBN 978-7-111-22634-5

定价:49.00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，不仅肇划了研究的范畴，还揭开了学术的源变，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、研究及度藏。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”的总体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，我们的联系方式如下：

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			

吉 祥

1990年5月

介 绍 书 籍

本书是作者多年从事计算机科学教育工作的经验总结，也是作者多年从事计算机科学教育工作的经验总结。本书共分五章，第一章介绍计算机的发展概况，第二章介绍计算机系统的组成，第三章介绍计算机系统的组成，第四章介绍计算机系统的组成，第五章介绍计算机系统的组成。本书可作为高等院校计算机专业及相关专业的教材，也可供从事计算机工作的工程技术人员参考。



译者序

理论计算机科学是推动计算机技术和应用向前发展的巨大动力。形式语言、自动机、可计算性、计算复杂性以及相关内容构成的计算理论，是理论计算机科学的基础内容之一。学习、研究这些内容，不仅为进一步学习、研究理论计算机科学打下基础，而且对增强形式化能力和推理能力也有帮助作用，这些能力对从事计算机技术中的形式化方法、技术等研究，是不可缺少的。

本书是由美国莱特州立大学计算机科学及工程系的 Thomas A. Sudkamp 教授编写的高等学校教材，主要介绍形式语言、自动机、可计算性、计算复杂性和上下文无关语言的确定性分析等内容。本书不仅介绍了计算机科学的基础，而且通过概念的严格表述，以及使用通俗的例子来阐释定理，从而期望能够帮助学生提高数学论证能力以及对计算理论知识的全面深入的理解。书中每章后面都有大量习题，通过完成这些习题，学生可以加深对本章内容的理解。本书是理论计算机科学的优秀教材之一。

本书可以用作计算机科学、计算机工程及其相关专业的教材，也可以作为从事计算理论、形式语言以及计算机系统研发的研究人员和工程技术人员的参考书。

引进国外的优秀计算机教材，对我国的计算机教育事业的发展会起到积极的推动作用，也是与世界接轨、建立世界一流大学不可缺少的条件之一。我们把本书介绍给国内从事计算机教育事业的同行们，以供参考。

参加本书翻译的有：孙家骥同志，负责各章译稿的详细修改和全书的统稿；郝丹同志负责第 1 到第 4 章及前言的翻译；刘江红同志负责第 5 到第 7 章的翻译；程邵斌同志负责第 8 到第 10 章的翻译；李琰同志负责第 11 到第 13 章的翻译；侯姗姗同志负责第 14 到第 17 章的翻译；黄晓晨同志负责第 18 到第 20 章及附录的翻译。

由于我们的能力有限，难免有不当之处，敬请读者不吝赐教。

译者

2007 年 4 月

译者简介



孙家骥 北京大学信息科学技术学院教授，博士生导师，中国计算机学会高级会员。主要研究方向：计算机语言、编译系统、程序理解、软件工程。主持和参加过多项国家科技攻关项目、“863”重大项目及与国外的合作项目；近年来发表学术论文 50 余篇，编写出版教材、翻译学术著作 10 余部。曾获北京大学优秀教学奖、机电部“七五”科技攻关先进个人荣誉证书。

前言

本书第3版的出版目的与前两版相同，即介绍计算科学理论，从而为计算机专业的初级和中级学生提供计算机科学理论的健全的数学表示。促使我对上一版进行更新的原因有三个：通过提供附加的动机介绍和例子来增强表达效果；扩大知识范围，特别是在计算复杂性这一领域；为教师讲授计算机科学理论的课程提供更多的灵活性。

许多面向应用领域的学生质疑学习理论知识的重要性。然而，正是这门课强调了计算机科学问题的宏观视图。现在的编程语言和计算机体系结构很陈旧，另外，与此相关的大家感兴趣的问题都已经找到了相应的解决方案，本书中的很多内容都与这些相关。什么类型的模式可以用某种算法识别？语言如何被形式化地定义和分析？算法计算的固有能力和局限性是什么？有哪些问题的解决方法需要的时间和空间过多以至现实中无法实现？如何比较两个问题相对的困难程度？上述问题在本书中都将逐一论述。

组织

由于绝大多数本科计算机科学专业的学生都没有或缺少足够的抽象数学的基础知识，因此本书不仅介绍计算机科学的基础，而且希望能够提高学生的数学论证能力。我们通过概念的严格表述，以及使用通俗的例子来解释一些定理，来达到上述目的。每章最后都有一些练习，从而强化和加深对本章内容的理解。

为了便于大家学习，我们假设大家都没有特殊的数学预备知识。因而，第1章介绍计算理论的数学工具：基本集合理论、递归定义以及数学归纳证明。除了1.3节和1.4节介绍的特殊内容外，第1章和第2章介绍了覆盖全书的基础知识。1.3节介绍基数和对角化。它们被用在构造不确定语言和不可计算方法的存在性证明中。1.4节检查反证法中的自引用的使用。这种技术用在不确定性证明中，包括证明停机问题无解。对于那些已经学完离散数学课程的学生，第1章的大部分内容可以看作是对这些知识的复习。

考虑到计算基础的不同课程可以强调不同的主题，本书的表述和组织方式设计为允许一门课程深入探究某些特殊主题，同时提供一种加强对主要主题的研究能力，这种能力通过介绍和深入探讨计算机科学理论研究范围的材料来获取。课程的核心内容集中于形式化的自动机语言理论的经典表示，可计算性和不确定性、计算复杂性，以及作为编程语言定义和编译器设计基础的形式化语言，详见下面表格。小节后面的星号表示这部分内容可以略过但并不影响整本书表述的连贯性。带星号的小节通常包括应用的表示、相关主题的介绍或是主题中一个复杂结论的详细证明。

形式语言和自动机理论	计算理论	计算复杂性	编程语言的形式语言
第1章：1-3, 6-8	第1章：所有	第1章：所有	第1章：1-3, 6-8
第2章：1-3, 4*	第2章：1-3, 4*	第2章：1-3, 4*	第2章：1-4
第3章：1-3, 4*	第5章：1-6, 7*	第5章：1-4, 5-7*	第3章：1-4
第4章：1-5, 6*, 7	第8章：1-7, 8*	第8章：1-7, 8*	第4章：1-5, 6*, 7
第5章：1-6, 7*	第9章：1-5, 6*	第9章：1-4, 5-6*	第5章：1-6, 7*
第6章：1-5, 6*	第10章：1	第11章：1-4, 5*	第7章：1-3, 4-5*
第7章：1-5	第11章：所有	第14章：1-4, 5-7*	第18章：所有
第8章：1-7, 8*	第12章：所有	第15章：所有	第19章：所有
第9章：1-5, 6*	第13章：所有	第16章：1-6, 7*	第20章：所有
第10章：所有		第17章：所有	

形式语言和自动机理论的经典表述考察了文法和乔姆斯基体系中抽象机之间的关系。本书同时也描述了确定型有限自动机、下推自动机、线性无界自动机和图灵机的可计算性性质。抽象机的计算能力的分析,是通过使用图灵机和无限制文法产生的语言,来识别语言的等价性而建立的。

可计算性理论考察了算法问题解决的能力和局限。可计算性包括确定性和丘吉尔—图灵理论。其中,后者通过建立图灵可计算性和 μ -递归函数的等价性而获得支持。对角化证明用来证明图灵机的停机问题是无解的,而问题化简则用来构造有关算法计算的一系列问题的不确定性。

对于计算复杂性的研究,我们首先考虑计算所需要的资源的度量方法。我们选取图灵机作为评估复杂性的框架,时间和空间的复杂性是通过图灵机计算中使用的转换数量和内存数量来计算的。 \mathcal{P} 类问题可以使用确定型图灵机在多项式时间内解决,这类问题被认为是具有有效算法解决方案的一类问题。在这之后, \mathcal{NP} 类问题和 \mathcal{NP} 完全理论也会给与介绍。逼近算法用来获得 \mathcal{NP} 完全优化问题的近似最优解。

形式化语言理论在计算科学中最重要的应用是使用文法来指定编程语言的语法。这门课程强调使用形式化技术来定义编程语言和提出有效的分解策略,它起始于用来生成语言的上下文无关文法和用于识别模式的有限自动机的介绍。介绍语言的定义之后,第18章至第20章考察了LL和LR文法以及通过这些类型的文法定义的语言的确定性分解的性质。

练习

掌握计算科学的理论基础的过程不是一场只需旁观的体育运动会。一个人只有通过解决问题,并考察主要结论的证明,才能够充分理解理论的概念、算法和精妙之处的。也就是说,对整体理解需要更多的细微努力。为了达到这一目的,我们在每章后面都编写了一些练习。这些练习包括很多方面,既有本章介绍的主题的构造练习,也有对理论的扩展。

每部分的练习中都有几个是带星号的,这些问题要比本章的其他问题更难,在本质上更理论些,或者更特别和有趣。

符号

计算科学理论是有效性计算的能力和缺陷的数学检查。与其他形式化分析类似,这种表示必须能够提供概念、结构和操作的精确且无二义性的定义。下面的表示法将贯穿本书。

条 目	描 述	例 子
元素和字符串	字符是字母表中的小写斜体字母	a, b, abc
函数	小写斜体字母	f, g, h
集合和关系	大写字母	X, Y, Z, Σ, Γ
文法	大写字母	G, G_1, G_2
文法的变量	大写斜体字母	A, B, C, S
抽象机	大写字母	M, M_1, M_2

使用罗马字母表示集合和数学结构在一定程度上不太标准,然而这样做能使得一种结构的构成成分很容易地被识别出来。例如,上下文无关文法的结构是 $G = (\Sigma, V, P, S)$ 。仅仅从字体上我们就可以看出, G 包含三个集合和变量 S 。

整本书使用三种计数系统,每章、节、条目都给出一条引用。一个计数序列记录了定义、引理、定理、推论和算法;另一个序列用来识别例子;图、表以及练习使用章节序号来标识。

证明的结尾以■标记结束,例子的结尾以□标记结束。符号索引,包括这些符号的介绍,以及它们出现的页数[⊖]都在附录I中给出。

⊖ 指英文原书的页码,而非本书页码。——编辑注

补充

我们给出的部分练习的解答，仅仅是为了辅助教师授课[⊙]。

声明

首先，我要感谢我的妻子 Janice 和女儿 Elizabeth。由于她们的善良、耐心和顾全大局，才使我能成功地完成本书。我还要感谢 the Institut de Recherche en Informatique de Toulouse 的同事和朋友。这本书修订的第一稿是在 2004 年我访问 IRIT 期间完成的。特别要感谢 Didier Dubois 和 Henri Prade 的慷慨和好客。

每一版我都会多感谢一些对本书作出贡献的人，恳切感谢所有使用这本书的学生和教师，以及那些提出批评、校勘、修改与建议，从而促进我完善本书的人。其中很多意见我已经加入到这一版中。谢谢你们花费时间把意见发给我，希望你们以后会仍然如此。我的信箱是 tsudkamp@cs.wright.edu。

这本书的不同版本曾经得到许多计算机专家的评论，其中包括下面这些教授：Andrew Astromoff (San Francisco State University)，Dan Cooke (University of Texas-EI Paso)，Thomas Fernandez，Sandeep Gupta (Arizona State University)，Raymond Gumb (University of Massachusetts-Lowell)，Thomas F. Hain (University of South Alabama)，Michael Harrison (University of California at Berkeley)，David Hemmendinger (Union College)，Steve Homer (Boston University)，Dan Jurca (California State University-Hayward)，Klaus Kaiser (University of Houston)，C. Kim (University of Oklahoma)，D. T. Lee (Northwestern University)，Karen Lemone (Worcester Polytechnic Institute)，C. L. Liu (University of Illinois at Urbana-Champaign)，Richard J. Lorentz (California State University-Northridge)，Fletcher R. Norris (The University of North Carolina at Wilmington)，Jeffery Shallit (University of Waterloo)，Frank Stomp (Wayne State University)，William Ward (University of South Alabama)，Dan Ventura (Brigham Young University)，Charles Wallace (Michigan Technological University)，Kenneth Williams (Western Michigan University) 和 Hsu-Chun Yen (Iowa State University)，在此感谢你们。

我还要感谢 Addison-Wesley 出版公司计算机科学教育分公司和 Windfall Software 中参与这个项目的小组成员的帮助。

Thomas A. Sudkamp
Dayton, Ohio

[⊙] 需要练习解答的教师请按书后的教师服务沟通表中列出的联系方式联系培生教育集团北京代表处。

绪 论

计算机科学理论起源于那些有助于促进科学进步的问题：how 和 what。回答了这两个问题之后，接着是促进许多经济决策的问题：how much。这本书的目的就是解释这些问题对于学习计算机科学的重要性，并尽可能地提供答案。

形式语言理论最初是由“语言是如何定义的”这个问题引出的。在尝试捕捉自然语言的结构和各种自然语言之间的细微区别的过程中，语言学家 Noam Chomsky 提出了一个称为文法的形式化系统，它用来定义和产生文法正确的句子。大约在同时，计算机科学家正在定义编程语言的确定性和无二义性的问题。于是，这两方面的研究相互交汇就产生了编程语言 ALGOL 的文法，这种语言使用的是一种等价于上下文无关文法的形式定义。

对可计算性这一领域的探索源于两个基本问题：“什么是算法？”和“算法计算的能力和局限性是什么？”。第一个问题的回答需要计算的形式化模型。计算机和高级编程语言的结合，可以清晰地构成一个计算系统，这似乎能为可计算性的研究提供理想的框架。但是我们还需要考虑到是方法的困难程度。使用什么样的计算机？它应该有多大的存储器？使用什么样编程语言？另外，选择一种特殊的计算机或语言可能会对第二个问题的回答带来意外的、不希望的影响。在一台已配置好的计算机上能够解决的问题，并不一定也能够在另一台计算机上解决。

一个问题是否是算法可解的，有一个算法解或者没有，应该独立于使用的计算模型。因此，我们需要一个能够进行所有可能的算法计算的系统来恰当地描述可计算性问题。通用算法计算的特点自二十世纪三十年代以来就成为了数学家和逻辑学家研究的一个主要领域了。其间，大家提出了很多不同的系统作为计算模型，包括递归函数、Alonzo 的 lambda 演算、Markov 系统和 Alan Turing 设计的抽象机。所有这些系统，以及其他以此为目的设计的系统，都能解决同样的问题集。在第 11 章中，我们给出了丘奇—图灵机的一种解释，即：一个问题有算法解当且仅当它可以被任何上述系统解决。

正因为它简单，而且其构成成分与当今的计算机相似，所以我们使用图灵机作为研究计算机的框架。图灵机与计算机有很多相似之处：它接收输入、写入内存并产生输出。虽然与计算机相比，图灵机的指令还很基本，但是，如果适当定义图灵机的指令序列，它就可以模拟计算机。尽管如此，图灵机类型还是避免了传统计算机的物理限制，即：不存在计算的时间和内存数量的上限。因此，任何计算机解决的问题都可以用图灵机解决，但反之则不然。

接受图灵机作为有限计算的统一模型之后，我们开始强调“什么是算法计算的能力和局限性？”丘奇—图灵机理论保证一个问题是可解决的，当且仅当能够设计一个合适的图灵机来解决它。于是，证明一个问题无解便简化成，无法设计一个图灵机来解决这个问题。第 12 章使用这种方法来证明几个有关我们预测计算结果的能力的重要问题是无解的。

一旦某个问题确认是可解的，那么人们就开始考虑解决这个问题时的效率和优化了。How much 这个问题初始化了计算复杂性的研究。图灵机再次提供了一个无偏颇的平台，用来比较不同问题的资源需求。图灵机的时间复杂性度量了计算需要的指令数。时间复杂性把可解决问题划分成两类：易处理的和难处理的。如果一个问题可以被图灵机解决，并且计算的执行指令数是输入的多项式函数，那么这个问题就被认为是易处理的；而不能在多项式时间内解决的问题被认为是难处理的，因为即使去解决这个问题的简单实例也需要大量的计算资源。

图灵机不但是我们考虑的惟一抽象机，而且它还是能力逐渐增强的一系列机器中的最顶尖的一个。有效计算的分析源于确定型有限自动机的性质研究。确定型有限自动机每读入一次，就要根据机器的状态和输入待处理的字符决定执行哪一条指令。尽管确定型有限自动机的状态很简单，但是它在很多方面都有应用，包括模式识别、开关电路设计和编程语言的词法分析。

另一种功能更强大的机器族，称为下推自动机，通过给有限自动机增加外部栈的存储来实现。下推自动机中增加的栈扩展了有限自动机的计算能力。由于图灵机的存在，我们对于可计算性的研究将会体现两种语言族的计算能力的特色。

语言定义和可计算性是这本书的两个主题。它们并非是计算机科学理论广阔领域中不相关的两个课题。相反，它们不可避免地纠缠在了一起。机器的可计算性可以用来识别语言；如果一个字符串在计算过程中被证明是语法正确的，那么这个字符串就可以被相应的机器接收。因此，每台机器都有相应的语言，这种语言是由被这台机器接收的字符串构成的集合。每一族抽象机的计算能力都被这族机器接收的语言来标识。记住了这一点，我们就可以深入地探讨语言的定义和有效计算的相关话题了。

第 1 章 语言定义和可计算性

1.1 语言定义和可计算性 1

1.2 正则语言 15

1.3 上下文无关语言 35

1.4 图灵可识别语言 55

1.5 图灵可计算语言 75

1.6 图灵可枚举语言 95

1.7 图灵可判定语言 115

1.8 图灵可识别语言与图灵可枚举语言 135

1.9 图灵可判定语言与图灵可识别语言 155

1.10 图灵可判定语言与图灵可枚举语言 175

1.11 图灵可判定语言与图灵可枚举语言 195

1.12 图灵可判定语言与图灵可枚举语言 215

1.13 图灵可判定语言与图灵可枚举语言 235

1.14 图灵可判定语言与图灵可枚举语言 255

1.15 图灵可判定语言与图灵可枚举语言 275

1.16 图灵可判定语言与图灵可枚举语言 295

1.17 图灵可判定语言与图灵可枚举语言 315

1.18 图灵可判定语言与图灵可枚举语言 335

1.19 图灵可判定语言与图灵可枚举语言 355

1.20 图灵可判定语言与图灵可枚举语言 375

1.21 图灵可判定语言与图灵可枚举语言 395

1.22 图灵可判定语言与图灵可枚举语言 415

1.23 图灵可判定语言与图灵可枚举语言 435

1.24 图灵可判定语言与图灵可枚举语言 455

1.25 图灵可判定语言与图灵可枚举语言 475

1.26 图灵可判定语言与图灵可枚举语言 495

1.27 图灵可判定语言与图灵可枚举语言 515

1.28 图灵可判定语言与图灵可枚举语言 535

1.29 图灵可判定语言与图灵可枚举语言 555

1.30 图灵可判定语言与图灵可枚举语言 575

1.31 图灵可判定语言与图灵可枚举语言 595

1.32 图灵可判定语言与图灵可枚举语言 615

1.33 图灵可判定语言与图灵可枚举语言 635

1.34 图灵可判定语言与图灵可枚举语言 655

1.35 图灵可判定语言与图灵可枚举语言 675

1.36 图灵可判定语言与图灵可枚举语言 695

1.37 图灵可判定语言与图灵可枚举语言 715

1.38 图灵可判定语言与图灵可枚举语言 735

1.39 图灵可判定语言与图灵可枚举语言 755

1.40 图灵可判定语言与图灵可枚举语言 775

1.41 图灵可判定语言与图灵可枚举语言 795

1.42 图灵可判定语言与图灵可枚举语言 815

1.43 图灵可判定语言与图灵可枚举语言 835

1.44 图灵可判定语言与图灵可枚举语言 855

1.45 图灵可判定语言与图灵可枚举语言 875

1.46 图灵可判定语言与图灵可枚举语言 895

1.47 图灵可判定语言与图灵可枚举语言 915

1.48 图灵可判定语言与图灵可枚举语言 935

1.49 图灵可判定语言与图灵可枚举语言 955

1.50 图灵可判定语言与图灵可枚举语言 975

1.51 图灵可判定语言与图灵可枚举语言 995

第 2 章 语言定义和可计算性

2.1 语言定义和可计算性 1

2.2 正则语言 15

2.3 上下文无关语言 35

2.4 图灵可识别语言 55

2.5 图灵可计算语言 75

2.6 图灵可枚举语言 95

2.7 图灵可判定语言 115

2.8 图灵可识别语言与图灵可枚举语言 135

2.9 图灵可判定语言与图灵可识别语言 155

2.10 图灵可判定语言与图灵可枚举语言 175

2.11 图灵可判定语言与图灵可枚举语言 195

2.12 图灵可判定语言与图灵可枚举语言 215

2.13 图灵可判定语言与图灵可枚举语言 235

2.14 图灵可判定语言与图灵可枚举语言 255

2.15 图灵可判定语言与图灵可枚举语言 275

2.16 图灵可判定语言与图灵可枚举语言 295

2.17 图灵可判定语言与图灵可枚举语言 315

2.18 图灵可判定语言与图灵可枚举语言 335

2.19 图灵可判定语言与图灵可枚举语言 355

2.20 图灵可判定语言与图灵可枚举语言 375

2.21 图灵可判定语言与图灵可枚举语言 395

2.22 图灵可判定语言与图灵可枚举语言 415

2.23 图灵可判定语言与图灵可枚举语言 435

2.24 图灵可判定语言与图灵可枚举语言 455

2.25 图灵可判定语言与图灵可枚举语言 475

2.26 图灵可判定语言与图灵可枚举语言 495

2.27 图灵可判定语言与图灵可枚举语言 515

2.28 图灵可判定语言与图灵可枚举语言 535

2.29 图灵可判定语言与图灵可枚举语言 555

2.30 图灵可判定语言与图灵可枚举语言 575

2.31 图灵可判定语言与图灵可枚举语言 595

2.32 图灵可判定语言与图灵可枚举语言 615

2.33 图灵可判定语言与图灵可枚举语言 635

2.34 图灵可判定语言与图灵可枚举语言 655

2.35 图灵可判定语言与图灵可枚举语言 675

2.36 图灵可判定语言与图灵可枚举语言 695

2.37 图灵可判定语言与图灵可枚举语言 715

2.38 图灵可判定语言与图灵可枚举语言 735

2.39 图灵可判定语言与图灵可枚举语言 755

2.40 图灵可判定语言与图灵可枚举语言 775

2.41 图灵可判定语言与图灵可枚举语言 795

2.42 图灵可判定语言与图灵可枚举语言 815

2.43 图灵可判定语言与图灵可枚举语言 835

2.44 图灵可判定语言与图灵可枚举语言 855

2.45 图灵可判定语言与图灵可枚举语言 875

2.46 图灵可判定语言与图灵可枚举语言 895

2.47 图灵可判定语言与图灵可枚举语言 915

2.48 图灵可判定语言与图灵可枚举语言 935

2.49 图灵可判定语言与图灵可枚举语言 955

2.50 图灵可判定语言与图灵可枚举语言 975

2.51 图灵可判定语言与图灵可枚举语言 995

2
3
4

目 录

出版者的话	第 4 章 上下文无关文法范式	57
专家指导委员会	4.1 文法转换	57
译者序	4.2 消去 λ 规则	58
前言	4.3 去掉链规则	62
绪论	4.4 无用符	64
	4.5 乔姆斯基范式	67
	4.6 CYK 算法	69
	4.7 去掉直接左递归	71
	4.8 格立巴赫范式	73
	4.9 练习	77
	参考文献注释	80
第一部分 基础	第 5 章 有限自动机	81
第 1 章 数学预备知识	5.1 一个有限状态自动机	81
1.1 集合论	5.2 确定型有限自动机	82
1.2 笛卡儿积、关系和函数	5.3 状态图和例子	84
1.3 等价关系	5.4 非确定型有限自动机	88
1.4 可数集合和不可数集合	5.5 λ -转换	91
1.5 对角化和自反	5.6 去掉非确定性	94
1.6 递归定义	5.7 DFA 的最小化	99
1.7 数学归纳	5.8 练习	103
1.8 有向图	参考文献注释	107
1.9 练习	第 6 章 正则语言的性质	108
参考文献注释	6.1 有限状态机接收正则语言	108
第 2 章 语言	6.2 表达式图	109
2.1 字符串和语言	6.3 正则文法和有限自动机	111
2.2 语言的有穷规格说明	6.4 正则语言的封闭性质	114
2.3 正则集合和表达式	6.5 非正则语言	115
2.4 正则表达式和文本搜索	6.6 规则语言的泵引理	116
2.5 练习	6.7 Myhill-Nerode 定理	119
参考文献注释	6.8 练习	122
	参考文献注释	125
第二部分 文法、自动机和语言	第 7 章 下推自动机和上下文无关语言	126
第 3 章 上下文无关文法	7.1 下推自动机	126
3.1 上下文无关文法和语言	7.2 PDA 的变种	129
3.2 文法和语言的例子	7.3 上下文无关语言的接收	132
3.3 正则文法	7.4 上下文无关语言的泵引理	136
3.4 验证文法	7.5 上下文无关语言的封闭性	138
3.5 最左推导和二义性		
3.6 上下文无关文法和编程语言定义		
3.7 练习		
参考文献注释		

7.6 练习	140	12.2 问题归约和不可判定性	211
参考文献注释	143	12.3 其他的停机问题	213
第三部分 可计算性			
第 8 章 图灵机	146	12.4 莱斯定理	215
8.1 标准图灵机	146	12.5 不可解决的词问题	216
8.2 作为语言接收器的图灵机	148	12.6 波斯特对应问题	218
8.3 可供选择接收标准	150	12.7 上下文无关文法中的不可判定 问题	221
8.4 多道图灵机	151	12.8 练习	223
8.5 双向图灵机	151	参考文献注释	225
8.6 多带图灵机	153	第 13 章 μ -递归函数	226
8.7 非确定型图灵机	157	13.1 原始递归函数	226
8.8 用来枚举语言的图灵机	162	13.2 一些原始递归函数	228
8.9 练习	166	13.3 有界操作符	230
参考文献注释	169	13.4 除法函数	234
第 9 章 图灵可计算函数	170	13.5 歌德尔数字和串值递归	235
9.1 函数的计算	170	13.6 可计算部分函数	237
9.2 数值计算	172	13.7 图灵可计算函数和 μ -递归函数	240
9.3 图灵机的顺序操作	174	13.8 修订的丘奇—图灵论题	243
9.4 函数的合成	178	13.9 练习	245
9.5 不可计算函数	180	参考文献注释	249
9.6 关于编程语言	181	第四部分 计算复杂性	
9.7 练习	184	第 14 章 时间复杂性	252
参考文献注释	186	14.1 复杂性度量	252
第 10 章 乔姆斯基层次	187	14.2 增长的速度	253
10.1 无限制文法	187	14.3 图灵机的时间复杂性	256
10.2 上下文有关文法	191	14.4 复杂性和图灵机的变种	259
10.3 线性有界自动机	192	14.5 线性加速	260
10.4 乔姆斯基层次	195	14.6 语言时间复杂性的属性	262
10.5 练习	195	14.7 计算机计算的模拟	266
参考文献注释	197	14.8 练习	268
第 11 章 判定问题与丘奇—图灵 论题	198	参考文献注释	270
11.1 判定问题的描述	198	第 15 章 \mathcal{P} 、 \mathcal{NP} 和库克定理	271
11.2 判定问题和递归语言	199	15.1 非确定型图灵机的时间复杂性	271
11.3 问题归约	201	15.2 \mathcal{P} 类和 \mathcal{NP} 类	272
11.4 丘奇—图灵论题	203	15.3 问题表示和复杂性	273
11.5 通用机	204	15.4 判定问题和复杂性类	275
11.6 练习	207	15.5 哈密尔顿回路问题	276
参考文献注释	208	15.6 多项式时间归约	278
第 12 章 不可判定性	209	15.7 $\mathcal{P} = \mathcal{NP}?$	279
12.1 图灵机的停机问题	209	15.8 可满足性问题	280
		15.9 复杂类的关系	287

15.10 练习 287
 参考文献注释 289
 第 16 章 NP-完全问题 290
 16.1 归约和 NP-完全问题 290
 16.2 三元可满足性问题 291
 16.3 三元可满足性的归约 292
 16.4 归约和子问题 299
 16.5 最优化问题 302
 16.6 近似算法 303
 16.7 近似方案 305
 16.8 练习 307
 参考文献注释 308
 第 17 章 其他复杂性类 309
 17.1 派生的复杂性类 309
 17.2 空间复杂性 310
 17.3 空间复杂性和时间复杂性的关系 312
 17.4 P-空间, NP-空间和萨维奇定理 315
 17.5 P-空间完全性 318
 17.6 一个难解问题 320
 17.7 练习 321
 参考文献注释 321

第五部分 确定型语法分析

第 18 章 语法分析引论 324
 18.1 文法图 324
 18.2 自顶向下语法分析 325
 18.3 归约和自底向上语法分析 328
 18.4 自底向上语法分析器 329
 18.5 语法分析和编译 331

18.6 练习 332
 参考文献注释 333
 第 19 章 LL(k)文法 334
 19.1 上下文无关文法中的预读 334
 19.2 FIRST 集合、FOLLOW 集合和预读集合 336
 19.3 强 LL(k)语法 338
 19.4 FIRST_k 集合的构造 339
 19.5 FOLLOW_k 集合的构造 341
 19.6 强 LL(1)文法 342
 19.7 强 LL(k)分析器 344
 19.8 LL(k)文法 345
 19.9 练习 346
 参考文献注释 348
 第 20 章 LR(k)文法 349
 20.1 LR(0)上下文 349
 20.2 LR(0)分析器 351
 20.3 LR(0)机 352
 20.4 被 LR(0)机接收 356
 20.5 LR(1)文法 360
 20.6 练习 365
 参考文献注释 366

附录 I 标记索引 367
 附录 II 希腊字母表 370
 附录 III ASCII 字符集 371
 附录 IV Java 的 BNF 范式定义 372

参考文献 379
 索引 384

第一部分

基础

理论计算科学包括语言定义、模式识别、算法计算的能力和局限性、问题的复杂性分析以及它们的求解研究。这些问题都是以集合论和离散数学为基础的。第 1 章复习形式语言理论和计算理论中需要的数学概念、操作和表示符号。

形式语言理论源于语言学、数理逻辑和计算机科学。第 2 章我们将给出语言的集合论定义。这个定义很充分，因此这个定义涵盖了自然（口头和书面）语言和形式语言，但是这种通用性是以牺牲提供生成语言字符串的有效方法为代价的。为了克服这个缺点，递归定义和集合操作用来给出语言的有限规约说明。接着，我们介绍正则集合，这是出现在自动机理论、形式语言理论、开关电路和神经网络中的语言族。在第 2 章的结尾还给出一个使用正则表达式的实例——正则集合的简化表示——用来描述文本搜索的模式。

5
6

集合论

集合论是数学的一个分支，研究集合的性质。集合是由一些确定的、互不相同的对象组成的整体。集合中的对象称为元素。集合的表示方法有列举法和描述法。集合的运算包括并集、交集、差集、补集、对称差、笛卡尔积、幂集等。集合论在计算机科学、语言学、逻辑学等领域有广泛的应用。

$$\{c, s, t\} = X$$

$$\{a, b, s, d, u\} = Y$$

集合的运算包括并集、交集、差集、补集、对称差、笛卡尔积、幂集等。集合论在计算机科学、语言学、逻辑学等领域有广泛的应用。

$$\{x \mid x \text{ 是自然数且 } x \leq 10\}$$

集合的运算包括并集、交集、差集、补集、对称差、笛卡尔积、幂集等。集合论在计算机科学、语言学、逻辑学等领域有广泛的应用。

集合的运算包括并集、交集、差集、补集、对称差、笛卡尔积、幂集等。集合论在计算机科学、语言学、逻辑学等领域有广泛的应用。

第1章 数学预备知识

集合论和离散数学为形式语言理论、可计算性理论和计算复杂性分析提供了数学基础。我们首先回顾集合论的表示和基本操作。集合的基数度量集合的大小，并提供无穷集合大小的准确定义。德国数学家 George Cantor 深入研究集合的属性后得出一条有趣的结论，就是存在不同大小的无穷集。尽管 Cantor 的工作仅表明存在一个完整的无穷集规模层次，但是这已经足够支持我们把无穷集合分成两类的目的了。这两类分别是可数的和不可数的。如果集合的元素数目与自然数一样多，那么这个集合是可数的无穷集。如果元素数目比自然数多，就是不可数无穷集。

在本章中，我们将使用对角化论证 (diagonalization argument) 结构来证明定义在自然数集合上的函数集合是不可数无穷集。我们在有效过程 (effective procedure) 和可计算函数 (computable function) 的意义上达成共识后 (这也是本书第三部分的主要目的)，将能够确定可以用算法计算的函数集合的大小。通过比较这两个集合的大小，就可以证明存在这样的函数，它们的值不能使用任何算法过程计算得到。

一个集合可能由任意一组对象组成，我们对那些机械化生成元素的集合感兴趣。然后，我们介绍可以产生集合元素的递归定义；接着构造递归生成的集合与数学归纳法之间的关系。归纳已经被证明能够为递归产生的无穷集合中的元素性质提供一个通用的证明技巧。

在本章的最后，我们将复习有向图和树等知识，这是贯穿本书的两种结构，并以图形方式的解释了形式语言理论和计算理论的概念。

1.1 集合论

我们假设读者熟悉初等集合论的表示。在这节中，我们主要回顾这个理论的概念和记号。符号 \in 表示成员资格； $x \in X$ 表示 x 是集合 X 的成员或元素。带斜线的符号表示否， $x \notin X$ 即表示 x 不是 X 的一个成员。如果两个集合包含相同的成员，那么这两个集合相等。在本书当中，我们使用大写字母表示集合。特别地， X 、 Y 和 Z 用来表示任意集合。集合的元素用斜体表示。例如， a 、 b 、 A 、 B 、 $aaaa$ 和 abc 的字符和字符串形式表示集合的元素。

括号 $\{\}$ 用来给出集合的定义。集合中成员个数很少的时候，可以直接给出集合的定义，也就是把集合的元素列出来。集合

$$X = \{1, 2, 3\}$$
$$Y = \{a, b, c, d, e\}$$

就是使用显式的方式定义的。如果集合有大量有限个元素，或无穷数目的元素时，就必须使用隐含方式给出定义。集合的隐式定义是通过描述集合中元素的指定条件给出的。例如，包含所有的完全平方的集合定义为

$$\{n \mid n = m^2, m \text{ 为任意自然数}\},$$

隐式定义中竖线 \mid 读作“满足”。整个定义读作“ n 的集合，满足 n 是某个自然数 m 的平方。”

上面的例子中提及了自然数集合 (natural numbers)。这个重要的集合，用 \mathbb{N} 表示，包括 $0, 1, 2, 3, \dots$ 。空集 (empty set) 用 \emptyset 表示，这个集合里没有成员，也可以显式定义为 $\emptyset = \{\}$ 。

集合由它的全体成员完全确定。定义中元素的表示顺序并不重要。显式定义

$$X = \{1, 2, 3\}, Y = \{2, 1, 3\}, Z = \{1, 3, 2, 2, 2\}$$

表示同一个集合。Z 的定义中包含数字 2 的多个实例。集合中的重复定义并不影响其成员资格。集合相等要求集合具有相同的成员，如上面这个例子，集合 X 、 Y 和 Z 都有自然数 1、2 和 3 作为成员。