

程序复杂性度量

王振宇 著

国防工业出版社



程序复杂性度量

PROGRAM COMPLEXITY METRICS

王振宇 著
WANG ZHENYU

国防工业出版社

· 北京 ·

图书在版编目(CIP)数据

程序复杂性度量/王振宇著. —北京:国防工业出版社,
1997.8

ISBN 7-118-01692-6

I. 程… I. 王… III. 程序复杂性-度量 N. TP311.11

中国版本图书馆 CIP 数据核字(97)第 00254 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 100044)

北京怀柔新华印刷厂印刷

新华书店经售

*

开本 850×1168 1/32 印张 5 $\frac{3}{4}$ 141 千字

1997 年 8 月第 1 版 1997 年 8 月北京第 1 次印刷

印数:1—1000 册 定价:13.50 元

(本书如有印装错误,我社负责调换)

致 读 者

本书由国防科技图书出版基金资助出版。

国防科技图书出版工作是国防科技事业的一个重要方面。优秀的国防科技图书既是国防科技成果的一部分,又是国防科技水平的重要标志。为了促进国防科技事业的发展,加强社会主义物质文明和精神文明建设,培养优秀科技人才,确保国防科技优秀图书的出版,国防科工委于1988年初决定每年拨出专款,设立国防科技图书出版基金,成立评审委员会,扶持、审定出版国防科技优秀图书。

国防科技图书出版基金资助的对象是:

1. 学术水平高,内容有创见,在学科上居领先地位的基础科学理论图书;在工程技术理论方面有突破的应用科学专著。
2. 学术思想新颖,内容具体、实用,对国防科技发展具有较大推动作用的专著;密切结合科技现代化和国防现代化需要的高新技术内容的专著。
3. 有重要发展前景和有重大开拓使用价值,密切结合科技现代化和国防现代化需要的新工艺、新材料内容的科技图书。
4. 填补目前我国科技领域空白的薄弱学科和边缘学科的科技图书。
5. 特别有价值的科技论文集、译著等。

国防科技图书出版基金评审委员会在国防科工委的领导下开展工作,负责掌握出版基金的使用方向,评审受理的图书选题,决定资助的图书选题和资助金额,以及决定中断或取消资助等。经评审给予资助的图书,由国防工业出版社列选出版。

国防科技事业已经取得了举世瞩目的成就。国防科技图书承

担着记载和弘扬这些成就,积累和传播科技知识的使命。在改革开放的新形势下,国防科工委率先设立出版基金,扶持出版科技图书,这是一项具有深远意义的创举。此举势必促使国防科技图书的出版,随着国防科技事业的发展更加兴旺。

设立出版基金是一件新生事物,是对出版工作的一项改革。因而,评审工作需要不断地摸索、认真地总结和及时地改进,这样,才能使有限的基金发挥出巨大的效能。评审工作更需要国防科技工业战线广大科技工作者、专家、教授,以及社会各界朋友的热情支持。

让我们携起手来,为祖国昌盛、科技腾飞、出版繁荣而共同奋斗!

国防科技图书出版基金
评审委员会

国防科技图书出版基金

第二届评审委员会组成人员

名誉主任委员	怀国模		
主任委员	黄宁		
副主任委员	殷鹤龄	高景德	陈芳允
	曾铎		
秘书长	刘琯德		
委员	尤子平	朱森元	朵英贤
(按姓氏笔划为序)	刘仁	何庆芝	何国伟
	何新贵	宋家树	张汝果
	范学虹	胡万忱	柯有安
	侯迁	侯正明	莫梧生
	崔尔杰		

序

自从 Halstead 的名著 *Elements of Software Science* 问世,程序复杂性度量已经有了 20 年左右的历史。但客观地说,它还是一项成熟的软件技术,原因很简单,那就是我们对程序这种人类发明的产物认识得还不够,对程序的复杂性依然知之甚少。但是,计算机程序或软件对人类又是那么重要,它们的复杂性又被公认是已经困扰我们多年的“软件危机”的根源之一,因而无法回避它们。面对这种挑战,我们只有继续研究,加深认识。我相信,有朝一日,程序复杂性度量一定能够成为软件开发和维护的有用工具和手段。

我本人对程序复杂性度量的研究始于 80 年代初,当时就有赵一心同志参加了研究工作。那时只算是个人兴趣使然。到了 80 年代末至 90 年代,先后得到了国家自然科学基金和国防预先研究项目的支持,从顺序的结构化程序的复杂性度量又转到 Ada 程序的并发复杂性度量。本书反映了个人和研究小组的有关研究成果。

关于程序复杂性度量,虽然每年都有大量论文发表,但除了 Halstead 的那本 20 年前的名著 (*Elements of Software Science*) 外,国内几乎再也看不到系统的著述。这本书,一方面是个人和研究小组的工作总结,另一方面又力图能有一定的系统性,对相关工作进行必要的阐述。

书中不妥甚至谬误之处,恳请软件界同行不吝指正。

致 谢

书中多次引用了国内外同行的研究成果,在引用的地方都予明确标出。这里向这些作者们表示由衷的谢意。

衷心感谢国家自然科学基金委员会信息科学部、国防科工委及系统工程研究所、中国船舶工业总公司、第七研究院、第七〇九研究所有关领导和机关对这项研究的支持。

研究过程中,先后有赵一心、陈利、王志江(华中师范大学计算机科学系)、陈敏、赵恒、张立等同志参加,特别是在方法的程序实现方面,书中的部分结果也是共同劳动的产物。在软件度量和测量科学的关系方面,同王宪荣同志进行过有益的讨论。

此外,南京大学徐家福教授、国防科技大学陈火旺教授、国防科工委系统工程研究所何新贵教授也对这项研究给了很多鼓励。国防科技图书出版基金评委会对此书的出版给予了大力支持,谨此一并致谢。

王振宇 识

1996年6月于武昌

目 录

第一章 引论	1
1.1 程序复杂性度量概念及其研究范围	1
1.2 程序复杂性度量在软件开发中的作用	3
1.3 程序复杂性度量的研究现状	4
1.3.1 程序复杂性度量的模型研究	5
1.3.2 程序复杂性度量方法的研究	5
1.3.3 程序复杂性度量工具的研究	7
1.4 程序复杂性度量与算法复杂性分析的分工	7
第二章 软件度量与测量	10
2.1 测量科学的一般概念	11
2.2 测量的表示理论	12
2.3 测量的用途	13
2.4 直接测量和间接测量	14
2.5 标度(scale)类型和意义	14
2.6 Weyuker 性质	18
2.7 测量过程	21
第三章 顺序程序复杂性度量	23
3.1 顺序程序复杂性的模型	23
3.2 行计数法	24
3.3 Henry-Kafura 的信息流度量 ^[5]	25
3.4 Thayer 复杂性度量 ^[31]	26
3.5 McCabe 的着色数法度量 ^[2]	29
3.5.1 程序的有向控制图	29
3.5.2 McCabe 度量的图论基础	31

3.5.3 McCabe 度量的用途	32
3.5.4 Ada 程序的一个 McCabe 度量工具	35
3.5.5 对 McCabe 复杂性度量的评论和修改意见	37
3.6 结构—路径复杂性度量	38
3.6.1 Cantone 的特征多项式	38
3.6.2 程序结构—路径复杂性度量的形式定义	39
3.6.3 程序循环嵌套结构的树图表示	41
3.6.4 结构—路径复杂性度量的性质	44
3.6.5 非递归子程序调用对复杂性的影响	50
3.6.6 递归调用对复杂性的影响	52
 第四章 软件科学法	 54
4.1 运算符与运算元	54
4.2 程序长度	57
4.3 程序容量(volume)	59
4.3.1 潜在容量(potential volume)	60
4.3.2 运算符和运算元之间的关系	62
4.4 程序级别	63
4.5 智慧量	65
4.6 程序的不纯洁性	65
4.7 编程工作量	66
4.8 语言级别	68
4.9 预测程序的长度和编程时间	69
4.9.1 自 η_2^* 和 λ 求 η_1	69
4.9.2 自 η_2^* 和 λ 求编程工作量和时间	70
4.10 出错假设	70
4.11 模块化	72
4.12 对软件科学法的评论	73
4.13 对软件科学法的修改意见	74
 第五章 信息隐藏度量	 76
5.1 信息隐藏概念的发展	76
5.2 信息隐藏度量	78
5.2.1 模块级度量	78

5.2.2 系统级度量	79
5.3 主观验证	81
5.4 事例研究验证	82
5.5 结论	88
 第六章 Ada 程序并发复杂性度量	 90
6.1 并发程序的特征及程序并发复杂性研究现状	90
6.1.1 并发程序的特征	90
6.1.2 程序并发复杂性研究现状	91
6.1.3 对程序并发复杂性度量研究现状的分析	92
6.2 Ada 语言的并发机构	93
6.3 Shatz 关于分布式程序复杂性度量的研究	97
6.4 不确定性对 Ada 并发程序复杂性的影响	99
6.4.1 同接受语句关联的不确定性	102
6.4.2 同选择语句关联的不确定性	104
 第七章 基于软件科学法的 Ada 并发复杂性度量	 106
7.1 对顺序复杂性和并发复杂性不加区分的度量	106
7.1.1 Tsalidis 和 Hatzimanikatis 等的 AFADA	106
7.1.2 VAX/VMS 平台上的软件科学法度量 HALSTEAD	107
7.2 并发程序的运算符—运算元模型	109
7.3 Ada 程序中的并发运算、运算符和运算元	111
7.4 软件科学法用于 Ada 并发复杂性度量的几个指标	115
7.5 关于运算符和运算元计数的几点考虑	115
7.6 SSACC 的实现考虑	117
7.7 Ada 程序复杂性度量工具 SSAC	117
 第八章 基于会合关系的 Ada 并发复杂性度量	 124
8.1 引言	124
8.2 Ada 会合之间的关系和关系的分类	125
8.2.1 先后或接续关系	125
8.2.2 嵌套关系	126
8.2.3 会合关系表	127
8.2.4 独立关系	129

8.3 会合接续关系和会合接续图	130
8.4 会合接续图的边数作为度量并发 Ada 程序复杂性的一个指标	137
8.5 会合接续图涉及的任务数	138
8.6 会合嵌套序列	140
8.7 会合嵌套树	145
8.8 会合嵌套树的加权节点数作为并发程序复杂性度量的一个 指标	147
8.9 基于会合关系的 Ada 并发复杂性度量	149
8.10 会合次序图	150
8.10.1 基本思路	152
8.10.2 会合次序图的概念	152
8.10.3 会合次序图的定义	153
8.10.4 嵌套关系转化为接续关系	154
8.10.5 不确定性的处理	159
8.10.6 另外一些会合次序图的例子	161
8.10.7 几点注记	162
8.11 会合次序图的复杂性度量	163
参考文献	164

Contents

Chapter 1 Introduction

1.1	Concept and scope of program complexity metrics	1
1.2	Role of program complexity metrics in software development	3
1.3	Research status of program complexity metrics	4
1.3.1	Research on Models of program complexity metrics	5
1.3.2	Research on methods of program complexity metrics	5
1.3.3	Research on tools of program complexity metrics	7
1.4	Differences between program complexity metrics and algorithm complexity analysis	7

Chapter 2 Software metrics and measurement

2.1	General concept of measurement science	11
2.2	Representation theory of measurement	12
2.3	Use of measurement	13
2.4	Direct measurement and indirect measurement	14
2.5	Scale type and its meaning	14
2.6	Weyuker properties	18
2.7	Measurement process	21

Chapter 3 Sequential program complexity metrics

3.1	Model of sequential program complexity	23
3.2	Line counting method	24
3.3	Henry-Kafura's information stream metrics	25
3.4	Thayer's program complexity metrics	26
3.5	McCabe's cyclomatic number metrics	29
3.5.1	Directed control diagram of programs	29

3.5.2	Graph-theoretic basis of McCabe's metrics	31
3.5.3	Use of McCabe's metrics	32
3.5.4	A tool of McCabe's metrics for Ada program	35
3.5.5	Reviews and improvement to McCabe's complexity metrics	37
3.6	Structure-path program complexity metrics	38
3.6.1	Cantone's characteristic polynomial	38
3.6.2	Formal definition of structure-path program complexity metrics	39
3.6.3	Tree diagram representation of loop nesting structure in programs	41
3.6.4	Properties of structure-path program complexity metrics	44
3.6.5	Influence of non-recursive procedure calls to complexity	50
3.6.6	Influence of recursive procedure calls to complexity	52

Chapter 4 Software Science method

4.1	Operators and operands	54
4.2	program length	57
4.3	Program volume	59
4.3.1	Potential volume	60
4.3.2	Relations between operators and operands	62
4.4	Program levels	63
4.5	Intelligence content	65
4.6	impurity in program	65
4.7	Programming efforts	66
4.8	Language levels	68
4.9	Prediction of program length and programming time	69
4.9.1	Finding η_1 from η_2^* and λ	69
4.9.2	Finding programming efforts and time from η_2^* and λ	70
4.10	Error hypothesis	70
4.11	Modularization	72
4.12	Reviews on software science method	73
4.13	Improvement comments on software science method	74

Chapter 5 Information hiding metrics

5.1	Development of information hiding concept	76
-----	---	----

5.2	Information hiding metrics	78
5.2.1	Metrics of module level	78
5.2.2	Metrics of system level	79
5.3	Subjective metrics	81
5.4	Case study verification	82
5.5	Conclusions	88

Chapter 6 Concurrent program complexity metrics of Ada programs

6	Features of concurrent programs and research status of concurrent program complexity metrics	90
6.1.1	Features of concurrent programs	90
6.1.2	Research status of concurrent program complexity metrics	91
6.1.3	Analysis of research status of concurrent program complexity metrics	92
6.2	Concurrent mechanism of Ada language	93
6.3	Shatz's research on distributed program complexity metrics	97
6.4	Influence of non-determinism to Ada concurrent program complexity metrics	99
6.4.1	Non-determinism related to accept statements	102
6.4.2	Non-determinism related to select statements	104

Chapter 7 Ada concurrent program complexity metrics based on software science method

7.1	Metrics	106
7.1.1	AFADA of Tsalidis, Hatzimanikatis et al	106
7.1.2	HALSTEAD, a software science metrics on VAX/VMS platform	107
7.2	Operator-operand model of concurrent program	109
7.3	Concurrent operations, operators and operands in Ada programs	111
7.4	Some indices in using software science to Ada program complexity metrics	115
7.5	Considerations on operator and operand counting	115

7.6	Implementation considerations of SSACC	117
7.7	SSAC, an Ada program complexity metrics tool	117

Chapter 8 Ada concurrent program complexity metrics based on rendezvous relations

8.1	Introduction	124
8.2	Relations between rendezvouses and classification of the relations	125
8.2.1	Preceding or successive relation	125
8.2.2	Nesting relation	126
8.2.3	Rendezvous relation table	127
8.2.4	Independent relation	129
8.3	Rendezvous successive relation and rendezvous successive graph	130
8.4	Edge number as an index in measuring concurrent Ada program complexity metrics	137
8.5	Number of tasks involved by a rendezvous successive graph	138
8.6	Rendezvous nesting sequence	140
8.7	Rendezvous nesting tree	145
8.8	Weighted node number as an index in measuring concurrent Ada program complexity metrics	147
8.9	Ada concurrent program complexity metrics based on rendezvous relations	149
8.10	Rendezvous ordering graph	150
8.10.1	Basic idea	152
8.10.2	Concept of rendezvous ordering graph	152
8.10.3	Definition of rendezvous ordering graph	153
8.10.4	Transformation of nesting relation to successive relations	154
8.10.5	Handling non-determinism	159
8.10.6	More examples of rendezvous ordering graphs	161
8.10.7	Some notes	162
8.11	Complexity metrics of rendezvous ordering graph	163
	References	164

第一章 引 论

大家知道,自 60 年代以来,“软件危机”一直困扰着计算机界和软件界。大家也都意识到,随着计算机应用的规模和复杂性的增加,计算机系统的网络化和分布化,“软件危机”也主要来自软件的规模越来越大、越来越复杂,或者说,复杂性太高。但软件(或程序)复杂性的定义是什么?影响软件(或程序)复杂性的,究竟是什么因素?是不是程序越长、计算量越大,程序就越复杂?它同计算复杂性有什么关系和异同?

软件工作者必须充分而准确地理解用户的需求。用户最根本的三点需求是:所需软件产品的质量、交付的时间和费用。也可以把这三点看作是软件产品最基本的特性。而谈论了近 30 年的“软件危机”的基本表现形式也恰好是:软件质量不能满足用户要求;软件产品不能按预定的时间交付;开发成本超过预定计划。把它们说成一句话就是:不能在预定时间、花费预定成本开发出达到预定质量的软件。那么,软件(或程序)复杂性又是以什么方式影响软件的质量、交付和开发成本的呢?

这些问题不是用几句话就能回答的,而且也不是本书所能全面回答的。然而,本书的主题“程序复杂性度量”确实是同这些问题密切相关的。本书力图阐明程序复杂性度量的有关理论、技术和方法,并探讨它同这些问题的关系。

1.1 程序复杂性度量概念及其研究范围

按照自然科学和工程领域传统的定义,度量是用量值来评价对象和实验现象的某些属性的工具。当然,并不是每一种对象或实