“十二五”江苏省高等学校重点教材

21世纪高等学校计算机专业实用规划教材

形式化方法导论

张广泉 编著

清华大学出版社

“十二五”江苏省高等学校重点教材（编号：2014-2-045）

21世纪高等学校计算机专业实用规划教材

形式化方法导论

张广泉 编著

清华大学出版社
北京

内 容 简 介

形式化方法是指有严格数学基础的软件和系统开发方法，支持软件与系统的规约、设计、验证与演化等活动。随着软件可信需求的不断增长，形式化方法的重要性和关注度日益提高。

全书共12章，第1章概述形式化方法，第2章介绍形式化方法发展早期的经典内容，其余部分共分3篇：上篇（第3～5章）为系统建模篇，着重介绍迁移系统、有穷自动机、Petri网等基本计算模型；中篇（第6和第7章）为形式规约篇，着重讨论时序逻辑及其在并发系统属性描述的应用；下篇（第8～12章）为形式验证篇，除介绍演绎证明方法外，着重介绍验证并发、实时及混成系统的各种模型检测方法及相关验证工具。全书提供了大量应用实例，每章后均附有习题。

本书适合作为高等院校计算机、软件工程、网络工程、信息安全、自动化等专业高年级本科生、研究生的教材，同时可供相关领域的研究人员和技术开发人员参考。

图书在版编目(CIP)数据

形式化方法导论/张广泉编著. --北京：清华大学出版社，2015
21世纪高等学校计算机专业实用规划教材
ISBN 978-7-302-41161-1

Ⅰ. ①形…　Ⅱ. ①张…　Ⅲ. ①形式语言　Ⅳ. ①TP301.2

中国版本图书馆CIP数据核字(2015)第184734号

责任编辑：黄　芝　薛　阳
封面设计：何凤霞
责任校对：李建庄
责任印制：刘海龙

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦A座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课 件 下 载：http://www.tup.com.cn，010-62795954
印 装 者：清华大学印刷厂
经　　销：全国新华书店
开　　本：185mm×260mm　**印　张**：16.75　**彩　插**：1　**字　数**：421千字
版　　次：2015年12月第1版　　**印　次**：2015年12月第1次印刷
印　　数：1～2000
定　　价：39.00元

产品编号：064497-01

出版说明

随着我国改革开放的进一步深化，高等教育也得到了快速发展，各地高校紧密结合地方经济建设发展需要，科学运用市场调节机制，加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度，通过教育改革合理调整和配置了教育资源，优化了传统学科专业，积极为地方经济建设输送人才，为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是，高等教育质量还需要进一步提高以适应经济社会发展的需要，不少高校的专业设置和结构不尽合理，教师队伍整体素质亟待提高，人才培养模式、教学内容和方法需要进一步转变，学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007 年 1 月，教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》，计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”，通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容，进一步深化高等学校教学改革，提高人才培养的能力和水平，更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中，各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势，对其特色专业及特色课程(群)加以规划、整理和总结，更新教学内容、改革课程体系，建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上，经教育部相关教学指导委员会专家的指导和建议，清华大学出版社在多个领域精选各高校的特色课程，分别规划出版系列教材，以配合“质量工程”的实施，满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机专业课程领域，以专业基础课为主、专业课为辅，横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 反映计算机学科的最新发展，总结近年来计算机专业教学的最新成果。内容先进，充分吸收国外先进成果和理念。

(2) 反映教学需要，促进教学发展。教材要适应多样化的教学需要，正确把握教学内容和课程体系的改革方向，融合先进的教学思想、方法和手段，体现科学性、先进性和系统性，强调对学生实践能力的培养，为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略，突出重点，保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上；特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版，逐步形成精品教材；提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本，合理配套。专业基础课和专业课教材配套，同一门课程有针对不同层次、面向不同应用的多本具有各自内容特点的教材。处理好教材统一性与多样化，基本教材与辅助教材、教学参考书，文字教材与软件教材的关系，实现教材系列资源配套。

(5) 依靠专家，择优选用。在制定教材规划时要依靠各课程专家在调查研究本课程教

材建设现状的基础上提出规划选题。在落实主编人选时，要引入竞争机制，通过申报、评审确定主题。书稿完成后要认真实行审稿程序，确保出书质量。

繁荣教材出版事业，提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度，希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21世纪高等学校计算机专业实用规划教材

联系人：魏江江 weijj@tup.tsinghua.edu.cn

前　言

软件产业是信息产业的核心，是国家信息化的基础和支撑。软件是工业 4.0 和中国制造 2025 的使能和驱动。为推进产业结构优化升级，加快培养软件人才的步伐，近年来教育部大力发展高校软件工程教育，软件工程已从最初的计算机科学与技术的一个学科方向调整为包括软件工程理论与方法、软件工程技术、软件服务工程和领域软件工程等学科方向的、独立的一级学科。软件工程理论与方法是软件工程一级学科的基础，作为其核心内容之一，形式化方法是指有严格数学基础的软件和系统开发方法，支持软件与系统的规约、设计、验证与演化等活动。随着软件可信需求的不断增长，形式化方法的重要性和关注度日益提高。

形式化方法相关的教学工作已经得到欧美国家高等学校的重视和推广，知识体系和课程教学内容日趋完善。而目前国内高校关于形式化方法教育还相对薄弱，主要因素之一是缺乏比较全面、系统介绍形式化理论、方法和应用的教材。

本书是在学习、总结形式化方法领域国内外相关文献的基础上，结合作者多年从事形式化方法教学和科研的实践撰写而成的，本书具有以下几个特点：

(1) 通过详细分析和梳理，提炼出形式化方法核心、本质的原理、方法和技术，其中自动机和时序逻辑是贯穿全书内容的两大重要基础。

(2) 重点阐述以模型检测为主要内容的形式化验证方法，使学生在有限的学时范围内，能有效地掌握形式化方法自动化部分的核心内容。

(3) 注重实践与应用，详细介绍 SPIN、UPPAAL 和 PRISM 等典型的形式化验证工具的使用方法，结合实例分析，达到理论学习与实际应用的有机结合。

全书共 12 章。其中第 1 章概述形式化方法的发展历程和基本内容，第 2 章介绍形式化方法发展早期的经典内容，即串行程序的正确性证明。其余部分针对并发系统，分为上、中、下三篇阐述形式化建模、规约和验证方法。其中：

上篇(第 3～5 章)为系统建模篇，主要介绍三个典型的并发系统计算模型。第 3 章介绍基于状态迁移的计算模型——迁移系统，第 4 章介绍描述有穷状态系统的计算模型——有穷自动机，它也是计算机科学中最基本的数学模型，第 5 章介绍最早的并发计算模型——Petri 网。

中篇(第 6 和第 7 章)为形式规约篇，着重讨论并发系统属性的主要规约方法及应用。第 6 章介绍真假值依赖时间而变化的非经典逻辑——时序逻辑，它是描述并发系统属性的重要工具，第 7 章重点阐述并发系统最基本的两类属性——安全性和活性，及其时序逻辑描述方法。

下篇(第 8～12 章)为形式验证篇，着重介绍主要的形式验证方法及相关验证工具。第

8章介绍基于时序逻辑的演绎证明方法及验证工具STeP，第9～12章重点阐述模型检测方法、工具及其在并发、实时及混成系统中的应用，这是形式化方法自动化的核心内容，也是本书的重点。其中第9章介绍经典的模型检测算法、验证工具SPIN及应用，第10章介绍基于二叉决策图(BDD)的符号模型检测方法、验证工具SMV及应用，第11章介绍模型检测与概率分析方法相结合的概率模型检测方法、验证工具PRISM及应用，第12章介绍实时与混成系统的模型检测方法、验证工具UPPAAL及应用。

全书提供了大量应用实例，每章后均附有习题。

本书由张广泉担任主编，负责全书内容的组稿、统稿和修改工作。顾玉磊、宋相君、宋振华、项周坤、沈兴勤、郑林峰、张红美等参与了书稿整理、文字录入和校对工作，祝义副教授、孙庆英老师、魏慧老师等参与了部分书稿的校对工作，在此对他们的辛勤劳动表示感谢。此外，在本书的编写过程中，参考了大量国内外相关文献，在此对本书所引用文献的作者深表感谢。

本书编写工作得到江苏省“十二五”高等学校重点教材立项建设和苏州大学教材培育项目的资助，以及江苏省自然科学基金(BK2011281)、中国科学院软件研究所计算机科学国家重点实验室开放课题(SYSKF0908、SYSKF1201)、南京大学计算机软件新技术国家重点实验室开放课题(KFKT2012B15)的支持和帮助。中国科学院软件研究所焦莉研究员、李广元副研究员、朱雪阳博士、晏荣杰博士仔细阅读了本书初稿，提出了许多重要的修改意见与建议，在此表示衷心感谢。本书的编写还得到中国科学院软件研究所周巢尘院士、林惠民院士、沈一栋研究员、张健研究员、张文辉研究员、詹乃军研究员、南京大学李宣东教授、上海大学缪淮扣教授、南京航空航天大学黄志球教授、东南大学李必信教授等及苏州大学教务部和计算机科学与技术学院的关心和支持，清华大学出版社编辑黄芝和薛阳为本书出版做了大量工作，在此一并表示诚挚的感谢。

由于编者水平有限，书中难免有不当之处，敬请读者批评指正。如有问题，请发邮件至gqzhang@suda.edu.cn。

张广泉

2015年10月于苏州大学天赐庄校区

目　录

下篇 形式验证

第1章 绪 论

本章学习目标

(1) 掌握形式化方法的基本概念。

(2) 了解形式化方法的发展历程。

(3) 了解形式化方法的基本内容。

软件是否可信已成为一个国家的经济与国防等系统能否正常运转的关键因素之一，在一些诸如核反应堆控制、航空航天以及铁路调度等重要领域更是如此。这类系统要求绝对安全可靠，不容半点疏漏，否则将导致灾难性后果。例如，1996 年 6 月 4 日，耗资 80 亿美元的欧洲航天局阿丽亚娜 501 火箭发射升空 37 秒后爆炸。原因是主发动机打火 37 秒后制导和姿态信息完全遗失，而信息遗失是由于惯性制导系统的软件出现规约和设计错误造成的。又如，2000 年 10 月，诺基亚软件中的一个错误造成德国一家移动电话公司的通信服务被中断三个多小时。类似的报道屡见不鲜，这样的问题如果发生在战争环境下，后果不堪设想。因此，如何保障这些系统的安全性和可靠性成为计算机科学与控制论领域共同关注的一个焦点问题。

软件可靠性主要取决于两方面，一是软件开发的方法与过程；二是软件产品的测试与验证。在目前大多数的工程实践中，软件产品的设计与开发仍缺乏坚实的科学基础和成熟的方法学，软件产品质量主要还是通过测试和模拟等方法来保障的。由于测试用例的覆盖率难以达到百分之百，加之系统的运行通常与外部环境有关(如反应式并发系统)，其执行往往具有不确定性，测试极为困难。测试代价很大，而且无法保证发现所有潜在的错误。为了从根本上保证软件系统的可靠安全，包括图灵奖得主 A. Pnueli 在内的许多计算机科学家都认为，采用形式化方法(Formal Methods)对系统进行验证和分析，是构造安全可信软件的一个重要途径。

1.1 形式化方法的发展历程

形式化方法是指有严格数学基础的软件和系统开发方法，支持计算机系统及软件的规约、设计、验证与演化等活动。随着高可信软件的兴起，形式化方法作为构造相关软件的重要途径，关注度日益提高。

形式化方法最早可追溯到 20 世纪 50 年代后期关于程序设计语言编译技术的研究。当时 J. Backus 提出 BNF 描述 ALGOL60 语言的语法，涌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是从20世纪60年代后期开始的，针对当时所谓的“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，以期能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。

由于传统的测试方法只能发现程序的错误，而不能证明程序没有错误，早期的形式化方法(20世纪70年代前后)主要研究如何使用数学(逻辑)方法，进行(串行)程序正确性证明(第2章)，比较著名的证明方法有Floyd前后断言法、Hoare公理化方法和Dijkstra最弱前置条件方法等。在上述方法的基础上，许多计算机科学家相继提出了不同的串行程序正确性证明方法。如Manna的子目标断言法、不动点方法和计算归纳法等，Ashcroft方法和Burstall间歇断言方法等；这一时期的程序正确性证明方法主要是以逻辑推理为基础的演绎证明方法。

20世纪70年代中期开始，并发程序设计逐渐成为程序理论的主要研究课题之一。不同任务之间的同步、信息交换及死锁等是并发程序不同于串行程序的主要动态特性。1976年，S. Owicki和D. Gries对Floyd-Hoare方法进行扩充，使之含有并发性的推理规则。在此前后，人们认识到经典一阶逻辑和Hoare逻辑等在描述和验证并发程序方面的不足，模态逻辑(Modal Logic)和时序逻辑(Temporal Logic)逐渐被引入并发程序领域。1974年，R. Burstall首先建议使用模态逻辑进行程序推理。1977年，以色列科学家A. Pnueli首次提出了用时序逻辑对并发程序性质进行形式描述并验证的思想，开创了并发程序验证的新途径，并于1996年获得图灵奖。1982年，Z. Manna和A. Pnueli提出了基于时序逻辑的并发程序演绎证明方法并研制了相关证明工具STeP。该方法通过在程序中插入时序断言，采用与前后断言法类似的方法来证明一个并发程序满足某时序逻辑公式。

演绎验证方法的缺点是不能够做到完全自动化，还需与用户交互，要求用户提供验证中创造性最强部分(建立断言等)的工作。演绎证明方法需要大量的时间和专门的技术，因而效率较低，只能通过很小的例子进行展示。

20世纪80年代以来，随着超大规模集成电路技术的日趋成熟，并行和分布式系统得到迅速发展，鉴于演绎验证的局限性，自动化验证技术开始引起了人们的关注。图灵奖获得者E. M. Clarke、E. A. Emerson和J. Sifakis等人于1981年提出的模型检测(Model Checking)方法是一种基于状态空间搜索算法的自动验证方法，最初的模型检测算法是采用分支时序逻辑(CTL)来描述系统的规约，故称为CTL模型检测。稍后，又出现了线性时序逻辑(LTL)模型检测。但是，LTL模型检测算法时间一般是PSPACE完全的，而CTL模型检测算法时间一般是线性或多项式的。模型检测在硬件设计和通信协议的形式验证上取得了巨大成功，著名的模型检测工具有SMV和SPIN等，其中G. Holzmann研发的SPIN获2001年ACM软件系统奖。

模型检测应用面临的主要问题是状态空间爆炸问题。由于系统的有穷状态模型的状态数量往往随其模型的并发分量的增加呈指数增长，对复杂系统建模时，其可达的状态空间常常难以在计算机存储器中全部构建，也就无法进行模型检测了。20世纪80年代后期以来研究人员提出了许多缓解状态空间爆炸问题的方法，这些方法大致可分为基于简化全局状态空间和基于检测局部状态空间两大类，主要有符号模型检测、有界模型检验、对称模型检

测、偏序模型检测、On-the-fly 模型检测以及抽象与组合方法等。此外，为了分析系统的概率属性，近年来还提出了概率模型检测技术及相关的分析与验证工具（如 PRISM 等）。

20 世纪 90 年代以来，随着实时系统（Real-time System）与混成系统（Hybrid System）在工业及国防等领域的应用越来越广泛，如何保障这些系统的可靠性成为人们关心的焦点。其中，实时系统模型检测的研究已经取得重大进展，主要表现在三个方面：

① 出现了用于表示实时系统的各种数学模型，如时间 Petri 网、时间自动机以及各种进程代数的时间扩充。其中时间自动机的影响和应用最为广泛。

② 提出了能描述实时系统的模态/时序逻辑。

③ 针对这些实时系统的数学模型和逻辑，设计了各种模型检测的算法，并实现了相应的分析与验证工具，如 UPPAAL、KRONOS 和 HyTech 等。

进入 21 世纪以来，随着云计算、物联网、大数据的兴起，如何保障动态、开放、多变的网络环境下软件的可信性引起了国内外学术界和政府部门越来越多的关注。“高可信软件”先后被美国、欧盟列为优先资助的研究方向，国内也开展了可信软件国家重大基础研究计划，从而进一步推动了形式化方法的研究和应用。

经过几十年的研究和应用，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，到现在应用于不同领域、不同阶段的基于逻辑、自动机、网络、进程代数、代数等众多形式化方法。形式化方法能在系统开发早期发现系统中的不一致、歧义、不完全和错误，已被证明是一种行之有效的减少设计错误、提高软件系统可信性的重要途径。目前的发展趋势是将其逐渐融入软件开发过程的各个阶段。

1.2 形式化方法的基本内容

形式化方法的主要内容包括：

① 系统建模（System Modeling）。通过构造系统 S 的模型 M 来描述系统及其行为模式。

② 形式规约（Formal Specification）。通过定义系统 S 必须满足的一些属性 φ，如安全性、活性等来描述系统约束。

③ 形式验证（Formal Verification）。证明描述系统 S 行为的模型 M 确实满足系统的形式规约 φ（即验证 $M \models \varphi$）。

1.2.1 系统建模

计算机系统可分为串行和并发两大类，分别有不同的建模方法。串行系统（也称顺序系统）是一种较常规形式的系统，其在计算终止后产生一个最终结果。因而，通常将串行系统看作从初始状态到终止状态（或终止结果）的一个函数，因此，一个串行系统可以通过其输入输出关系来描述，即连接可能的初始状态到可能的计算结果之间的条件，这类系统可通过 Z、VDM、B 等方法进行建模[①]。

计算机界的并发现象始于 20 世纪 60 年代，其中并发的概念由 C. A. Petri 于 1962 年首

① 限于篇幅，本书不介绍串行系统建模和进程代数相关的内容，有兴趣的读者可参阅有关文献。

先提出。若一个系统内部发生的两个事件之间没有因果关系或可以按任意次序发生，则称这两个事件是并发的，存在并发事件的系统称为并发系统。如操作系统就是一个典型的并发系统，人类社会也可看作一个并发系统。较串行系统而言，并发系统要复杂得多，这是由于并发执行的程序在执行过程中各程序交替点的不确定性，引起了各程序走停点及交替过程的不确定性。这使它丧失了串行程序的全部特征：顺序性、封闭性、可再现性；也带来了新的特性：不确定性和并发性。

人们对并发系统的这种新特性缺乏认识和理解，常常产生困惑甚至混乱。如何为实际的并发系统设计和分析提供坚实的理论基础、提高其可信性，是今后几十年计算机科学和软件工程面临的重大挑战。笔者认为首先需要建立能够描述并发系统行为的计算模型，即并发模型。一方面，为了适合验证需要，模型应能捕捉到系统与正确性有关的行为特征；另一方面，为了简化和缩减被检测的系统，不使验证过于复杂，模型抽象时应该去掉一些不影响验证属性正确性的细节。例如，为数字电路建模时，通常按照门和布尔值进行推理，而不是实际的电压层；同样，在分析通信协议时，集中于消息的交换，而忽略实际的消息内容。合理抽象后的计算模型，可以帮助人们深入认识并发系统的本质特性，并为并发系统的形式规约和验证打下基础。

迄今为止，并发系统的计算模型已有许多种，如迁移系统、自动机、Petri 网以及基于进程代数的 CSP、CCS 等[①]。

迁移系统（第 3 章）是一种基于状态迁移的基本计算模型，即通过系统（程序）的状态集合以及对应状态间变迁的迁移（也称操作）集合刻画系统的行为模型。基于迁移系统，人们提出多种并发模型，如进程代数 CSP 模型、CCS 模型等；通过对基本迁移系统进行扩展，如将公平性引入迁移系统，可得到公平迁移系统；将时间约束引入迁移系统，可得到时间迁移系统等。

自动机（第 4 章）是计算机科学中最基本的一类抽象计算模型。其中有穷自动机是自动机理论的基础，它是一种描述有穷状态系统的抽象数学模型，许多并发模型都是在有穷自动机的基础上建立的。

需要说明的是，迁移系统和有穷自动机本质上都是交错并发模型，即系统进程间的并发执行并不是真并发，而是通过各个原子迁移以不确定的顺序交错执行来表示的，其计算行为表现为状态迁移序列。

Petri 网（第 5 章）是最早的并发计算模型，它是一种系统的既有数学分析又有图形描述的工具，既可以通过直观的图形刻画系统的结构，又可以引入数学方法对其性质进行分析，特别适合描述系统中进程间的顺序、并发、互斥、冲突及同步等关系。与迁移系统、自动机不同的是，Petri 网所描述的并发是“真并发”，在 Petri 网中系统不存在统一的时钟，除因果关系外没有其他信息可以用来判定两个事件的依赖关系。

1.2.2 形式规约

软件开发首先需要确定“做什么（what to do）”，而非“怎么做（how to do）”，这个阶段称为软件需求。需求规约就是以一种清晰、简明、一致且无歧义的方式，刻画客户或用户所需

① 限于篇幅，本书不介绍串行系统建模和进程代数相关的内容，有兴趣的读者可参阅有关文献。

系统中所有重要方面的一组陈述，这是软件开发最重要和最困难的阶段。一方面，规约是客户或用户与软件开发人员之间的接口界面，可看作他们之间的一种契约合同；另一方面，规约也是设计和编制程序的出发点和验证程序是否正确的依据。事实上，判断最终所开发出的系统（程序）是否正确，就是通过验证它是否满足其需求规约来进行的。

按形式化的程度，需求规约的描述可采用非形式化、半形式化和形式化三类方式。非形式化方式是指采用自然语言描述系统需求。尽管它易为用户理解，但可能存在矛盾、二义性、含糊性、不完整陈述以及抽象层次的混杂等问题，因而常导致需求描述错误，从而引起用户或客户对交付的系统不满意。此外非形式化方式也难以提供自动化支持。形式规约（也称形式规范或形式化描述）是用具有精确语义的形式化语言描述系统需求（性质），对形式规约通常要讨论其一致性（自身无矛盾）和完备性（是否完全、无遗漏地刻画所要描述的对象）等性质。

与系统建模类似，形式规约也可以分为面向串行程序的与面向并发程序的。对于串行程序而言，可看作从初始状态到终止状态的一个函数，这种初始状态和终止状态之间的关系特性是静态的而非动态的，可采用前后断言法、Hoare 逻辑等一阶逻辑以及代数方法（OBJ、Clear、ASL、ACT-One/Two…）对其进行规约[①]。

由于并发程序比串行程序表现出更为复杂的行为，其状态随着时间的推移不断改变，且可能不断影响外部环境，这种持续不终止的动态行为是经典一阶逻辑和 Hoare 逻辑所不能描述的。时序逻辑（第 6 章）是关于随着时间变化而不断改变其值的动态变元（也称时序变元）的一种模态逻辑。它除了含有经典逻辑的逻辑联结词和量词外，还包含一些时序算子。时序逻辑具有很强的表达能力，一些重要的并发系统属性（第 7 章），如安全性（指“坏的”事件永远不会发生，如部分正确性、互斥性、无死锁性等）、活性（指“好的”事件终将发生，如终止性、完全正确性、响应性等）都可以用时序逻辑公式表达。时序逻辑作为研究并发程序尤其是持续不终止的反应式程序（如操作系统、网络通信协议等）的强有力的形式化工具，目前已被广泛应用于并发系统（包括实时及混成系统）的规约和验证。

1.2.3 形式验证

形式验证与形式规约之间具有紧密的联系，形式验证就是验证已有的软件（系统）S 是否满足其规约 ϕ（即 $S \models \phi$），它也是形式化方法所要解决的核心问题。

形式验证方法主要分为两类：一类是以逻辑推理为基础的演绎验证；另一类是以穷尽搜索为基础的模型检测。

演绎证明用逻辑公式描述系统及其性质，通过一些公理或推理规则来证明系统具有某些性质。逻辑推理方法主要有自然推演、归结、Hoare 逻辑以及时序演算等。该方法也可分为面向串行程序的正确性证明（第 2 章）与面向并发程序的演绎证明（第 8 章）。常见的演绎证明工具有斯坦福大学的演绎验证工具 STeP、机器定理证明器或检验器如 ACL2、HOL、PVS、TLV、Coq 等；演绎验证的优点是既可以验证有穷状态系统，也可以使用归纳的方法来处理无限状态的问题，并且证明的中间步骤使用户对系统和被证明的性质有更多的了解。这类方法的不足之处是不能做到完全自动化验证，对于稍微复杂的系统，自动化的推理就难

① 限于篇幅，本书不介绍代数规约相关的内容，有兴趣的读者可参阅有关文献。