计算机软件工程丛书

软 件 自 动 化

徐家福　陈道蓄　吕　建　王志坚　著

清 华 大 学 出 版 社
广西科学技术出版社

(京)新登字 158 号
(桂)新登字 06 号

内 容 简 介

计算机软件领域存在功能不强、质量欠佳、生产率低三问题。实现软件自动化对于提高软件生产率至为重要。本书旨在讨论软件自动化的基本概念、基本方法与技术。综述了国内外的研究现状;分别论述了软件自动化的主要问题与实现途径(演绎综合,程序转换,归纳综合与过程实现);对软件规格说明与机器学习也有系统介绍,最后着重介绍了作者研制的四个实验性系统,便于读者理论联系实际。

本书是一本汇集南京大学计算机软件研究所关于软件自动化研究成果的学术专著,对软件开发和研究人员具有指导意义。

版权所有,翻印必究。
本书封面贴有清华大学出版社激光防伪标志,无标志者不得销售。

图书在版编目(CIP)数据

软件自动化/徐家福等著. —北京:清华大学出版社,1994
(计算机软件工程丛书)
ISBN 7-302-01540-6

Ⅰ.软… Ⅱ.徐… Ⅲ.①软件工程-自动化-概论②程序系统-自动化-概论Ⅳ.TP311.5

中国版本图书馆 CIP 数据核字(94)第 05077 号

出版者:清华大学出版社(北京清华大学校内,邮编 100084)
广西科学技术出版社(南宁市河堤路 14 号,邮编 530021)
印刷者:清华大学印刷厂
发行者:新华书店总店北京科技发行所
开 本:787×1092 1/16 印张:13.5 字数:316 千字
版 次:1994 年 9 月第 1 版 1994 年 9 月第 1 次印刷
书 号:ISBN 7-302-01540-6/TP·631
印 数:0001—2000
定 价:16.00 元

清华大学出版社　广西科学技术出版社
计算机学术著作出版基金

评 审 委 员 会

主 任 委 员　张效祥

副主任委员　周远清　汪成为

委　　员　王鼎兴　杨芙清　李三立　施伯乐　徐家福
　　夏培肃　董韫美　张兴强　徐培忠

出 版 说 明

近年来,随着微电子和计算机技术渗透到各个技术领域,人类正在步入一个技术迅猛发展的新时期。这个新时期的主要标志是计算机和信息处理的广泛应用。计算机在改造传统产业,实现管理自动化,促进新兴产业的发展等方面都起着重要作用,它在现代化建设中的战略地位愈来愈明显。计算机科学与其它学科的交叉又产生了许多新学科,推动着科学技术向更广阔的领域发展,正在对人类社会产生深远的影响。

科学技术是第一生产力。计算机科学技术是我国高科技领域的一个重要方面。为了推动我国计算机科学及产业的发展,促进学术交流,使科研成果尽快转化为生产力,清华大学出版社与广西科学技术出版社联合设立了"计算机学术著作基金",旨在支持和鼓励科技人员,撰写高水平的学术著作,以反映和推广我国在这一领域的最新成果。

计算机学术著作出版基金资助出版的著作范围包括:有重要理论价值或重要应用价值的学术专著;计算机学科前沿探索的论著;推动计算机技术及产业发展的专著;与计算机有关的交叉学科的论著;有较大应用价值的工具书;世界名著的优秀翻译作品。凡经作者本人申请,计算机学术著作出版基金评审委员会评审通过的著作,将由该基金资助出版,出版社将努力做好出版工作。

基金还支持两社列选的国家高科技重点图书和国家教委重点图书规划中计算机学科领域的学术著作的出版。为了做好选题工作,出版社特邀请"中国计算机学会"、"中国中文信息学会"帮助做好组织有关学术著作丛书的列选工作。

热诚希望得到广大计算机界同仁的支持和帮助。

清华大学出版社
广西科学技术出版社
计算机学术著作出版基金办公室

1992 年 4 月

序　　言

计算机是当代发展最为迅猛的科学技术，其应用几乎已深入到人类社会活动和生活的一切领域，大大提高了社会生产力，引起了经济结构、社会结构和生活方式的深刻变化和变革，是最为活跃的生产力之一。计算机本身在国际范围内已成为年产值达 2500 亿美元的巨大产业，国际竞争异常剧烈，预计到本世纪末将发展为世界第一大产业。计算机科学技术具有极大的综合性质，与众多科学技术相交叉而反过来又渗入更多的科学技术，促进它们的发展。计算机科学技术内容十分丰富，学科分支生长尤为迅速，日新月异，层出不穷。因此在我国计算机科学技术尚比较落后的情况下，加强计算机科学技术的传播实为当务之急。

中国计算机学会一直把出版图书刊物作为学术活动的重要内容之一。我国计算机专家学者通过科学实践，做出了大量成果，积累了丰富经验与学识。他们有撰写著作的很大积极性，但相当时期以来计算机学术著作由于印数不多，出版往往遇到不少困难，专业性越强越有深度的著作，出版难度越大。最近清华大学出版社与广西科学技术出版社为促进我国计算机科学技术及产业的发展，推动计算机科技著作的出版工作，特设立“计算机学术著作出版基金”，以支持我国计算机科技工作者撰写高水平的学术著作，并将资助出版的著作列为中国计算机学会的学术著作丛书。我们十分重视这件事，并已把它列为学会本届理事会的工作要点之一。我们希望这一丛书能对传播学术成果、交流学术思想、促进科学技术转化为生产力起到良好作用，能对我国计算机科技发展具有有益的导向意义，也希望我国广大学会会员和计算机科技工作者，包括海外工作和学习的神州学人们能积极投稿，出好这一丛书。

中国计算机学会

1992 年 4 月 20 日

序

软件领域存在功能不强、质量欠佳、生产率低三问题。症结一为本质，二为理论基础，三为开发手段。海内外学者多年研究探索，工作虽多，鲜有突破。实现软件自动化对于提高软件生产率至为重要，现有演绎综合，程序变换，归纳综合，以及过程实现诸途径，特点各异，互有短长。南京大学计算机软件研究所研究软件自动化十余年，已研制成六个实验性系统，并开设研究生课程多次。今将讲稿修改整理，汇成是书，以飨读者。

本书旨在讨论软件自动化的基本概念，基本方法与技术。共分八章，第一章为引言，第二章至第七章，按软件自动化的主要问题与途径，分别讨论，第八章着重介绍作者研制的实验性系统，便于读者理论联系实际。

陈道蓄、吕建、王志坚三位教授师从于吾，历有年所。根基扎实，多所创见。吾等不揣浅陋，著成此书，谬误之处，尚祈指正。

南京大学计算机软件研究所伊波、翟成祥、费宗铭诸君与笔者多次讨论，获益良多。蔡红小姐精心誊录与校勘。清华大学出版社大力支持。作者在此一并深为致谢。

徐家福

1993 年 10 月 1 日

于南京大学

目　　录

第一章　引言 …… (1)
1.1　研究动因 …… (1)
1.1.1　软件问题 …… (1)
1.1.2　解决途径 …… (2)
1.2　基本含义 …… (3)
1.2.1　广义理解 …… (3)
1.2.2　狭义理解 …… (3)
1.2.3　不同层次的理解 …… (3)
1.3　主要内容 …… (4)
1.3.1　软件开发 …… (4)
1.3.2　规格说明 …… (4)
1.3.3　自动生成 …… (5)
1.3.4　自动验证 …… (5)
1.4　实现途径 …… (5)
1.4.1　演绎综合 …… (5)
1.4.2　程序转换 …… (6)
1.4.3　归纳综合 …… (8)
1.4.4　过程实现 …… (8)
1.5　现状综述 …… (8)
1.5.1　成就 …… (9)
1.5.2　问题 …… (9)
1.5.3　途径 …… (10)
1.6　本书的目的和体式 …… (10)
1.6.1　目的 …… (10)
1.6.2　体式 …… (10)
第二章　软件规格说明 …… (11)
2.1　基础知识 …… (11)
2.1.1　层次级别与描述手段 …… (11)
2.1.2　过程抽象与数据抽象 …… (12)
2.2　软件规格说明方法 …… (13)
2.2.1　前后断言方法 …… (13)
2.2.2　HOS 方法 …… (16)
2.2.3　逻辑性质与可解性 …… (21)
2.2.4　代数方法 …… (22)
2.2.5　抽象模型方法 …… (31)
2.3　软件规格说明语言 …… (33)

2.3.1 Z 规格说明语言 …… (33)
2.3.2 Larch 语言 …… (41)
2.3.3 广谱语言 CIP-L …… (48)
2.3.4 规格说明语言 FGSPEC …… (58)
第三章 演绎综合途径 …… (66)
3.1 基本理论 …… (66)
3.1.1 逻辑基础 …… (66)
3.1.2 定理证明 …… (68)
3.1.3 演绎综合 …… (70)
3.1.4 数学归纳 …… (72)
3.1.5 程序综合技术 …… (74)
3.2 基本方法 …… (75)
3.2.1 生成构架 …… (75)
3.2.2 演绎规则 …… (78)
3.2.3 归纳构造 …… (86)
3.2.4 实例研究 …… (90)
3.3 方法评述 …… (93)
第四章 程序转换 …… (95)
4.1 纵向转换 …… (95)
4.1.1 转换模型 …… (96)
4.1.2 正确性构架 …… (96)
4.1.3 前件推导机制 …… (97)
4.1.4 知识表示机制 …… (98)
4.1.5 算法设计方法的选择机制 …… (104)
4.1.6 例 …… (105)
4.2 横向转换 …… (107)
4.2.1 转换对象 …… (107)
4.2.2 转换规则 …… (108)
4.2.3 Unfold/fold 转换方法 …… (108)
4.2.4 若干技术 …… (111)
4.2.5 Unfold/fold 方法的自动化 …… (112)
4.3 抽象数据类型的实现构架 …… (114)
4.3.1 代数规格说明的逐层转换 …… (114)
4.3.2 数据精化方法 …… (118)
4.4 程序转换系统 CIP-S …… (120)
4.4.1 理论基础 …… (121)
4.4.2 基本转换规则 …… (123)
4.4.3 转换策略 …… (126)
4.5 综合评述 …… (132)
第五章 过程实现途径 …… (134)
5.1 扩展编译 …… (134)

5.2 通用语言和系统 …… (135)
5.3 专用语言和系统 …… (136)
5.4 领域知识与自动化 …… (137)
第六章 归纳综合途径 …… (139)
6.1 归纳推理 …… (139)
6.1.1 基本概念 …… (139)
6.1.2 研究范围 …… (140)
6.1.3 评价标准 …… (141)
6.1.4 通用方法 …… (145)
6.2 方法概述 …… (146)
6.2.1 实例法 …… (146)
6.2.2 轨迹法 …… (148)
6.2.3 实用的归纳推理方法 …… (150)
6.3 程序综合 …… (151)
6.3.1 子句逻辑 …… (152)
6.3.2 综合算法 …… (153)
6.3.3 假设空间 …… (155)
6.3.4 搜索策略 …… (158)
6.4 方法评述 …… (161)
6.4.1 背景知识 …… (162)
6.4.2 层次构作 …… (163)
6.4.3 假设的证实 …… (163)
6.4.4 结语 …… (164)
第七章 软件自动化与机器学习 …… (165)
7.1 机器学习 …… (165)
7.1.1 学习的含义 …… (165)
7.1.2 学习途径 …… (167)
7.1.3 基于解释的学习 …… (168)
7.1.4 归纳学习 …… (169)
7.2 学习归约求解问题的分解方法 …… (170)
7.2.1 问题归约 …… (170)
7.2.2 算法构架学习 …… (172)
第八章 软件自动化实验性系统 …… (178)
8.1 软件自动化系统 NDAUTO …… (178)
8.1.1 GSPEC 语言 …… (178)
8.1.2 系统功能与实现途径 …… (180)
8.1.3 系统组成 …… (180)
8.2 算法设计自动化系统 NDADAS …… (183)
8.2.1 系统功能与特点 …… (183)
8.2.2 FGSPEC 语言 …… (183)
8.2.3 系统结构 …… (184)

8.3 归纳程序综合系统 NDIPS …… (189)
8.3.1 系统结构 …… (189)
8.3.2 程序综合 …… (190)
8.4 具有自学习能力的软件自动化系统 NDSAIL …… (192)
8.4.1 系统的功能和结构 …… (192)
8.4.2 算法构架学习和作用机制 …… (193)
8.4.3 基本算法学习 …… (196)
8.4.4 优化方法学习 …… (196)
参考文献 …… (198)
索引 …… (200)

第一章　引　　言

本章概述软件自动化的研究动因, 基本含义, 主要内容, 实现途径, 现状综述, 以及本书的目的和体式。俾读者能窥其梗概。

1.1　研究动因

1.1.1　软件问题

自 1956 年 IBM704 机上的 FORTRAN 出现后, 三十多年来, 计算机软件发展迅速, 重要性与日俱增, 但其现状鲜能令人满意, 主要问题为:

1. 功能不强

这里所谓功能不强, 并非个体含义, 不是指某项具体软件的功能不强, 而是整体含义, 即在目前软件技术的条件下, 就总体而论, 计算机软件的功能不强。其主要表现是: 多数软件缺少智能, 难于处理知识。不少所谓"专家系统", 实质上只是储存、检索系统。早在 1945 年 A. M. Turing 就发表了"计算机和智能"一文, 但迄今未见重大突破, 至少软件领域是如此。

2. 质量欠佳

如所熟知, 衡量软件质量的定性标准有: 简明性, 可靠性, 坚定性, 易维性, 功效性等等。就影响可靠性的决定性因素"正确性"而言, 经过四十余年的努力, 从"静态检查"、"动态调试"、"测试", 到"正确性验证"等, 理论和实践均未很好解决。熟知的归纳断言方法的提出已逾二十年, 文章虽然发表不少, 但由于具有难以克服的缺陷, 如验证程序开发者须对被验证的程序了解透彻, 验证程序有时较被验证程序更为复杂, 以及验证程序本身也有正确性问题等等, 所以, 迄今该方法的实用范围还局限于一些小规模的程序。又如坚定性问题, 为了保证软件具有一定的坚定性, 必须付出相应的开销, 而且目前还处于"就事论事"阶段, 尚未提炼出行之有效的具有普遍意义的方法。由于受到目前软件技术水平的限制, 就总体而论, 软件质量问题急待深入探讨。

3. 生产率低

四十多年来, 软件人员一直在寻求提高软件生产率的有效途径, 从初期的使用低级语言编制程序, 到广泛采用高级语言, 使程序人员摆脱"与机器有关"的繁琐细节。1968 年提出了"软件工程"以来, 企图用工程方法编制程序, 开发软件。然而, 迄今开发软件的主要工作仍然依靠人力, 费时低效。和计算机硬件的发展速度相比, 软件的发展颇不理想。有的国家有人统计过, 大型软件项目中, 软件开发人员的平均年产量只不过 2000 条高级语言语句左右。主要原因在于大型软件接口复杂。现有的开发手段落后。

1.1.2 解决途径

为了解决上述问题,需采取改进传统技术与发展新技术并进的方针。

1. 改进传统技术

传统软件技术的基础是冯氏单机。其主要特征是:

(1) 本质串行。冯氏机器的本质串行性导致传统软件技术主要针对顺序计算。由于基础是单机,并发性表现为多任务的夹插执行,多机系统出现后,才可能考虑真正的并行。

(2) 实现级语言。传统程序设计过程仅当问题明确、解法选定、算法确定后,才选用合适的语言来描述算法,并能借助各类处理程序进行处理。其所用语言为用以描述算法的语言,习惯上称作实现级语言。古典高级语言从 FORTRAN, ALGOL, COBOL, PASCAL, 直到 ADA 均属此类。与使用低级语言相比,程序人员的负担有所减轻,但对计算机用户而言,要求仍苛。

(3) 开发方法自顶向下。结构程序设计方法出现后,风靡一时,符合人们思考问题、解决问题的过程,先总体,后局部,由粗到精,逐步精化。结构性表现为层次性,便于稳扎稳打,步步为营。软件开发的瀑布模型与结构程序设计方法本质一致。

(4) 软件本身缺乏智能。软件一经设计定型,其功能即已确定。由于软件本身缺乏智能,无法在工作过程中视需要。通过学习,扩大功能,以适应新的需求。

(5) 理论基础薄弱。理论基础主要为自动机理论,形式语言理论等等,并且着重语法研究,语义研究薄弱,语用研究很少开展。

传统软件技术的代表性工作有:实现级语言及其处理程序,单机操作系统,设计、实现级工具,数据库管理程序,以及正确性验证程序等。

2. 发展新技术

软件新技术可有如下几类:

(1) 智能化技术。它指的是如何使软件具备智能的技术。如前所述,由于传统软件本身缺乏智能,致使系统一经设计定型,功能即完全确定,无法视需要在其工作过程中扩大功能。因此,为使软件能多快好省地服务于各个领域,必须研究与开发各类具备智能的软件。智能化技术是扩大软件功能的关键途径,必须予以高度重视。

(2) 自动化技术。它指的是如何使软件开发过程自动化的技术。它是提高软件生产率的根本途径,自动化技术是本书的主题。

(3) 集成化技术。它指的是如何使软件具有集成性的技术。集成性一词有两层含义:一为内部集成性。为使一系统能成为真正的开放式系统,易扩充系统,其各个组成部分之间的接口必须一致,否则,就很难将各个组成部分汇集为一个整体,也很难往原有系统中添加新的组成部分,这就是内部集成性的真谛;二为外部集成性。为使一系统能嵌入另一系统中作为其组成部分,前者的外部接口必须和后者的内部接口一致,否则,就难于将前者嵌入后者,这就是外部集成性。总之,接口一致性构成集成性的实质。发展集成化技术有助于在原有软件的基础上开发新软件,既有助于提高生产率,也有助于提高软件质量。

(4) 并行化技术。它指的是如何使计算机系统对各个对象的处理适当并行,借以提高系统处理实效。并行性是现代计算机系统区别于传统计算机系统的重要标志,其实现一般

需软硬结合。从软件方面看,有所谓并行粒度问题。粗粒度者如程序与程序(任务与任务)的并行,中粒度者如模块与模块的并行,细粒度者如运算与运算的并行。并行化技术是提高系统实效的关键技术,也是难度很大的技术。

(5) 自然化技术。它指的是如何使人与计算机系统的接口界面尽可能采用自然语言的技术。如所熟知,社会信息化的主要特征是计算机及其服务业面向人人。当然不能也不必要求"人人"均了解人工语言。从这个意义上说,只有实现了软件自然化,用户与系统的接口界面尽可能采用自然语言,计算机系统才可能家喻户晓,才可能实现社会信息化。

1.2 基本含义

软件自动化一词几乎和软件一词同时出现。原称自动程序设计。早在50年代,程序人员从程序设计实践中深感程序设计工作的烦琐、不易、低效,便企图在可能范围内将一些机械性工作委交机器本身去做。在那时,实现高级语言的编译就是自动程序设计。60年代,出现了编译程序的编译程序,各种自编译程序。软件工程出现以后,软件自动化的含义有了较大发展,其自动化的内容涉及到软件生存全期的各个阶段。软件自动化一词具有如下几种含义。

1.2.1 广义理解

软件自动化的广义理解指的是,尽可能借助计算机系统,实现软件开发。值得注意的是,这里的计算机系统,除泛指一般计算机系统外,重要的是特指主要用于软件开发的系统,特别是软件自动化系统;其次,"尽可能"一词反映软件自动化的相对性,"尽可能"的程度反映系统的自动化程度;第三,软件开发指的是,从问题的非形式描述,经形式的软件功能规格说明,设计规格说明,到可执行的程序代码,调试,及至确认,交付使用的全过程。亦即,维护阶段除外的软件生存全期。

1.2.2 狭义理解

软件自动化的狭义理解指的是,从形式的软件功能规格说明到可执行的程序代码这一过程的自动化。要注意的是,可执行的程序代码既可指低级语言程序代码,也可指高级语言(当然是实现性语言)程序代码。此外,自动化的程度也是相对的。程度的高低一般随系统而异。

1.2.3 不同层次的理解

1. 纵向理解

· 低级自动化。自动化系统只起程序人员的作用。亦即,从软件设计规格说明到可执行的程序代码这一过程的自动化。

· 中级自动化。自动化系统除了起程序人员的作用外,还起设计人员的作用甚至部分系统分析人员的作用。亦即,从形式的软件功能规格说明到设计规格说明,一直到可执行的程序代码这一过程的自动化。

· 高级自动化。自动化系统除了起程序人员的作用、软件设计人员，系统分析人员的作用外，还起部分领域专家的作用。亦即，从非形式的软件需求描述，经形式的软件功能规格说明、软件设计规格说明，直到可执行的程序代码这一全过程的自动化。

2. 模向理解

在上述各种纵向理解级别上，取决于人工干预的程度，又可区分各种不同的自动化级别。

1.3 主 要 内 容

1.3.1 软件开发

前已提及，软件开发指的是从软件的需求描述，经形式的软件功能规格说明，设计规格说明，到可执行的程序代码，及至确认，交付使用的全过程，亦即，维护阶段除外的软件生存全期。

软件开发风范指的是软件开发的原则与风格。例如，自顶向下的功能分解风范，自底向上的对象式风范等等。

软件开发模型指的是，软件开发风范的结构体现。例如，功能分解风范中的瀑布模型，对象式风范中的喷泉模型等等。

一般说来，具体软件的开发并不是单纯的自顶向下或自底向上的单向过程，而是既有自顶向下，又有自底向上的双向反复过程，只不过是何者为主而已。此外，由于在有些应用领域中往往用户要求很快看到软件开发的思路是否合适。希望很快看到问题原型所产生的结果，从而对用户所提的需求摒弃次要部分，保留主要部分，快速生成原型，待其结果用户认为合适后，再逐步添加其它需求，最终制作出用户所需的软件，即所谓“演化”模型。

开发方法与工具则是开发模型的具体体现。方法与工具是一个问题的两个侧面。一方面，方法是工具的基础，工具要以方法为基准；另一方面，方法又有赖于工具来体现。方法有两类，一类是全局性的方法，如结构程序设计方法，两步走方法等；另一类是局部性方法，如主要用于支持软件设计阶段的 JACKSON 方法等。工具有多种，如支持需求分析阶段的需求分析工具，支持设计阶段的设计工具，支持实现阶段的实现工具，支持测试和验证的测试验证工具等等。

1.3.2 规格说明

软件规格说明是对软件所应满足的要求的陈述。它是软件开发的依据，也是软件自动化的依据。根据陈述的性质与层次，又可区分为需求规格说明(习惯上称为需求定义)、功能规格说明、设计规格说明等等。目前，需求规格说明还难以用形式体系刻划，一般要借助自然语言，不过，从社会信息化的要求看，自然语言的形式化工作实为刻不容缓，功能规格说明与设计规格说明均可用形式体系刻划。这时，习惯上称作形式功能规格说明与形式设计规格说明。

用以书写规格说明的语言称为规格说明语言，涉及的数学工具主要有逻辑与代数。目前有专门用于书写功能规格说明的功能规格说明语言，有专门用于书写设计规格说明的

设计规格说明语言,也有既可书写功能规格说明,又可书写设计规格说明的所谓广谱式规格说明语言。至于书写需求规格说明的语言,一般还难以采用纯人工语言。

1.3.3 自动生成

自动生成指的是从需求规格说明或功能规格说明(取决于软件自动化的不同含义)由相应的软件自动化系统"自动"生成可执行的程序代码。这项工作的难度颇大,迄今由需求规格说明过渡到功能规格说明还难以自动生成,尚须借助"人与系统"的交互。由功能规格说明到设计规格说明,原则上可以自动进行。实际上,仍需借助"交互",至于由设计规格说明(指详细设计规格说明,而不是概要设计规格说明)到可执行的程序代码则可自动进行,而且技术已相对成熟。因此,可以说,软件自动化系统的自动化程度越高,难度越大。另一方面,针对某一应用领域而设计的专用软件自动化系统实现较易,相反,通用软件自动化系统则实现较难。自动生成是软件自动化系统的核心,也是困难的关键所在。

1.3.4 自动验证

软件的正确性是相对其功能规格说明而言的。凡是具备且仅具备相应功能规格说明中所列出的功能的软件就是正确的软件。自动验证包括三方面的工作。第一,要自动验证功能规格说明的正确性,因为如果功能规格说明有误,就表明前提有误。但是,何谓功能规格说明的正确性?当然,我们可以说,功能规格说明的正确性是相对需求规格说明而言的;但是,如前所述,目前需求规格说明还难以用形式体系刻划。这样,问题就更为复杂。因此,功能规格说明的正确性的自动验证还处于探索阶段;第二,可执行的程序代码的正确性的自动验证;第三,从功能规格说明到可执行的程序代码的转换过程的正确性的自动验证;理论上说,在肯定功能规格说明正确的前提下,只需验证"转换过程"的正确性,作为其转换出的结果,可执行的程序代码也就必然正确。

自动验证是软件自动化系统必不可少的内容,但是,目前这方面的工作还不多,真正实用的方法更少,还有待于这一领域的工作者去努力耕耘。

1.4 实现途径

1.4.1 演绎综合

软件自动化的演绎综合途径将数学中的构造性证明与软件开发相联系,把每一步证明解释为一个计算步骤。形式上说,给定规格说明:

求 y,使得 Q(x,y)成立,前提是 P(x)成立。相应程序的推导等价于如下定理的构造性证明:

定理: "x ∨y[P(x) Q(x,y)]

这一途径适用于规格说明较易用逻辑语言形式化的小型程序,而且,对较大的证明问题,目前的定理证明程序还难以有效地控制演绎空间中的搜索。一般认为,它只能用来支持软件开发中的一些独立的开发步骤。

1.4.2 程序转换

1. 含义:

程序转换的含义是,由一程序转换至另一满足其功能要求的程序。

2. 类别:

有两类程序转换,一类是纵向转换,即由一抽象级别较高的程序转至另一满足其功能要求的抽象级别较低的程序。另一类是横向转换,即在相同(或类似)抽象级别上程序间的转换。

3. 纵向转换:

纵向转换的代表性工作是 70 年代中期慕尼黑技术大学信息学研究所在 F.L.Bauer 教授主持下开始研究的软件自动化系统,即计算机辅助、直觉指导的程序设计(CIP)项目。

·CIP 的目标、课题、特点

目标是开发可形式保证程序正确性的程序开发系统。其课题有三:第一,设计并定义一广谱语言;第二,开发一交互式系统;第三,建立一指导程序开发中形式推理过程的方法学。其特点是,第一,各步间转换的实现只借助"保证正确性"的转换规则;第二,开发全过程由程序人员指导,亦即,由程序人员选定转换规则。

·广谱语言 CIP-L

广谱语言 CIP-L 是一种融实现性语言、设计性语言、以及功能性语言为一体的语言。它是一种全类型化的一阶逻辑语言,具有易扩充性、抽象性、模块性,以及非确定性。

·转换

程序模式:程序模式指的是带模式参数的程序族。

转换规则:转换规则由程序模式有序对(输入模式 a,输出模式 b)和适用性条件 c 组成,其图式如图 1-1。

图 1-1

适用性条件:适用性条件有语法条件(用以指明特定的模式变量所代表的语法实体),前后文条件(一般为谓词,为 OCCURS(XinE),以及语义条件(用以指明实体在语义上应满足的条件)。

正确性:给定一合适关系 ψ(称作正确性关系),转换

$$P \rightarrow P'$$

称为正确的,当且仅当

$$P \psi P'$$

成立。由此推知,正确性关系的必要条件是:自反性与传递性。

规则分类:

第一类:基本规则(对语言核而言)。如 Unfold(将函数调用换成相应的函数体),与 Fold(Unfold 之逆);

第二类:对语言核除外的其它语言构造定义的规则;

第三类:抽象数据类型的公理;

第四类:推理规则;

第五类:由前述各类规则导出的规则。

转换步骤:

第一,匹配:将待转换的程序和输入模式匹配。如然,转至第二;否则,更换输入模式,直至匹配为止。

第二,验证:验证适用性条件是否满足。如然,转至第三;否则,失败。

第三,从相应的输出模式,即可得出转换结果。

·系统

CIP 转换系统由用户接口界面、系统核(作用有三:第一,初启并动用系统操作;第二,注视系统内态;第三,准备输出)以及知识库(存放转换规则,抽象类型,程序模式。程序开发树形等等)组成。

4. 横向转换

横向转换的代表性工作是英国爱丁堡大学 R. M. Burstall 和 J. Darlington 等人研制的 POP-2 系统和在此基础上发展起来的 ZAP 系统。这类系统根据一组规则基本自动地将作用式递归程序转换成高效的循环程序。规则中包括递归和冗余计算的消去规则,展开规则和结构共享规则等等。

·推理规则

定义:引进一新递归等式,其左方并非任何先前等式左方的实例,如

$$f(a,b,c,d,n) \quad dot(a,b,n) + dot(c,d,n)$$

例化:引进现存等式的一置换实例,例如,上述等式可例化为

$$f(a,b,c,d,o) \quad dot(a,b,o) + dot(c,d,o)$$

展:设

$$E \quad E' \text{ 与 } F \quad F'$$

均为等式,且 F 中出现有 E 的某一实例,将其换成 E 的相应实例,得出 F",添加等式 F F"。

卷:设

$$E \quad E' \text{ 与 } F \quad F'$$

均为等式,且在 F 中出现有 E 的某一实例,将其换成 E 的相应实例,得出 F",添加等式 F F"。

抽象:引进 where 子句由先前等式

$$E \quad E'$$

导出一新等式: