高等院校精品教材

软件工程形式化方法与语言

李 莹 吴江琴 编著

ZHEJIANG UNIVERSITY PRESS
浙江大学出版社

软件工程形式化方法与语言

李　莹　吴江琴　编著

ZHEJIANG UNIVERSITY PRESS
浙江大学出版社

图书在版编目（CIP）数据

软件工程形式化方法与语言/李莹，吴江琴编著. —杭州：浙江大学出版社，2010.3

ISBN 978-7-308-06667-9

Ⅰ.软… Ⅱ.吴… Ⅲ.软件工程 Ⅳ.TP311.5

中国版本图书馆 CIP 数据核字（2009）第 041150 号

软件工程形式化方法与语言

李　莹　吴江琴　编著

责任编辑　杜希武
封面设计　刘依群
出版发行　浙江大学出版社
（杭州市天目山路 148 号　邮政编码 310007）
（网址：http://www.zjupress.com）
排　　版　杭州中大图文设计有限公司
印　　刷　浙江省良渚印刷厂
开　　本　787mm×1092mm　1/16
印　　张　13.25
字　　数　322 千
版 印 次　2010 年 3 月第 1 版　2010 年 3 月第 1 次印刷
书　　号　ISBN 978-7-308-06667-9
定　　价　29.00 元

版权所有　翻印必究　　印装差错　负责调换

浙江大学出版社发行部邮购电话　(0571)88925591

前　言

软件形式化方法最早可追溯到20世纪50年代后期对于程序设计语言编译技术的研究，即J. Backus提出BNF描述Algol60语言的语法，出现了各种语法分析程序自动生成器以及语法制导的编译方法，使得编译系统的开发从“手工艺制作方式”发展成具有牢固理论基础的系统方法。形式化方法的研究高潮始于20世纪60年代后期，针对当时所谓“软件危机”，人们提出种种解决方法，归纳起来有两类：一是采用工程方法来组织、管理软件的开发过程；二是深入探讨程序和程序开发过程的规律，建立严密的理论，以其用来指导软件开发实践。前者导致“软件工程”的出现和发展，后者则推动了形式化方法的深入研究。经过30多年的研究和应用，如今人们在形式化方法这一领域取得了大量、重要的成果，从早期最简单的形式化方法——一阶谓词演算方法到现在的应用于不同领域、不同阶段的基于逻辑、状态机、网络、进程代数、代数等众多形式化方法。形式化方法的发展趋势逐渐融入软件开发过程的各个阶段，从需求分析、功能描述(规约)、(体系结构/算法)设计、编程、测试直至维护。可以说，形式化不仅仅是对用户需求，而且也是对整个软件系统的严格定义。传统的软件开发方法由于大量的使用自然语言和多种图形符号，结果是尽管经历了仔细地复审，最后的系统规约说明中仍然包含歧义的、含糊的、矛盾的、不完整的需求描述及混乱的抽象层次。使用形式化方法可以克服这些缺点。

形式化方法的本质是基于数学的方法来描述目标软件系统属性的一种技术。不同的形式化方法的数学基础是不同的，有的以集合论和一阶谓词演算为基础，有的则以时态逻辑为基础。形式化方法需要形式化规约说明语言的支持。由于形式化方法种类非常多，本书选取并介绍了三种代表性的形式化方法，它们分别是以集合论和一阶谓词演算为基础的Z语言，以时态逻辑为基础的XYZ，还有以直觉数学学派为基础的类型理论。

本书既可以作为计算机专业的研究生的形式化课程教材，又可以用作专业人员的参考书。虽然真正从事形式化方面的工作的人员不多，但是有必要通过对该课程的学习，使学生在理论、技术和方法上都得到了系统而有效的训练，有利于提高软件人员的素质和能力。

全书有16章节，可以分为以下几个部分：

第一部分是从第1章到第10章，介绍了Z语言的背景知识，包括集合论和一阶谓词演算等概念及其形式化表达方法；

第二部分是从第11章到14章，介绍了Z语言构型及其规格说明的结构化，引入求精理论；

第三部分是第15章，介绍了Martin-Löf类型理论，及其规则定义和推导演算；

第四部分是第16章，介绍了XYZ系统在时序逻辑语言方面的主要内容。

本书由李莹编写第一章到第十四章，由吴江琴编写第十五章和第十六章。另外，在本书的编写整理过程中也得到了蒋健、金路等老师和同学们的帮助，在此表示深切的谢意。由于水平有限，书中难免有一些不妥之处，敬请广大读者批评指正。

作　者

2008年9月

目 录

试读结束：需要全本请在线购买： www.ertongbook.com

第1章

引 论

1.1 软件工程

软件工程是研究构造高质量的软件的工程方法。这里软件意指大型程序。针对小型程序的大部分设计与编程技术，不能简单地推广到大型软件的开发中来。指导软件工程活动的一个重要原则，是软件的质量。这里所说的质量有内在质量和外部质量两个方面。外部质量包括：正确性、健康性、可扩展性、可重用性和效率；内在质量包括模块性和连续性。现在，让我们先简单地定义一下这些有关的软件质量的概念。

正确性和可靠性，是软件系统执行由它的需求定义与规格说明(specification)所定义的服务的能力。

健康性，是软件系统在非正常情况下继续正常工作的能力。

可扩展性，是软件系统适应对其需求定义及规格说明进行修改的容易程度。

可重用性，是软件模块作为构造其他应用的新软件的元件而重用的能力。

效率，是软件充分使用计算机各种资源以缩短完成其功能所需的时间的能力。

模块性，是软件分解为若干个通过固有且简单的接口相联系的自主元件的能力。模块性不仅在实现级很重要，在设计级也很重要。

连续性，是软件系统的这样一种性质，在对该软件的需求定义做少量的修改之后，不致引起对该软件系统做大的修改。这也意味着，需求定义的微小改动，只影响少量模块，不影响软件的整体结构。因此，连续性与模块性紧密相关。由于任何软件都可能在需求定义上有所改动，特别是在软件生存期的维护阶段，因此，连续性是一种内在质量指标，在软件开发中起非常重要的作用。

1.2 软件生存期

任何东西都有一定的生存周期。软件的生存周期可分为如图 1-1 所示的几个阶段：

注意，测试这项活动并不是一个阶段的普通活动。维护阶段不仅包括修正各种错误，而且包括软件系统的升级所必需的各种修改。因此，维护通常包括重复其他阶段的各项活动。

上面的软件生存期几个阶段的划分，勾画出软件开发过程的基本模型。但是，由于项目

的不同，项目的性质不同，开发队伍的素质不同，开发环境的不同，对模型的几个阶段的提法和着重点，有不同的理解。但是，关于一些重要问题的提法是一致的。它们是：

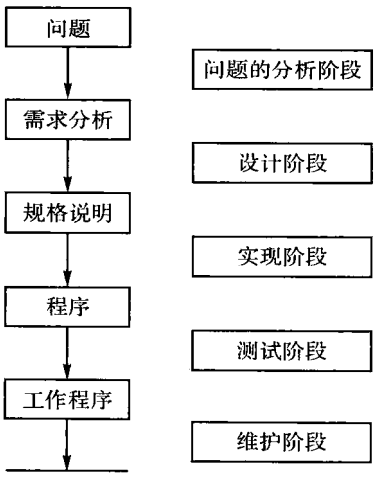

图 1.1 软件生存期

- 需求分析

客户与供货商会做明确理解并写下要解决的实际问题是什么。

- 系统的规格说明

系统必须满足的要求按双方一致认可的形式写下的协议。这包括系统必须实现的功能，以及包括性能、可靠性等外部质量指标的其他属性。

- 设计

根据可用的设备和所要求的性能，阐明满足规格说明的实现途径。这是上下两个阶段的联系桥梁，前一阶段主要描述“做什么”，这一阶段描述“如何做”，下一阶段更详细地描述“如何做”。

- 实现

把设计阶段中的描述转化为可执行的程序。

- 测试和集成

在完成部件测试之后，将其组装成完整统一的系统，并进行系统的测试，检查系统的正确性。

- 维护

完成了的系统要提交给最终用户，其整个生存期中，还可能要修正错误，增加新的特征，使其适应不同的使用环境。

上面的生存期模型是一般性的，还可以给出更详细的划分。但是，重要的问题是认识每个部分的不同要求，而不在于它们的详尽程度。

1.3 早期工作的重要性

大量的软件项目的经验说明，软件产品的开发或研制成本中，大部分都花在实际中的错

试读结束：需要全本请在线购买： www.ertongbook.com

误和测试维护之中。但是，更进一步的研究揭示，绝大部分错误都是由于项目开发过程中的早期阶段中的不精确而引起的。这就是说，实现和测试维护阶段中的高成本是由于规格说明和设计阶段的种种问题所引起的。

据有关资料统计，修正已交付用户的软件产品中的一个错误，大约要花费一万美元[FAG76]。一个已交付给许多用户的大中型系统，仅一百个错误就要增加一百万美元的开发成本。因此，在生存期的早期阶段，多花费一些精力，使用一些强有力的技术，以减少规格说明及设计中的错误，从而大量降低系统的总开发成本，是一件十分重要和有意义的工作。

1.4　规格说明及其形式化

简而言之，一个软件的规格说明，是关于整个软件是“干什么”的简单描述。整个描述可以用于不同的目的。例如：

- 用作程序的文档。
- 用作软件设计者与使用者之间的契约或协议。它描述双方的义务与权利，规定在何种条件下使用该软件，以及使用该软件产生的结果。
- 用作生存期的以后各个阶段的开发依据。一个好的规格说明，不仅对设计者有用，并且，对实现者和维护者都有用。规格说明的模块性将产生程序的模块性。规格说明是编程者在编程时使用的蓝图。
- 用于软件认证中形成认证程序集，认证中的测试实例，可以从规格说明产生。

由于规格说明的重要性，如何写好它就是一个值得重视的问题。总的说来，规格说明应当是完整的、一致的、精确的、紧凑的和无歧义的。凡自然语言写的规格说明，由于语言本身的问题，这个规格说明中，许多是不完整的、前后不一致的、不精确的和有歧义的。我们称其为非形式的规格说明。利用基于某些数学概念和记号的语言编写的规格说明，称为形式化的规格说明。由于此类基于数学概念的语言本身意义的精确性与无歧义性，它使软件系统的设计人员能够精确地描述这个软件是干什么的，它的行为特征是什么。形式化的规格说明的一个重要优点是，它使设计者能够利用严格的数学推理。该规格说明的性质可以像数学中的定理一样进行推理与证明。因此，在写出规格说明之后，编程实现之前，就可以查出设计错误。例如，不一致性与不完全性。在形式化的规格说明中进行数学推理的另一个好处是，能够形式地验证实现(程序)是否满足它的规格说明，是否一致。这两方面的工作实际上就是程序正确性证明。

构造型的形式化规格说明，可以直接执行(可能性能很差)。因而，它可用于快速原型法软件开发技术之中。利用构造型的形式化规格说明，人们能够自上而下地设计，自上而下地验证，自上而下地测试。这里采用的自上而下的概念的意思是，规格说明在任何实现之前进行。构造型的形式化规格说明的这一优点，在快速原型软件开发中得到广泛的运用。这就使得软件的设计者与用户之间得到良好的沟通，在实现这个软件的编程工作开始之前，获得该软件的第一年使用经验，获得用户的正确反馈信息，为改进系统设计提供可靠的依据。

既然形式化的规格说明相对于非形式化的规格说明有那么多的优点，是否可以说非形式化就没用了呢？非也！在当前软件工业界，非形式化方法还是占统治地位的。主要原因

是，形式化方法对软件设计人员的素质要求很高，因而难于大规模普及。形式化方法的开发工具与环境还有待于进一步开发，证明技术还不够成熟。因此，这两种方法是相辅相成的，相互补充的。对于至关重要的软件(或部件)，采用形式化技术，以确保其质量。对于一般性工作量极大的软件，采用非形式技术。但是，可以预见：随着软件科学技术的不断发展，自动化的、形式化的软件技术将得到广泛的应用。

1.5 一些重要的形式化规格说明语言

为了书写形式化的规格说明，许多计算机科学家从不同的角度，提出了许多不同的形式化规格说明语言。由于所根据的数学基础不同，方法与途径不同，形成了以下几个主要流派，它们是：

- 公理方法，利用前置条件与后置条件描述程序的行为。这个学派的代表人物有 Floyd，Hoare 和 Dijkstra【Floyd67，Hoare72，Dijkstre76】。
- 基于集合论和一阶谓词演算的 meta—IV 语言和 Z 语言。这种语言已广泛用于书写大型软件的规格说明与设计。在描述程序语言的指称语义时，利用这类语言可以方便地定义高阶函数，并由此定义程序语言的复杂控制构造的意义。利用 meta—IV 描述的形式化软件开发方法，称为维也纳开发方法，简称 VDM。本书采用 Z 语言来写软件的规格说明及其设计。
- 代数规格说明，是关于抽象数据类型的代数描述。代数规格说明语言有 OBJ【Gog88】及 ACT【Ehrig83】。
- 进程描述语言，用于描述开发进程的行为。主要有 Hoare 的顺序通讯进程 CSP【Hoa85】及 R·Milner 的通讯系统理论 CCS【Mil80】。
- 专用的规格说明语言，例如，在计算机网络与通讯系统中，广泛地使用形式化方法来研制与开发各种网络协议，已有大量的成功实例，形成了若干个由国际标准化组织认可的语言。例如，ISO LOTOS，ISO ESTELLE，ISO SDL，CCITT Z·100，CCITT SDL 等。

1.6 关于本书使用的 Z 语言

形式化规格说明语言 Z 是由英国牛津大学 1979 年第一次提出来的极高级的语言。它很适应于写一般的计算机系统的规格说明。现在，它已在工业界得到使用，有了许多成功的经验。1989 年，国际标准化组织已经公布了它的正式标准。

Z 语言建立于集合论和数理逻辑的基础上。集合论包括标准的集合运算符、笛卡尔积和幂集。数理逻辑包括一阶谓词演算。二者合二为一。形成了一个易学易用的数学语言。

Z 语言的第二个特点是它是使数学得以结构化的方式。数学对象与它上面的操作结合起来形成构型(schema)。构型语言可被用来描述系统的状态及改变系统的性质，对一个设计的可能求精细化进行推理。

Z 语言是一个强类型系统。数学语言中的每个对象都有唯一的类型，类型作为当前的

规格说明中一个最大集合来表示。类型在程序设计实践中是非常有用的概念，据此可以检查一个规格说明中每个对象的类型的一致性。

Z 语言的第三个特点是可以使用自然语言。人们用数学陈述问题，发掘解法，证明所作的设计满足规格说明的要求。同时，人们可以使用自然语言将数学与现实世界中的对象相关联，因为人们可以选择富有含义的变量名和辅以 V 的注文。一个好的规格说明应当是读者一看就懂的白话文。

Z 语言的第四个特点在它的求精。通过构造一项设计的模型，利用简单的数学类型标识所需的行为，可以开发一个系统。通过构造关于一个系统的设计决策的另一个模型，作为第一个模型的一个实现，就是一次求精。这种求精的过程可以一直继续到产生可执行的代码。

因此，Z 语言是具有强大构造机构的数学语言。同自然语言结合起来，它可被用来产生形式化的规格说明。利用数理逻辑的证明技术，可以对执行规格说明进行推理。对一个规格说明进行求精，得到接近于可执行代码的另一个描述。

但是，Z 语言没有提供关于计时的或并发的行为的描述。但是，其他某些语言适于作这样的描述，如 CSP 和 CCS。可以将 Z 与这些形式化方法结合起来，产生含并发行为的系统的规格说明。

第2章

命题逻辑

本章介绍Z语言的逻辑语言部分。在论述之中，我们始终贯穿一根主线－推理与证明：在陈述语言的每个成分时，总要说清楚，何时引入这个成分，何时消去这个成分。

将这些成分拼起来，这些规则就形成一个自然演绎系统：从一命题可推演出什么，什么条件下这个命题成立。这样，就给出了Z语言中的命题推理、性质证明和结果建立的框架。

2.1 命 题

命题是关于事实的陈述。这种事实或为真或为假，但不能既真又假。

例如，下面几句话都是命题：

- 苹果是水果。
- 西红柿是水果。
- 苹果不是唯一的水果。

真的命题具有值真(true)，假的命题具有值假(false)。

Z语言中，可以用5种运算符将命题连接起来，下表按运算符优先级别由高至低的排列：

¬	否定	非
∧	合取	与
∨	析取	或
⇒	蕴含	蕴含
⇔	等价	等价于

表中三列分别给出了运算符的符号、名字及中文读法。利用这些运算符可以形成新的命题，即复合命题。例如：

$\neg p \vee q \vee r \Leftrightarrow q \Rightarrow p \wedge r$

等价于

$(((\neg p) \wedge q) \vee r) \Leftrightarrow (q \Rightarrow (p \wedge r))$

复合命题的值由其组成部分唯一确定。

2.2 合　取

合取式 $p \wedge q$ 的值为真，当且仅当 p 的值和 q 的值都为真。这可由下面的真值表说明：

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

现在，如果要证明 $p \wedge q$ 为真。根据真值表，就要证明 p 和 q 都为真。如果已知 $p \wedge q$ 为真可断定 p 为真时 q 为真。这样，我们可把这两件事总结为下面两条推理规则：

$\frac{p \quad q}{p \wedge q}[\wedge +]$　$\frac{p \wedge q}{p}[\wedge -1]$　$\frac{p \wedge q}{q}[\wedge -2]$

这三条规则是自然演绎系统的一部分：

$\frac{\text{前提 1}\cdots\text{前提 n}}{\text{结论}}$【名字】　附属条件

推理规则中的前提表可以没有。推理规则的意义是：当前提为真时，结论也为真。

演绎系统的规则有两种。对于运算符 OP，OP 消去规则描述从 p OP q 中推出什么，OP 引进规则描述在什么条件下推出 p OP q。利用这些规则引入和消去不同的运算符。就可能从一组命题或假设，推出其他命题。如果假设是空的，推出来的命题就被称为定理

现在我们有两种证明方法。一种是通过构造所要证明的命题的真值表；另一种是利用引入和消去规则。

问题：证明 $p \wedge q$ 可交换。

方法一：构造两个真值表：

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

p	q	$q \wedge p$
T	T	T
T	F	F
F	T	F
F	F	F

把两张表合成一张表，得到：

p	q	p∧q	q∧p
T	T	T	T
T	F	F	F
F	T	F	F
F	F	F	F

注意，p∧q 与 q∧p 这两列完全相同：在任何情况下，二者取相同的真假值。因而二者相同。

方法二：为证明：如果 p∧q，则 q∧p。不妨证明

$$\frac{p \wedge q}{q \wedge p}$$

为此，我们展示了一棵证明树。树的叶子是前提，根是结论。

$$\frac{\dfrac{p \wedge q}{q}[\wedge -2] \quad \dfrac{p \wedge q}{p}[\wedge -1]}{q \wedge p}[\wedge +]$$

这棵树中，用了三条规则。一条规则的结论，形成另一条规则的前提，而且正好匹配。此树匹配的规则正好是我们想证明的，因为此树的叶子是两个相同的命题 p∧q，正是要证明的规则的前提；树的根对应于要证明的规则的结论。证毕。

这里，我们还要引进两个术语—假定与解除规则。在证明中引入的某些前提叫做假定。在证明中，假定必须被解除，做这件事的规则叫做解除规则。假定 p 由⌜p⌝[] 表示。下一节将会看到假定及解除规则的例子。

2.3 析 取

析取 p∨q 为真的充要条件是 p 为真或者 q 为真，其真值表如下：

p	q	p∨q
T	T	T
T	F	T
F	T	T
F	F	F

∨是“同或”运算符：两个析取量之一为真时，析取或为真，包括两个都为真在内。对于析取，有三条推理规则：

$$\frac{p}{p \vee q}[\vee +1]$$

$$\frac{q}{p \vee q}[\vee +2]$$

试读结束：需要全本请在线购买： www.ertongbook.com