JISUAN FUZAXING LILUN JICHU

计算复杂性理论基础

吕克伟 编著

国防工业出版社
National Defense Industry Press

计算复杂性理论基础

吕克伟　编著

国防工业出版社
·北京·

内容简介

计算复杂性理论是用数学方法研究计算机解决各种算法问题难易程度的理论。本书对这一理论的基础知识做了全面介绍，力争帮助读者掌握该理论的思想方法，为进一步开展计算机科学的相关领域的学习和研究奠定了基础。本书首先介绍计算复杂性理论的概述、一些计算问题和逻辑，然后详细介绍计算模型、P vs NP 问题、归约和 NP 完备性理论等；接着针对信息安全专业特点，详细介绍随机化算法、(非)一致电路；最后简单介绍几个较深入的课题：交互语言类、计数复杂类、概率可验证语言类等。

本书不仅适合作为计算机科学各专业高年级本科生和低年级研究生(特别是信息安全专业)基础课教材，也可供有关研究人员参考。

图书在版编目(CIP)数据

计算复杂性理论基础/吕克伟，编著. —北京 ：国防工业出版社，2013.6

ISBN 978-7-118-08599-0

Ⅰ. ①计… Ⅱ. ①吕… Ⅲ. ①计算复杂性－研究 Ⅳ. ①TP301.5

中国版本图书馆 CIP 数据核字(2013)第 116061 号

※

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号 邮政编码 100048)

天利华印刷装订有限公司印刷

新华书店经售

*

开本 880×1230 1/32 **印张** 6¾ **字数** 191 千字

2013 年 6 月第 1 版第 1 次印刷 **印数** 1—2000 册 **定价** 28.00 元

(本书如有印装错误，我社负责调换)

国防书店：(010)88540777 发行邮购：(010)88540776

发行传真：(010)88540755 发行业务：(010)88540717

前　言

现实世界中我们会遇到各种各样的计算问题，有的容易，有的困难。例如，数字排序问题是一个容易的问题。当我们需要按照升序或降序排列一张数字表时，可以通过小型机器很快地处理上百万的数。与之相比，要制定一所大学的课程表的时间表问题则相对难得多。如果一个学校有上千个班，那么制定一份最好的课程表则需要若干世纪。是什么原因使得一些问题计算困难而另一些问题计算容易呢？这恰是计算理论的传统核心领域之一“计算复杂性理论”的核心问题。

自20世纪60年代以来，以图灵机为基本计算模型，以时间或空间为计算能力的度量，计算复杂性理论被人们深入细致地研究，但是仍然不知道其核心问题的答案。尽管如此，该理论研究也取得一系列重要成果，特别是发现利用归约思想按照计算难度给问题分类的完备系统。目前，虽然我们不能给出某些问题计算困难的证明，但是利用该系统我们可以给出其计算困难的证据。与此同时，人们建立了许多计算模型用以分析问题的计算困难性的本质，如概率计算模型、布尔电路、交互证明系统等。对这些新模型的研究，一方面使我们对困难问题有更深入的了解，另一方面也产生了意想不到的应用。最典型的一个应用就是与密码学的结合，使得密码学有了质的飞跃。

本书首先介绍计算复杂性概述、一些计算问题和逻辑，然后详细介绍计算模型、P vs NP问题、归约和NP完备性理论等；接着针对信息安全和算法设计等专业特点，详细介绍随机化算法、(非)一致电路；最后简单介绍几个较深入的课题：交互语言类、计数复杂类、概率可验证语言类等。我们试图通过对计算复杂性理论的基础知识通俗直观地介绍，帮助读者掌握该理论的思想方法，为进一步开展计算机科学的相关领域的学习和研究奠定基础。因此，本书不仅适合作为计算机科

学各专业高年级本科生和低年级研究生(特别是信息安全专业)基础课教材,也可供有关研究人员参考。具体内容结构如下:

第0章引言对计算复杂性理论所研究的对象、基本思路以及进展给出一个简单概述。建议读者在读完本书后再返回来阅读引言将更有收获。第1章通过介绍一些典型问题及其求解算法使读者对计算复杂性理论所研究的对象和思想有一个相对直观的认识。第2章对布尔逻辑、一阶逻辑、公理化体系、存在二阶逻辑以及逻辑中的一些计算问题给予一个详细综述。对于熟悉逻辑的读者可以略过本章。第3章介绍计算复杂性理论的基本概念、基本计算模型:(多带)确定图灵机、(多带)非确定图灵机和通用图灵机及其功能,以及基于时间、空间的一些主要复杂类。第4章介绍对角化方法、停机问题等不可判性问题和递归可枚举语言的形式表达。第5章介绍一些计算复杂类以及类与类之间的关系、时间和空间分离定理、可达性方法。第6章介绍归约思想,利用归约比较问题的计算困难度,说明SAT的NP-完备性以及P-与NP-完备问题的逻辑刻画,然后介绍NP-关系、Oracle图灵机和自归约。第7章介绍若干NP-完备问题、P与NP之间一些问题研究、函数问题和coNP。第8章介绍随机化算法的一些例子、概率图灵机、概率计算及其相关语言类和性质。第9章介绍电路复杂度、单调电路和非一致多项式时间类。在第10章介绍几个较深入的课题:介绍交错图灵机与多项式谱系、交互证明系统、概率可验证证明系统和计数类#P。最后三章对于学习理论密码学的读者很有帮助。

本书是作者在中国科学院研究生院工作期间的讲义,在编写出版期间先后得到了国家自然科学基金(61272039)、先导项目(XDA06010702)和密码专项的资助,在此表示感谢。另外,由于作者水平有限,书中难免有不妥之处,敬请同行专家批评指正。

作者

2012.7.30

目　录

第0章 引 言

作为计算机科学的一个活跃领域，计算复杂性理论主要研究（数学）问题的内在难度，反映算法可行性及其所用合理资源的下界。常用的资源包括计算所需要的时间、空间（用于存储）等。

计算复杂性理论研究始于20世纪50年代末60年代初，J. Hartmanis、R. Stearns、M. O. Rabin和M. Blum等人首先开始此方面的研究。Blum不但提出了有关计算复杂性的一些公理，而且对复杂类做了进一步的归纳。随着计算复杂性理论的发展，Blum将计算复杂性理论应用于密码学，这对计算机的系统安全性和通信安全性有十分重要意义；同时，也将该理论应用于“软件工程中的程序正确性验证”。1989年5月，M. Blum与S. Kannan在第21届ACM计算理论专题研讨会上报告了成果。以上这些专家被认为是计算复杂性理论的主要奠基人。

计算复杂性理论的一个主要目标是确定任何有定义的计算任务的复杂度，另一个主要目标是理解不同计算现象之间的关系（如将一个计算任务的复杂性与另一个比较，由此导致了归约思想的产生）。我们认为前一个目标是要求对特定计算问题给予明确回答，而后一个则是关于计算问题之间的关系。有趣的是，目前计算复杂性理论在后一目标的研究成就更显著。事实上，解决“确定型”问题的失败导致了处理“相关”问题的方法的繁荣。事实上，从某种意义上说，建立问题之间的关系要比解决某类问题更有启发作用，如，NP－完备理论的产生。

目前，计算复杂性理论虽然不能完全确定问题的内在复杂度，如，有效找到给定的（可满足）公式的满足赋值或者已知可3－染色图的3－染色，但是，我们可以证明这两个看似不同的计算问题是计算等价的。类似地，复杂性理论的其他方面的问题也是很有趣的。下面先简

单概述一下组成复杂性理论的一些有趣问题。

P vs NP 问题 日常经验告诉我们解决一个问题要比判定解的正确性更困难(如,思考一个谜语或者数独游戏)。这是巧合还是客观事实?我们能想象到解决问题和验证解答正确性的难度没有显著区别的世界吗?在这样一个假想的世界中,解决问题这个词语就失去它的意义。这个困惑的否定就是"$P \neq NP$",其中 P 代表能有效解决的问题,而 NP 代表解答能被有效验证的问题。

喜欢理论的学者也许会考虑证明定理和验证证明正确性之间的关系。实事上,找到证明是解决问题的一种特殊方法,而验证证明则对应于验证解答的正确性。"$P \neq NP$"意味着"证明定理"要比"验证定理证明的正确性"困难。此时,NP 代表能在适当证明(亦称为"证书")帮助下被有效验证的断言集合,而 P 代表能被无条件验证的断言集合。

P vs NP 问题似乎超越我们现有能力,但它导致了 NP - 完备理论的发展,这是一类具有深远意义的重要问题。简单地说,该理论研究确定了一个难度等同于 NP 的计算问题的集合, P vs NP 问题的发展与每一个这样的问题相关。如果任何一个这样的问题容易求解,那么所有问题都在 P 中。于是,在"$P \neq NP$"假定下,要证明一个问题是 NP - 完备的只要提供其困难性证据。NP - 完备性是刻画计算问题困难性的一个中心工具,并且在计算机科学和相关学科中被广泛应用,又是研究不同计算问题之间关系的基础(如果不涉及它们自己的内在复杂性)。

另外,NP - 完备性也说明所讨论问题的内容非常丰富,足以"表示"NP 中其他问题,这是很重要的现象。一般来讲,复杂性理论是研究在问题陈述中解答不明显的那类问题,即,这个问题包含所有必要信息,需要仅仅通过处理这些信息获得答案。因此,复杂性理论也是关于信息处理的理论,是从一种"表示"(已知信息)到另一种"表示"(希望得到的信息)的转化。实事上,一个计算问题的解仅仅是已知信息的不同"表示",即,在该"表示"中答案是明确的。例如,一个布尔公式是否是可满足的问题的答案隐含在公式本身里,求解的目的是让它明确。因此,复杂性理论就是通过"表示"来展示明确和隐含之间的

区别。此外,它还对信息不清楚程度给予量化。

总之,计算复杂性理论为各种被前人思考过的问题提供了新视角,这不仅包括前面提及的证明和"表示"观点,也包括随机性、知识、交互、保密性等概念。

随机性 随机性的概念曾长期困惑着人们,因为以前所描述的随机性观点是存在性的。人们会问"什么是随机的"并且想知道它是否存在于所有事情中(否则,这个世界是确定的)。而计算复杂性理论对随机性的观点则是行为性的,其基础是:如果对象不能被任何有效程序区分,就认为它们是等价的。如,如果预测掷币结果是不可能的,那么掷币是随机的(即,相信世界是非确定的)。同样地,称一个字符串(或串的分布)是随机的,如果不能区分它与均匀分布(不管是否能够生成后者)。这样定义的随机性(或伪随机性)是可有效扩展的,即:在一个合理的复杂性假设下,短的伪随机字符串能够被确定地扩展为一个长的伪随机串。实事表明,随机性与困难(难解)性之间有着紧密的关系:首先,注意到伪随机性的定义涉及到困难性(即,区分伪随机对象和均匀分布对象是不可能的);其次,单向函数(容易求值而难于求逆的函数)的存在性假设蕴含存在将短的随机种子扩展成为长伪随机序列的确定程序(伪随机生成器);同时,伪随机生成器的存在性也蕴含着单向函数的存在。

知识、保密性与交互系统 计算复杂性理论提供了知识(不同于信息)的概念。它把知识视为一个困难计算的结果,因此一切能被有效实现的对象都不能被认为是知识。如,应用于公共信息的难于计算的函数值是知识,而简单计算结果则不被认为是知识。特别地,如果有人给予你这样函数值,那么他给了你知识。这就又涉及到一个新的概念,即,没有增加知识的交互(零知识交互)。这样的交互是有用的,因为它可以使对方相信事先提供的特定知识的正确性。

交互的可能动机之一是获得知识。事实表明交互在各种背景下是有用的。如,当与一个证明者交互时,验证一个断言的正确性要比得到证明容易得多。这样的交互证明的力量源于其随机性(即,验证程序是随机的)。如果验证者的问题能被提前确定,那么证明者只需在交互中提供证明的复制品就可以。另一个关于知识的概念是保密

性,知识是一部分人知道而另一部分人不知道的东西(至少仅靠自己不容易得到),因此,在某些环境中知识就是秘密。

众所周知,计算复杂性理论与密码学密切相关,而密码学则是研究"易于使用但难以破坏"的系统理论。这样的系统理论的典型性质不仅包含保密性、随机性和交互性,也包括与"易于正确使用而难以使系统偏离规则"的行为相关的复杂性差距。因此,密码更多地是基于复杂性理论假设得到一个典型变换,该变换将相对简单的计算工具(如,单向函数)转变为更复杂的密码系统(如,安全加密方案)。因此,过去20年里,密码学的最大进展是将密码学建立在计算复杂性理论的基础上,也恰是计算复杂性理论在密码学中的应用促成了密码学的飞跃,使密码学从一门艺术发展成为一门严格的科学。二者相互交织、相互促进,恰如 Goldwasser 所说:"密码学和复杂性理论的结合是天作之合。"

前面我们已经提及计算复杂性理论的两种基本类型的问题"搜索求解(或找到解)"和"做出判定(如,断言的真假判定)",也说明在某些场合二者是相关的。现在我们说明另外两种类型的问题:计算解的数目和生成随机解。这两个问题都与寻找相应问题的任意解一样困难,但事实表明,对于某些问题而言它们实质上不会更难,甚至在某些条件下,近似计算解的数目问题和生成一个近似随机解会比寻找任意解更容易。

计算解的数目和近似解 寻找近似解的复杂性研究已经得到很多关注。一类近似问题是定义在可能解集合上的目标函数,除了寻找达到最优值的解,还包括寻找"几乎最优"解。就许多例子而言,找到近似解与找到最优解一样难。在其他一些情况,合理程度的近似比解决精确搜索问题容易。近似解的困难性与概率可验证证明相关。概率可验证证明是一种允许存在快速概率验证证明的问题,每一个证明都能被有效转变为一个允许基于常数个比特探测的概率验证证明。

显然,近似是多种计算问题要求的自然放宽。另一种自然放宽是平均复杂性的研究。尽管未明确说明,前面讨论的问题都涉及对算法的最坏情况分析,但是,我们认为平均复杂性是比最坏情况复杂性更强的概念,但它的发展远不如最坏情况复杂性。在1996年,Ajtai 和

Dwork 构造了一个密码系统,展示了最坏情况与平均情况的联系。这是第一个安全性基于最坏情况的密码系统。

在复杂性理论中以随机性为中心(如在伪随机性、概率证明系统和密码系统的研究中显而易见),人们会问,在各种应用中所需要的随机性能否在现实生活中获得?一个受到很多关注的具体问题是得到"真"随机的可能性,即,我们能否用"有缺陷"随机性资源得到几乎完美的随机资源(从坏的资源中抽取好的随机性)。当然,答案依赖于有缺陷资源的模型。这项研究被证明与复杂性有关,与随机抽取器和伪随机发生器紧密关联。

目前为止,我们一直关注计算问题的时间复杂性,一直依赖于时间与效率的自然结合,但是,时间不是我们关注的唯一资源。另一个重要的资源是空间,用于计算所消耗的(暂时)存储量。空间复杂性的研究发现了许多引人注意的现象,这些现象似乎指明了时间复杂性和空间复杂性的根本区别。例如,在空间复杂性的背景下,验证(任意指定)断言成真的证明与验证同类断言的非真的证明的复杂性是相同的。

后面将从算法开始逐步详细地介绍复杂性理论的基本内容,但不涉及近似计算(求解)、随机抽取器、伪随机发生器以及密码学的内容,有兴趣的读者可以参阅参考文献中相关内容。

为了更方便地表示所用的时间、空间等量,给出如下符号定义。

约定:除了特殊的说明外,本书中所有的图都是有限的有向图。

【定义 0.1】 令 $\mathbf{N}$ 表示所有非负整数的集合,除非特殊说明外,在本书中假定所涉及的函数都是从 $\mathbf{N}$ 到 $\mathbf{N}$ 的函数,并且对于一般函数 f,定义 $f(n)=\max\{[f(n)],0\}$。

设 f 和 g 是两个从 $\mathbf{N}$ 到 $\mathbf{N}$ 的函数,记 $f(n)=O(g(n))$,如果存在正整数 c 和 n_0,使得对于所有 $n\geqslant n_0$,有 $f(n)\leqslant cg(n)$,即,f 的增长不会比 g 快。此时,亦记 $g(n)=\Omega(f(n))$,即 g 的增长不会比 f 慢。

若 $f(n)=O(g(n))$ 且 $f(n)=\Omega(g(n))$,则记 $f(n)=\Theta(g(n))$,即 f 和 g 具有相同的增长比例。

记 $f(n)=o(g(n))$,若对任意正整数 c,存在 n_0 使得对于所有 $n\geqslant n_0$,有 $f(n)<cg(n)$;类似地,记 $g(n)=\omega(f(n))$。 ■

如,若 $p(n)$ 为 d 次多项式,则 $p(n)=O(n^d)$,即 $p(n)$ 的增长比例由第一个非零项决定。又 $p(n)=\Omega(n^d)$,从而 $p(n)=\Theta(n^d)$。

若 $c>1$ 是一个整数,则 $p(n)=O(c^n)$,但是 $p(n)\neq\Omega(c^n)$,$p(n)=o(c^n)$,即任何多项式的增长严格慢于指数函数。

类似地,$\log_2 n=O(n)$,$\log_2 n=o(n)$,$\log_2^k n=O(n)$,$\log_2^k n=o(n)$,其中,k 为任意正数。

习 题

0.1 设 $f(n)$ 与 $g(n)$ 是下列函数中任意两个:

(1) n^2; (2) n^3; (3) $n^2\log_2 n$; (4) 2^n; (5) n^n; (6) $n^{\log_2 n}$;

(7) 2^{2n}; (8) 2^{2n+1}; (9) 当 n 为奇数时为 n,否则为 2^n。

判定下列关系是否成立:

(1) $f(n)=O(g(n))$;

(2) $f(n)=\Omega(g(n))$;

(3) $f(n)=\Theta(g(n))$;

(4) $f(n)=o(g(n))$;

(5) $f(n)=\omega(g(n))$。

0.2 一个运行时间为 $O(n\log_2 n)$ 的分类算法恰好用 1μs 可以分类 1000 条数据。假定分类 n 条数据的时间 $T(n)$ 与 $n\log_2 n$ 成正比,即 $T(n)=c\cdot n\log_2 n$。试给出 $T(n)$ 的公式表达,并估计分类 1000000 条数据所需要的时间。

0.3 一个运行时间为 $O(f(n))$ 的算法处理数据的时间为 $T(n)=cf(n)$,这里 $f(n)$ 是一个已知的关于 n 的函数。若该算法处理 1000 条数据花费 10s,那么,当 $f(n)=n$ 和 $f(n)=n^3$ 时花费多少时间可以处理 100000 条数据?

0.4 证明 $T(n)=a_0+a_1 n+a_2 n^2+a_3 n^3$ 为 $O(n^3)$。

第1章　一些计算问题

计算复杂性理论与算法设计和分析是计算机科学的两个领域，都是极力估计算法复杂度，衡量在现实资源要求下能做什么和不能做什么，但它们是从两个相反的方向研究算法问题的。算法是一个详细的逐步解决问题的方法。自然地，证明一个问题有有效可解的算法是一个更受欢迎的结果，因为一个有效算法不仅可以直接求解问题，也是该问题有效可解的一个证明。然而，每个问题都有确定的算法复杂度。计算复杂性理论就是要弄清楚求解问题所需要的最小资源，证明求解困难问题不能有更节省资源的办法，这是问题本身的固有难度。

当设计一个算法时，我们只需要形成算法并分析，这将得到求解问题所需要的极小资源上界。而计算复杂性理论则是提供下界，即，每个求解该问题的算法都必需的资源下界。对于上界的证明，设计和分析一个有效算法是足够的，而每个下界则是要研究求解一个特定问题的所有算法。对于这些算法组成的集合，我们能够掌握的性质就是它们能够求解该问题。因此，为了证明一个特定问题的求解要求一定的极小资源，我们必须要考虑到关于该问题的所有算法，这恰是计算复杂性的难点所在。

当然，二者是相互联系、相辅相成的。当证明一个问题没有有效解时，我们会发现问题的一些本质。这些本质通常能够揭示问题的困难性所在，有助于我们得到问题困难性证明的突破口。此外，当我们发现所研究的问题不是有效可解的时候，首先我们得到的好处是“放弃寻找该问题的有效算法”，避免浪费精力于一个不可达到的目标（如：理智的人不可能去设计永动机）；其次，通过加强限制条件等办法，我们可以找到有效可解的问题（所谓加强限制条件是指问题的特殊化，或要求更弱的解形式）。因此，“证明一个问题不是有效可解

的”与“设计有效算法”一样都有意义，这便于人们设计更高效的算法和评价某个算法的优劣。

什么是问题呢？一般性的问题的定义太宽泛，难以形式化。为了能够研究更多可处理的问题，我们仅考虑需要解决且有价值的计算问题。它们可以通过计算机处理并且其正确解的集合是确定的。但是，就计算复杂性理论而言，并不是所有计算机可处理的问题都是计算问题。计算问题具体定义如下：

【定义 1.1】 一个计算问题如下组成：

(1) 对可允许的输入集合的描述：在计算机的有限字母表上，每个输入可以表示为一个字符串。

(2) 对映射每个输入到正确输出(回答或结果)的函数的描述：每个函数亦是一个有限字母表上的有限序列。 ■

由定义可见，一个问题不是由单个的询问构成，而是由一族询问构成。对于一个给定的问题，这些询问都是相关的，而且有一个简单的“填空”式结构。每个输入就是一个实例。对于不同的询问，每个不同的输入被填入空格，如下列的形式：

实例：一个正整数 n。

问题：n 是否为素数？

对于问题的回答，则需要寻找算法。尽管精确的形式化不重要，但是通常问题与算法都需要分别作为语言和图灵机被形式化并被分析。根据实际可解性的直观意义，多项式时间的可计算性是计算问题重要的理想性质。另外，亦存在一些问题，不管效率如何，它们没有求解的算法。

根据对问题的回答的要求不同，问题可分为以下几类：

(1) 搜索性问题。对于给定的问题，找到一个可能的解。如，对于图的可达性问题，则是给定两点 u、v，找到两点间的一条通路。

(2) 最优化问题。对于给定的问题，找出最佳的解。如上例中，找出从 u 到 v 的最短路径。若仅要求一个最优解的值，则称问题为算值问题。

(3) 验证问题。对于给定的问题及一个解，验证该解是否为该问题的解。如上例中，给定一条通路，验证是否是最短的。

(4) 判定性问题。给定一个问题,判定是否有解。如,上述问题中,判定是否有从点 u 到点 v 的通路。

显然,搜索性问题比判定性问题更难,而判定性问题比验证性问题更难。即,

$$\text{搜索性问题} \xrightarrow{\text{求解}} \text{判定性问题} \xrightarrow{\text{求解}} \text{验证性问题}$$

对于最优化问题,多数情况下可以利用判定性问题求解。

下面,我们考虑一些问题的例子。

【例 1.1】 图的可达性问题(GRAGH REACHABILITY)。

一个图 $G=(V,E)$ 由一个顶点的有限集 V 和边的有限集 E 组成,其中 $V=\{1,2,\cdots,n\}$,$E=\{(i,j):i,j\in V\}$。

关于图,有许多计算问题,最基本即是图的可达性(GRAGH REACHABILITY):

给定一个图 G 和两个顶点 $1,n\in V$,是否有一条从 1 到 n 的通路(Path)? 称此问题为可达性问题,记为 REACHABILITY。

如图 1-1 所示,$G=(V,E)$,$V=\{1,2,3,4,5\}$,$E=\{(1,4),(4,3),(2,1),(2,3),(5,4),(3,5)\}$,问:

(1) 是否有从 1 到 5 的通路?

(2) 找出一条从 1 到 5 的通路?

(3) 验证:(1,2,3,5)是否是一条从 1 到 5 的通路?

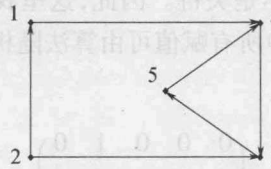

图 1-1 图 G

显然,(1,4,3,5)即为一条从 1 到 5 的通路,但(1,2,3,5)不是一条通路。这 3 个问题分别是判定性、搜索性和验证性问题。大家可以考虑:寻找从 1 到 5 的最短通路。

如大多数有趣的问题,图的可达性问题满足:

(1) 无限多个可能的实例。