形式化框架下置换和查找类算法的组装生成

石海鹤 著

科学出版社

江西师[redacted]力成果

形式化框架下置换和查找类算法的组装生成

石海鹤 著

科学出版社
北京

内 容 简 介

算法的可靠性和开发效率对于软件的可信性及应用发展具有重要意义。算法自动化是提高算法开发效率、保证算法可靠性的重要途径。

置换和查找是计算机学科中的两类特殊问题，可应用算法设计策略的灵活性使其算法更具多样性，算法生成的自动化程度难以提高。本书结合软件形式化方法 PAR，将生成式程序设计思想引入到算法开发中来，组装生成了典型的置换和查找类算法，以及若干未见于现有文献的算法，构建了具备相应生成能力的系统，显著提高了两类算法的开发效率和可靠性，可望从方法学和实践上为特定领域高可靠算法的开发提供新思路。

本书包含了我们在置换和查找类算法自动生成方面的创新性研究成果，对软件开发人员和研究人员具有参考价值。

图书在版编目(CIP)数据

形式化框架下置换和查找类算法的组装生成/石海鹤著. —北京：科学出版社，2017.5

ISBN 978-7-03-052213-9

Ⅰ.①形… Ⅱ.①石… Ⅲ.①算法设计 Ⅳ.①TP301.6

中国版本图书馆 CIP 数据核字（2017）第 054909 号

责任编辑：胡庆家 赵彦超 / 责任校对：彭 涛

责任印制：张 伟 / 封面设计：耕者工作室

科学出版社 出版

北京东黄城根北街 16 号

邮政编码：100717

http://www.sciencep.com

北京京华虎彩印刷有限公司 印刷

科学出版社发行 各地新华书店经销

*

2017 年 5 月第 一 版 开本：720×1000 B5

2017 年 5 月第一次印刷 印张：9 3/4

字数：197 000

定价：68.00 元

（如有印装质量问题，我社负责调换）

前　言

随着社会对信息技术依赖性的日益增长，处于信息技术核心的计算机软件的可信性被提到一个新的高度，国内外都很重视并已开展大量的研究工作。目前，对软件可信性的需求已从安全性至关重要的领域，如国防、航空航天、医疗器械等领域扩展到能源、通信、财经、制造业等关键领域。

算法程序是指用可执行程序设计语言或抽象程序设计语言描述的算法，作为软件的核心，其可靠性和开发效率对于软件的可信性及应用发展具有重要意义。算法程序自动化研究从形式化功能规约到可执行程序这一过程的自动化，是提高算法开发效率、保证算法可靠性的重要途径之一。然而，算法设计是软件开发中知识高度密集的创造性劳动，其自动化仍是软件自动化领域的研究难点。

置换和查找是计算机学科中的两类特殊问题，可应用的算法设计策略很灵活，产生了丰富的置换和查找类算法程序，从同样的问题规约出发，不同的问题分划和规约变换所导致结果算法的形式和性能差异尤为明显，这使得领域算法程序的共性难以刻画，算法生成的自动化程度和效果不理想。

本书结合著者所在学术团队已取得的软件形式化方法 PAR 及其支撑平台，选取置换和查找这两类特殊问题作为研究对象，开展了形式化方法制导下的算法程序自动化研究，构建了置换和查找类算法领域高可靠构件库，组装生成了归并排序、快速排序、堆排序、Shell 排序、散列查找等 30 余个典型的置换和查找类已知算法，以及增量选择排序等若

干未见于现有文献的算法，并开发了具备相应生成能力的系统，从而显著提高了两类算法的开发效率和可靠性，为生成问题的求解算法类提供了一种可供借鉴的方法和途径。

本书旨在介绍我们在置换和查找类算法自动生成方面的创新性研究成果。共分 8 章，第 1 章为引言，第 2 章概述算法程序自动化的国内外研究现状，第 3 章介绍了形式化框架 PAR，第 4 章阐述 PAR 框架下的算法形式化开发法则和策略，第 5 章建立置换和查找类算法领域的生成模型，第 6 章和第 7 章分别将前述结果应用于具体的置换和查找算法的生成，第 8 章是总结与讨论。

本书的工作得到国家重大基础研究（“973”计划）前期研究专项（2003CCA02800）和国家自然科学基金项目（60273092，61363013，61662035）的资助。薛锦云教授指导了本书中的研究工作，王捍贫教授、李舟军教授、张健研究员、沈一栋研究员、陈海明研究员、张文辉研究员、詹乃军研究员等专家提出了许多宝贵的意见和建议。此外，书中引用了一些专家学者的论著和研究成果。作者在此一并表示真诚的感谢。

限于水平，书中错误和不妥之处难免，敬请读者批评指正。

石海鹤

2016 年 10 月 1 日

江西南昌

目　录

第1章 引 言

1.1 研究背景

随着社会对信息技术依赖性的日益增长，处于信息技术核心的计算机软件的可信性被提到一个新的高度，国内外都很重视并已开展大量的研究工作。Hoare 组织的关于软件验证的国际性项目被认为是计算机科学界 21 世纪的一个重大挑战性问题[1-2]；美国 DARPA，NASA，NSA，NSF，NCO/ITRD 等机构都积极参与可信项目的研究开发并制定一系列官方报告[3-4]，美国政府制定的 2006—2015 年国家软件发展战略——“下一代软件工程”中，也将提高软件可信性放在四大战略任务的首位。国内学者纷纷参与到这一国际课题的讨论中[5-8]。目前，对软件可信性的需求已从安全性至关重要的领域，如国防、航空航天、医疗器械等领域扩展到能源、通信、财经、制造业等关键领域。

算法程序是指用可执行程序设计语言或抽象程序设计语言描述的算法，作为软件的核心，其可靠性和开发效率对于软件的可信性及应用发展具有重要意义。算法程序自动化研究从形式化功能规约到可执行程序这一过程的自动化，是提高算法开发效率、保证算法可靠性的重要途径[5-6]。图灵奖获得者 Jim Gray[9] 将“自动程序设计”（Automatic Programming）列为 21 世纪信息技术领域的 12 个重要研究目标之一，*Newsweek International*[10]，*Computer World*[11]，*IEEE Software*[12] 等多家新闻期刊也对该领域成果进行了报道。

算法程序自动化可分为算法设计和程序实现两个阶段。程序实现阶段的工作是完成从求解算法到计算机上可执行语言程序的自动转换，这一阶段的工作近年来进展较为顺利，技术已相对成熟。而算法设计阶段需要从刻画“做什么”的问题功能规约自动地生成反映“如何做”的求解算法，由于算法设计（特别是精妙算法的设计）是软件开发中知识高度密集的创造性劳动，目前人工智能技术难以处理这类创造性劳动，现有的各种算法程序的优化技术也难以产生精妙算法，使得算法设计自动化已成为软件自动化的难点和关键，在软件自动化领域尚未很好解决。

1.2 研究目标和内容

算法形式化方法严格、精确地定义了用户需求，形式化规约的求精过程具有可推导、易证明的特性，在保证算法的正确性、展示算法设计的过程、揭示算法蕴含的思想等方面具有重要的意义，是寻求算法设计的本质特征和一般规律的有效途径，有利于实现算法程序自动化。

PAR[13-15]是在对现存算法程序设计方法局限性和大量算法程序特性深入研究的基础上提出的一种实用的形式化方法，是多项国家级项目连续资助下形成的原创性成果。该方法将特殊问题中使用的分划和递推技术相结合，涵盖了如动态规划法、贪心法、分治法、穷举法等多种已知的算法设计技术，恰当地区分了开发过程中的创造性劳动和非创造性劳动，为算法程序自动化提供了有力支持。

由于通用的算法自动生成是很难的问题，人们往往研究某些特殊领域的算法自动生成。置换和查找是计算机学科中的两类特殊问题，前者是指将序列中 n 个元素的位置按某种要求进行重排，排序是其中重要的一类问题，即产生输入序列的有序置换；后者也称检索，是指

确定某个给定元素在输入序列中的位置。对于这两类问题，可应用的算法设计策略很灵活，产生了丰富的置换和查找类算法程序，从同样的问题规约出发，不同的问题分划和规约变换所导致结果算法的形式和性能差异尤为明显，这使得领域算法程序的共性难以刻画，算法生成的自动化程度和效果不理想。从一个更广泛的方面着眼，它们是研究一般情况下如何着手解决计算机程序设计问题的有价值的实例研究[16]。

本书以软件形式化方法 PAR 及其支撑平台为基础，将生成式程序设计思想、泛型程序设计技术以及抽象数据类型机制综合应用于算法程序生成领域，对置换类和查找类算法程序的自动生成进行了研究，通过寻找算法程序设计的规律，提出新的科学符号、方法和技术来逐步减少算法程序开发中的创造性劳动，从而逐步提高从问题规约生成算法程序的自动化程度。本书的主要研究内容和贡献概括如下：

（1）概述了算法程序自动化的研究现状

从算法程序设计方法学、语言、算法设计能力、支撑工具及其应用于置换和查找算法生成等几个方面阐述了国内外算法程序自动化的主要研究成果，讨论了当前研究的不足以及今后的研究方向。

（2）提出了使用 PAR 推导一类问题求解算法的问题分划法则及规约变换策略

以结果算法的效率作为决定问题分划和规约变换的标准，研究了用 PAR 开发算法时问题分划、递推关系构造等若干关键技术，寻找高效算法程序开发的特征及规律并尽可能提炼归纳成了切实可行的法则和策略，这包括问题分划法则和规约变换策略，有效降低了形式化开发算法的难度与复杂度，为算法程序自动化提供了理论基础。

（3）借助领域建模的概念和方法对置换和查找类算法进行抽象，建立了领域特定语言

收集和分析了置换和查找类算法领域的相关信息，抽象出这两个

领域的算法基本操作并使用 PAR 提供的自定义抽象数据类型机制进行描述，提出并实现了排序表（SortingList）泛型 ADT，堆（Heap）泛型 ADT，查找表（SearchingList）泛型 ADT 以及散列表（Hash）泛型 ADT，使用 Hoare 公理方法刻画了部分 ADT 的形式化规约，建立了算法领域特定语言。

（4）设计并实现了置换和查找类算法领域的五个泛型算法构件

刻画了置换问题的代数性质，通过分析置换类算法和查找类算法的共性和可变性，融合 ADT 机制，以及泛型和生成式程序设计等技术，对置换和查找类算法进行科学分类，设计并形式化开发了这两个领域的五个泛型算法构件 DBPSort，UBPSort，HSort，UnorderSearch 和 OrderSearch，使用前、后置断言作为泛型约束机制刻画了其中操作参数的性质和行为，以保证参数替换的合理性和安全性，这为置换和查找算法的生成提供了算法生成形式化模型。

（5）扩充 Apla-Java 程序生成系统，并自动生成了 30 余个具体的典型置换、查找类已知算法以及未见于现有文献的算法

用领域内的类型构件实例化算法构件，以参数替换的方式自动生成了归并排序、快速排序、堆排序、Shell 排序、散列查找等 30 余个典型的置换和查找类已知算法，以及增量选择排序等若干未见于现有文献的算法，并在 Apla-Java 程序生成系统中予以实现，显著提高了算法程序的开发效率。研究结果能够发现新的算法而不只是解释或证明一些已知算法。

本书利用形式化方法 PAR 在揭露算法本质及算法设计规律、形式化开发复杂算法等方面的优势，就置换和查找类问题求解算法的自动生成进行了探索，拓展了 PAR 方法和 PAR 平台，把算法设计从单纯依靠灵感和技巧的活动转变为规范化的程序生成活动，显著提高了算法程序的开发效率和可靠性，为有效算法的设计、可靠部件库的设计及重用、自动程序设计提供了支持。

1.3 本书组织结构

全书共分8章。

第1章引言。

第2章综述了国内外算法程序自动化的主要研究成果及其应用于置换和查找类算法生成方面的情况。

第3章概述了PAR方法，主要包括循环不变式的新定义及新开发策略、Radl语言、Apla语言以及算法程序开发方法。

第4章提出了基于PAR的算法形式化开发法则和策略，这包括问题分划法则和规约变换策略，并通过一个实例展示了它们的应用过程和效果。

第5章刻画了置换问题的代数性质，对置换和查找类算法领域进行了分析，设计并实现了泛型类型构件和算法构件，建立了这两个领域的特定语言和算法生成模型。

第6章应用前述研究结果为荷兰国旗问题及排序问题开发和自动生成了系列具体的求解算法程序。

第7章为查找问题自动生成了系列已知的典型查找算法程序以及若干无名算法程序。

第8章将本书工作和相关工作进行了比较，在总结全书内容的基础上，讨论了进一步的工作。

第 2 章　算法程序自动化方法概述

算法程序形式化是算法程序自动化的基础和关键，因此在实际研究算法程序自动化的过程中，总是会结合着对算法程序形式化的研究。算法程序形式化和自动化在理论和实践上都处于方兴未艾的发展阶段，国内外很多学者针对不同的问题，采用不同的技术在算法程序形式化和自动化方面开展了许多研究工作。虽然有些工作较少涉及算法设计自动化，但是其研究方法、研究途径和研究成果均构成了算法设计自动化研究的基础。置换问题不但本身十分有趣，而且在理论和实践上都具有十分重要的意义，专著［17］专门以 7 个不同的排序算法为案例，分析比较了各种形式化开发方法。从一个更广泛的方面着眼，置换是研究一般情况下如何着手解决计算机程序设计问题的有价值的实例研究，这一直吸引着许多学者从各个角度不断探索并且取得了大量极富创造性的成果。同样地，它也是形式化和自动化领域人们研究最多的问题。

接下来我们将从算法程序设计方法学、语言、算法设计能力、支撑工具及其应用于置换和查找算法生成等几个方面，对该领域的主要研究成果进行阐述。

2.1　基于演绎推理的方法

该方法将算法程序设计看作演绎过程，从给定的问题规约出发，使用各种规则进行演绎和搜索，借助演绎推理综合出程序。问题规约

通常使用基于一阶谓词逻辑的前后置断言来表示。

使用定理证明技术方面，较典型的是 Z. Manna 和 R. Waldinger 的 Tableau 方法[18-19]，它将数学定理的构造性证明与程序开发相联系，将问题规约变成待证的定理，把开发步骤解释为证明步骤，证明过程伴随程序的构造，证明的成功结束表示程序构造完毕，即可从证明中抽取相应的类 LISP 函数式程序。系统中使用一张由断言、目标和输出表达式三部分组成的序表（Sequent），其中断言和目标用一阶逻辑表达而输出表达式用类 LISP 语言表达，该表的含义是，如果每一断言的所有实例为真，那么至少有一目标的实例为真且相应的输出表达式满足给定的规约。系统运行就是使用各种规则不断的改变这张表而不改变其含义，直到完成证明为止，相应的输出表达式即为所导出的程序。为了克服纯定理证明技术所带来的局限性，该方法将合一、归结、数学归纳法和程序变换技术相结合，统一在定理证明框架之下。Martin-Löf 构造性类型理论既是一种逻辑，又是一种程序设计语言，在其中程序规约可作为类型，而类型又是公式，公式的证明对应于程序的开发，Nuprl 系统[20-21]、Coq 系统[22]就是基于 Martin-Löf 类型理论的程序开发系统，交互的以自顶向下的方式构造证明，证明结束后便可从中抽取相应的函数式程序。这类方法具有健全的数学基础，综合一个满足给定规约的程序问题形式上等价于找到此规约满足的一个构造性证明。演绎本质上是搜索推理路径的问题，由于搜索方法的低效，尚未开发出规模较大的复杂程序。另外，它所寻求的构造性证明，往往对应的是低效算法。

Buchberger 提出一种“惰思考”（Lazy Thinking）的算法综合方法[23]，将原始规约看成需证明的定理，证明和算法发现交替进行，进行第一回合的证明；当证明进行不下去时，再添加相应的知识（也即发明算法的一部分）到知识库中，再试着进行第二回合的证明与发现，直到碰到某种情形无法继续，再添加知识到知识库，如此下去，

直到最后成功地完成证明，得到完整的知识库。这里的“惰思考”表现为证明时遇到无法进行下去的情形，再来考虑该添加什么样的已知条件才能使得证明可以继续，如此循环，直到证明完成。

Floyd[24]，Dijkstra[25-26]，Gries[27]等提出并发展了一种基于演绎推理的程序开发方法，将程序和其正确性证明“手拉手”的开发出来，且程序的每个片段的证明总是先于该片段代码而得到。文献［28］-［30］基于演绎推理技术进一步开展了相关的研究，通过加入更多的演算技术来展示如何从规约得到程序。IBM 的 Yakhnis[31]等研究人员研究了程序推导技术和正确性证明的重用问题，创建了一个泛型算法库。用户求解问题时，从库中选一规约与待求解的问题规约匹对，建立标识符映射，通过替换泛型算法的相应标识符来得到求解问题的算法，但是他们没有提供对中间过程而仅提供对最后结果的重用，也没有提供任何支持工具。这类方法的重点放在开发程序而不是设计算法上，它将程序正确性证明的理论融合到开发过程中，边开发边保证程序的正确性，自动化程度低，且使用该方法构造循环程序，必须先开发出循环不变式，这是公认的难题。另外，基于循环不变式的传统定义及开发策略，这些工作开发的循环不变式可能存在着不足以反映程序中每个循环变量的变化规律及不能刻画循环程序本质特征等问题。

Cocktail[32]是在其基础上发展的交互式工具，目的是支持 Dijkstra/Hoare 式的程序演算方法，对从规约推导出 GCL（Dijkstra 的卫式命令语言 Guarded Command Language）程序提供语义支持。Cocktail 用 Java 编写，系统中混合了 PTSs（Pure Type Systems）、ATP（Automated Theorem Prover）及 Hoare 逻辑等理论作为逻辑基础，保持对所有证明约束（Proof Obligations）的跟踪，支持程序员构建证明以及检查证明和程序的正确性。该工具实际上是分担程序员必须手工对证明约束进行管理的事务，目前主要用于教育环境。

在开发排序算法程序方面，Backhouse[33]使用一阶逻辑谓词精确