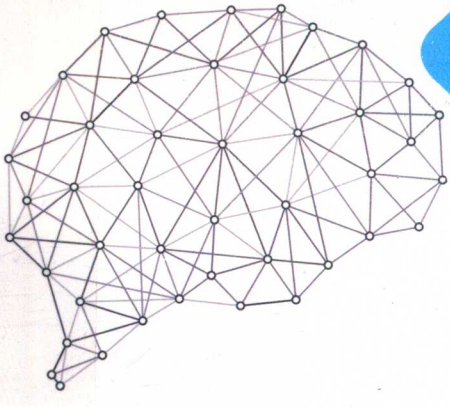
江西理工大学优秀博士论文文库

Büchi自动机模型检测及其安全性分析应用研究

王曦 欧阳城添◎著

Büchi Zidongji Moxing Jiance
Jiqi Anquanxing Fenxi
Yingyong Yanjiu



中南大学出版社
www.csupress.com.cn

江西理工大学优秀博士论文文库

Büchi 自动机模型检测及其安全性分析应用研究

王　曦　欧阳城添　著

中南大学出版社
www.csupress.com.cn
·长　沙·

图书在版编目（CIP）数据

Büchi 自动机模型检测及其安全性分析应用研究 / 王曦，欧阳城添著. --长沙：中南大学出版社，2019.2

ISBN 978-7-5487-3563-2

Ⅰ.①B… Ⅱ.①王… ②欧… Ⅲ.①自动检测系统—研究 Ⅳ.①TP274

中国版本图书馆 CIP 数据核字(2019)第 032136 号

Büchi 自动机模型检测及其安全性分析应用研究

王　曦　欧阳城添　著

□责任编辑　刘小沛
□责任印制　易红卫
□出版发行　中南大学出版社
社址：长沙市麓山南路　　邮编：410083
发行科电话：0731-88876770　　传真：0731-88710482
□印　　装　长沙鸿和印务有限公司

□开　　本　710×1000　1/16　□印张 10　□字数 199 千字
□版　　次　2019 年 2 月第 1 版　□2019 年 2 月第 1 次印刷
□书　　号　ISBN 978-7-5487-3563-2
□定　　价　40.00 元

图书出现印装问题，请与经销商调换

内容简介

在航空航天、交通运输、核电能源和医疗卫生等安全苛求领域，系统的安全性尤为重要，为了确保系统安全，防止灾难性事故的发生，科技人员提出了诸多理论和方法，其中模型检测以其简洁明了和自动化程度高的优点而引人注目，其算法和应用研究成了学术界和工业界研究的热点话题之一。本书在讲解模型检测基础理论与基本方法的基础上，主要介绍笔者以广义 Büchi 自动机为研究对象，在模型检测算法及其安全性分析应用研究方面所取得的独创性研究成果，主要包括基于启发式 NDFS 的模型检测算法、基于启发式 SCCs 的广义 Büchi 自动机判空检测算法、基于启发式 on-the-fly 的扩展 TGBA 模型检测算法、基于场景分析的系统形式化模型生成方法、基于模型检测的系统安全性验证方法、基于故障注入的模型检测分析、铁路车站联锁系统的安全性分析研究。

本书可以作为计算机软件与理论、计算机应用、软件工程、自动化控制、信息安全、网络空间安全等专业类研究生课程教材，也可以作为相关领域科技人员的参考用书。

前 言

随着计算机技术的日益成熟，以及硬件成本的迅速降低，各种结构复杂、功能强大的嵌入式系统被广泛应用到航空航天、交通运输、核电能源和医疗卫生等领域中，并发挥着不可替代的作用，但是，一旦这些系统失效，将给人类的生命、财产和环境造成重大甚至是灾难性的损失。因此，这类嵌入式系统通常称为安全苛求计算机系统(safety-critical computer system)，简称为安全苛求系统(safety-critical system)。这类系统不断地向纵深发展，系统的控制逻辑与功能实现日趋复杂，为了进一步提高系统的性能并降低使用和维护成本，许多原来由硬件实现的逻辑功能日益通过计算机软件来替代，这为大型复杂安全苛求系统的设计与开发提供了有效的实现途径。但由于软件自身的灵活性使得其存在很多不可判定的因素，其广泛应用于安全苛求系统的同时，也给系统的安全性带来了危害与隐患。

在对安全性有着最高要求的安全苛求系统中，软件的安全性尤为重要。因此，人们常把运行在安全苛求系统上的软件称为安全苛求软件(safety-critical software)。区别于普通软件，安全苛求软件以安全性作为第一性能特征，其失效将可能导致整个安全苛求系统的失效，进而给人类的生命财产和自然环境造成灾难性的损失。

为了确保安全苛求系统软件的质量，尤其是安全性，防止灾难性事故的发生，科技人员提出了诸多理论和方法，其中模型检测(model checking)以其简洁明了和自动化程度高而引人注目，其算法和应用研究成了学术界和工业界研究的热点话题之一。模型检测在以往的安全性分析及其应用研究中，还存在着以下几个方面的问题：

(1)由于模型检测基于状态搜索的基本思想，搜索的可穷尽性要求系统模型状态数有穷，而在实际的大型复杂软件的安全性分析与验证中，系统中的状态数目可能很大，以至无法在有限的时间和内存空间条件下进行完整的检验，使得模型检测面临着“状态空间爆炸(state space explosion)”的严重问题。因此如何有效缓解“状态爆炸”是模型检测技术能被广泛使用的一个重要前提。

(2)模型检测在以往的安全性分析中，主要关注的是系统的形式化建模，建模的准确性将直接影响到安全性分析的结果，因此，通常要求安全分析人员掌握娴熟的形式化技术，应用该类方法的难度比较大。

(3)对系统模型作形式化描述时主要根据系统设计的功能行为对系统建模，难以充分考虑安全苛求系统中复杂多变的实际运营场景对系统的安全性影响。

(4)通过故障注入技术来分析安全苛求系统的功能与设计在故障情况下是否能导向安全时，由于多故障注入技术比较复杂，因此通常关注的是单故障注入，对系统中存在的多故障情形考虑较少。

作者主要针对上述问题进行研究，也对基于 Büchi 自动机的模型检测理论与方法，以及其在安全性分析方面的应用研究做比较系统的介绍，书中大部分内容都是笔者近年来的研究成果。本书的主要研究工作和创新点如下：

(1)对广义 Büchi 自动机和标准 Büchi 自动机的模型结构、模型检测的基本原理和 on-the-fly 模型检测方法进行研究。on-the-fly 模型检测方法是一种边搜索边检测的算法，能较好地提高模型检测的时空效率，此算法在搜索系统状态空间的同时，能对系统模型是否满足属性要求进行验证，若在搜索过程中，找到了系统的一个可接受运行序列，便可实时终止系统状态空间的搜索，得出模型是否满足属性的结论。当前的 on-the-fly 判空检测算法主要用于 Büchi 自动机的判空检测，本书结合 on-the-fly 判空检测算法对 Büchi 自动机模型做出非空性判断的研究机理，研究嵌套的深度优先搜索 NDFS(nested depth-first searches，以下简称 NDFS)算法和基于强连通子图 SCCs(strongly connected components，以下简称 SCCs)的搜索算法，分析这两类算法在计算方法上的差异，为模型检测算法的进一步研究打基础。

(2)在研究 on-the-fly 判空检测算法和嵌套的深度优先搜索 NDFSs 算法的基础上，以带有多个可接受条件的广义 Büchi 自动机为研究对象，提出基于启发式 NDFS 的模型检测算法。该算法结合 on-the-fly 算法与启发式 NDFS 算法，能较快地判断出广义 Büchi 自动机的非空性，通过理论证明和实验验证了算法的正确性和实际可行性。与已有算法相比，在广义 Büchi 自动机非空的情况下，该算法能够有效减少系统状态空间的搜索，提高了检测效率，并且能够形成相应反例，为缓解形式化验证中存在的状态空间爆炸问题提供了较好的解决途径。

(3)在已提出的基于启发式 NDFS 的模型检测算法的基础上，进一步提出基于启发式 SCCs 的广义 Büchi 自动机判空检测算法，简称 HSCCsEC 算法，该算法以基于迁移的广义 Büchi 自动机为研究对象，在 on-the-fly 算法的基础上结合启发式深度优先搜索和 SCCs 判空检测算法，能较快地判断出广义 Büchi 自动机的非空性。通过实验对比与分析研究，发现在通常情况下，HSCCsEC 算法所需要的检测时间和内存空间比已有的相关算法均更少，判空检测的有效性进一步提高，因

而 HSCCsEC 算法具有明显的优势。

(4)对安全苛求系统造成安全隐患或危险的因素通常是 k 个故障树最小割集，当系统运行中存在某个故障树最小割集的全部底事件时，若不能导向安全侧，则该系统被认为是一个不安全的系统。采用模型检测方法验证安全苛求软件的安全性时，只需要在软件的形式化模型与系统属性模型的同步积自动机中，找到包含有构成某个故障树最小割集的全部底事件的一条运行路径时，便可作出系统模型违反了系统安全属性的判断。由于现有的广义 Büchi 自动机判空检测算法需要判断同步积自动机的运行序列是否满足其所有的可接受条件子集才能作出非空性判断，因此，现有的广义 Büchi 自动机判空检测算法不适于处理实际安全苛求系统的安全性分析与验证。

鉴于此，作者在提出基于启发式 NDFS 的模型检测算法和基于启发式 SCCs 的广义 Büchi 自动机判空检测算法的基础上，对基于迁移的广义 Büchi 自动机模型进行扩展，提出基于启发式 on-the-fly 的扩展 TGBA 模型检测算法(heuristic on-the-fly model checking algorithm for extended TGBA，简称 MCA_ETGBA 算法)。MCA_ETGBA 算法以 ETGBA(extended transition-based generalized Büchi automaton，简称 ETGBA)模型为研究对象，通过启发式 on-the-fly 判空检测算法对 ETGBA 作判空检测时，加强了对不能构成其可判定运行序列的结点的处理，节省了内存空间，提高了检测效率，从而能较快地作出非空性判断。通过算法的正确性证明、复杂性分析和应用实例验证了所提出算法的正确性和可行性。与已有算法相比，MCA_ETGBA 算法的通用性更强，判空检测的有效性更优，在通常情况下，其时空性能均较优，为安全苛求软件的安全性验证提供了切实可行的解决途径。

(5)采用形式化方法对系统的安全性进行分析与验证，是构造可靠安全软件系统的一个重要途径。当前的形式化安全性分析方法，面临着系统的形式化建模难问题。为了解决形式化验证中存在的形式化建模难的问题，并考虑到复杂多变的实际运营场景，提出基于场景分析的系统形式化模型生成方法。该方法首先采用 OCL(object constraint language，以下简称 OCL)前/后置条件分析法对 UML 时序场景作一致性分析，然后将 UML 时序图中对象交互的行为序列转换成 FSP(finite state process，以下简称 FSP)进程代数模型，进而得到系统的形式化模型。该方法为系统的形式化建模提供了新思路，从安全质量方面改善了安全苛求软件的设计与开发，丰富了基于模型的软件形式化开发方法。

(6)为了确保安全苛求系统的充分安全性，不仅需要研究系统在通常情况下的功能行为是否满足系统的安全性要求，还需要研究系统在带有单个故障和多个故障情形下的安全防护及故障－安全方面的处理能力。鉴于此，本书提出基于故障注入的模型检测分析方法，该方法能对安全苛求软件系统的形式化模型分别作单故障注入及多故障注入，并对带有各类故障模式的相应形式化扩展模型作安全

性验证与结果分析，得到违反系统安全性要求的单故障及多故障事件构成的故障树最小割集，并进一步得到用计算树逻辑表示的系统形式化安全需求。解决了系统安全性分析中的多故障注入问题及形式化安全需求的生成问题，使软件的安全性分析更全面，进一步完善了系统的安全性分析质量。

(7)铁路车站联锁系统是安全完善度等级为 SIL4 的安全系统，是典型的安全苛求系统，为了验证本书所提算法的实际可行性，提出系统的安全性分析流程，将本书的相关研究成果应用到铁路车站联锁系统的安全性分析中，并将安全性分析结果与相关铁路行业标准进行比较分析。分析结果与铁道行业标准 TB/T 3027—2002 计算机联锁技术条件中规定的相关技术条件的要求相一致，表明本书提出的安全性分析流程及模型检测分析方法是正确可行的。

本书的研究成果具有一定的理论价值和实践意义，可为安全苛求软件的开发、安全性分析与验证、安全评估提供有力的理论指导和技术支持，对我国高速铁路、国防武器装备、航空航天等安全苛求领域的系统安全性保障具有重要的现实意义。

本书的写作，既是笔者多年来辛勤学习的结果，也是各位老师、同学及家人帮助的结果。首先向我的博士导师徐中伟教授致以最诚挚的谢意和崇高的敬意。徐老师在安全苛求领域渊博的学识与独特的见解、在科学研究过程中严谨的治学态度、精益求精的工作作风和诲人不倦的为师风范，让我受益匪浅。徐老师对学术前沿和领域应用的敏锐把握给予了我无尽的启迪，引领我步入了安全苛求领域的研究前沿。在同济大学读博期间，徐老师为我创造了优越的科研和学习环境，从课题的理论研究到工程的实践过程，都给予了我充分的信任与个人发展空间，对于研究过程中遇到的难题，他总是不厌其烦地与我进行深入的探讨，给予我悉心的指导和帮助，我取得的每一点成绩都凝聚了徐老师的心血！师恩永难忘，在此对徐老师表示衷心的感谢！

感谢我的硕士导师张小红教授，她敏锐的思维，严谨的治学态度，谦逊随和、乐观豁达、宽厚待人的高尚品格，无不值得我学习和效仿。多年来，张老师一直在各方面都给予我无微不至的热情鼓励和深情关怀，师恩深重，终生难忘，在此对张老师表示衷心的感谢！

感谢恩师何光优老师、姚祖喜老师和吴爱桃老师对我一直以来的关心、支持、鼓励和帮助。感谢陈邦兴教授在安全标准、安全评估等方面给予我的指点。感谢蔡维娜老师多年来对我在攻读博士期间学习、工作和生活方面的照顾、支持和帮助。感谢梅萌老师、杜军威博士、喻钢博士、陈祖希博士、祝玉军博士、万勇兵博士、郑剑博士、杨书新博士、肖琴博士、张勇伟博士、刘关俊博士、江左文博士等在研究工作中的合作和帮助，以及在生活中的关心和鼓励。

感谢我聪明可爱的女儿欧阳逸佳在生活中带给我的快乐和精神上的慰藉。感

谢我慈爱的母亲和家公家婆、亲爱的兄弟、兄弟媳妇、妹妹和妹夫，是你们一直以来的默默鼓励、无私关爱和奉献与支持使我能取得今天的成绩。感谢我已经过世的伟大父亲，感谢他多年来对我的支持与辛勤养育之恩，即使在他身患重病深受病痛折磨时依然鼓励我不要伤心，极力支持我继续完成博士学位的攻读，父亲的教诲永远铭记于心，他虽已不在，但永远活在我心中！

本书的出版得到了国家自然科学基金项目（61462034、61561024、61562037）、江西省自然科学基金项目（20151BAB207035）、江西省教育厅科学技术研究项目（GJJ160632）、江西理工大学科研基金重点课程（NSFJ2014－K11）和江西理工大学资助出版，在此深表感谢。本书参考或直接引用了一些国内外论文和学术著作，在此向这些文献的作者表示感谢。本书的笔者任职于江西理工大学，在撰写本书的过程中，笔者查阅了大量文献，希望竭尽全力写好本书，但由于笔者水平有限，时间仓促，书中难免有不足之处，欢迎批评指正，关于本书的建议，请发往笔者的电子邮箱：wang_xi_happy@163.com。

王 曦 欧阳城添

2018年9月于江西理工大学

目 录

第1章 绪 论

1.1 研究背景

近年来，随着计算机技术的日益成熟，硬件成本的迅速降低，各种结构复杂、功能强大的嵌入式系统被广泛应用到航空航天、交通运输、核电能源和医疗卫生等领域中，并发挥着不可替代的作用，但是，一旦这些系统失效，将给人类的生命、财产和环境造成重大甚至是灾难性的损失和破坏。因此，这类嵌入式系统通常被称为安全苛求计算机系统(safety-critical computer system)[1]，简称为安全苛求系统(safety-critical system)[2,3]。这类系统不断地向纵深发展，系统的控制逻辑与功能实现日趋复杂，为了进一步提高系统的性能并降低使用和维护成本，许多原来由硬件实现的逻辑功能日益通过计算机软件来替代，这为大型复杂安全苛求系统的设计与开发提供了有效的实现途径。但由于软件自身的灵活性使得其存在很多不可判定的因素，其广泛应用于安全苛求系统的同时，也给系统的安全性带来了危害与隐患。在过去的几十年里，有很多发生于安全苛求系统中的灾难性事故都源于系统软件的失效。

1991 年海湾战争，美国爱国者导弹由于软件计时系统累计误差造成拦截失败，误伤了 28 名美国士兵。

1996 年，欧洲研制的阿丽亚娜 5 型运载火箭首航，因系统软件引发的问题导致火箭在发射 39 秒后偏轨，激活了火箭的自我摧毁装置，造成 3.7 亿美元的经济损失。

1997 年韩国一架客机由于飞行高度控制软件系统故障而坠毁，造成 200 多人死亡。

2000 年从美国 Multidata 公司引入的治疗规划软件，由于软件中存在错误的辐射剂量预设值，造成了有些患者因为接受了超标剂量的治疗而死亡。

2003 年韩国大邱，由于列车运行控制系统故障发生客运列车与货运列车相撞事故。

2004 年美国一架 F－22 飞机因飞行控制软件故障而坠毁。

2005 年华盛顿，由于列车运行控制系统没有及时地检测出前方有列车，发生地铁列车相撞事故。

2009 年华盛顿，由于信号系统故障，处于自动运行模式下的 112 次列车撞上静止等候的 214 次列车尾部，造成重大的人员伤亡事故。

2011 年温州，由于存在信号系统故障等问题，D3115 次动车与 D301 次动车发生追尾，造成重大的人员伤亡事故。

事例不胜枚举，由此可见，对安全性有着最高要求的安全苛求计算机系统中，软件的安全性尤为重要。因此，人们常把运行在安全苛求系统上的软件称为安全苛求软件(safety－critical software)。区别于普通软件，安全苛求软件以安全性作为第一性能特征，其失效将可能导致安全苛求系统失效，进而给人类生命财产和环境造成灾难性损失。

为了确保安全苛求软件的质量，尤其是安全性，防止灾难性事故的发生，科技人员提出了诸多理论和方法，其中模型检测(model checking)[4-6]以其简洁明了和自动化程度高而引人注目，其算法和应用研究成了学术界和工业界研究的热点话题之一[7,8]。

1.2 国内外研究现状

1.2.1 模型检测相关研究

模型检测[4,5,31-34]是关于系统属性(例如系统的活性、安全性、可达性等)验证的一种方法，它提供了一个完整的系统属性验证框架[35,36]，通常采用状态空间搜索的方法来自动检测一个给定的计算模型是否满足约定的系统属性。模型检测的基本思想是用状态迁移系统 M 来刻画系统的行为，用逻辑公式 F 描述系统的属性，验证的方法是判断状态迁移系统 M 是否满足公式 F，用数学公式表达为 M |＝F?，对有穷状态系统，可用计算机程序在有限的时间内自动判定 M 是否满足公式 F。模型检测能够自动进行验证，并且能够在系统 M 不满足公式 F 的情形下给出反例，据此可对系统 M 进行改进，从而方便了系统的修改和维护。模型检测的这些独特优点使得其在诸多形式化方法[37-42]中引人注目。

模型检测的研究始于 20 世纪 80 年代初，当时，Clarke，Emerson 等人提出了用于描述并发系统性质的计算树逻辑(computation tree logic，简称 CTL)[6,43-49]，设计了检测有穷状态系统是否满足给定 CTL 公式的算法，并实现了一个原型系统。这一工作为对并发系统的性质自动进行验证开辟了一条新途径，成为多年来计算机科学基础研究的一个热点[7]。随后不久出现的符号模型检测技术[31]使这

一方法向实际应用迈出了关键性的一步。目前，模型检测已被用于计算机硬件、通信协议、控制系统、安全性设计等方面的分析与验证中，取得了令人瞩目的成就，并从学术界辐射到了工业界。如今，许多大公司，如 Intel、HP、Philips 等成立了专门的小组负责将模型检测技术应用于生产过程中。Bryant，Clarke，Emerson 和 McMillon 因模型检测的创始性工作获得了 1998 年的 ACM 巴黎卡纳拉基斯理论与实践奖(Paris Kanellakis Award for Theory and Practice)[7]。

模型检测算法和应用研究[9, 50-75]成了近年来学术界和工业界研究的热点之一[7]。符号模型检测技术已取得了突破性进展，可以处理状态数多达 10^{120} 的系统。基于自动机理论的线性时序逻辑(linear temporal logic，简称 LTL)[76-82]模型检测[32, 33, 57]主要采用状态空间搜索的方法来自动检测一个给定的计算模型是否满足约定的系统属性，在算法实现上主要分为三个阶段：(1)将代表系统属性的 LTL 公式取反并转化为广义 Büchi 自动机；(2)将广义 Büchi 自动机转化为 Büchi 自动机；(3)对 Büchi 自动机与系统模型相对应的自动机作同步积运算，按需产生同步积自动机，此同步积自动机在结构形式上为 Büchi 自动机，采用 on-the-fly 判空检测算法判断它是否为空。在该类方法中，有关 LTL 公式取反并转化为 Büchi 自动机的算法在文献[56, 58, 61, 62]中已有广泛的研究，有关自动机同步积的运算可参阅文献[4]。on-the-fly 判空算法通常用于带单个可接受条件的 Büchi 自动机的判空检测[60, 63-65, 68]，在最坏情况下的运行时间与 Büchi 自动机的大小呈线性关系。但 Büchi 自动机的状态空间与 LTL 公式的长度呈指数关系，验证过程中容易引起状态爆炸。带多个可接受条件的广义 Büchi 自动机与 Büchi 自动机相比，其状态空间更少。由于广义 Büchi 自动机的判空检测比较复杂，目前有关这方面的研究不多。相关文献[59, 67, 71]介绍在通常情况下能检测出广义 Büchi 自动机的非空性，但在检测过程中，由于其对结点后继的选择是任意的，作出非空性判断的时空消耗比较大。

由于模型检测基于状态搜索的基本思想，搜索的可穷尽性要求系统模型的状态数有穷，故不能直接对无穷状态系统进行验证。即使对于有穷状态系统，模型检测也会面临着“状态空间爆炸(state space explosion)”的严重问题。如何有效缓解“状态爆炸”是模型检测能被广泛使用的一个重要前提。在这方面已有一些重要的方法被相继提出，主要包括符号模型检测方法、抽象技术、偏序归约、on-the-fly 技术、分解与组合、对称技术等。

符号模型检测(symbolic model checking，简称 SMC)[6, 31, 44-49]是一种采用符号方法表示状态空间的模型检测技术。该方法主要基于二元决策图(binary decision diagram，简称 BDD)[43, 83, 84]的符号化表示法，将系统的迁移关系表示成 BDD，其中 BDD 通过布尔函数 $f(x_1, x_2, \cdots, x_n)$ 表示 n 元组中每个变量 $x_i(1 \leqslant i \leqslant n)$ 的取值与集合之间的对应关系[85]，是一种比合取范式(conjunctive normal form，

简称 CNF)和析取范式(disjunctive normal form，简称 DNF)更便捷的表示形式，在数字电路和通信协议的各种设计和分析中有广泛应用。基于 BDD 的符号化表示已经在许多模型检测算法和系统中有着成功应用，其中最具影响的是美国 CMU(Carnegie Mellon University)计算机学院的 McMillan 博士在其博士论文中提出的符号模型检测工具 SMV 系统[86]，并通过它成功地发现了 IEEE Futurebus + 标准(IEEE Std 896.1 - 1991)中描述的 Cache 一致性协议中的错误，这也是使用自动验证工具首次发现工业标准的错误。但是 BDD 表示也并非万能的，寻求比 BDD 更简明的符号化表示方式是符号模型检测今后的一个研究方向。

抽象技术[4]是除符号模型检测方法外处理“状态爆炸”问题的另一种非常有效的方法。传统的模型检测方法主要适用于控制系统，不太适用于与数据路径有关的电路系统或具有复杂数据结构的反应式系统(reactive system)。符号模型检测方法虽然可以检测一些与数据处理有关的系统，但其验证的复杂性通常比较高。对于这类系统的验证，通常需要采用数据抽象技术，即在系统的精确数据值和一个小的抽象数据值之间建立一个映射关系，通过扩展状态和转换之间的映射，产生一个比实际系统小得多的抽象系统。还有一种重要的抽象技术是状态合并，为了压缩状态空间，它通过消除一些不影响规范验证的变量状态，得到简化的自动机模型，通过验证简化模型的性质来降低模型检测的复杂性。

在具有并发关系的模型中，状态变量或迁移序列的交替执行往往会导致状态爆炸这一现象，而偏序归约(partial order reduction)[85]是为此而提出的一种重要技术。偏序归约(partial-order reduction)是基于偏序(满足自反、反对称和传递关系)计算的计算模型，在具有偏序计算关系的多个交替序列中，验证时只需要分析其中的一个交替序列即可，因此偏序归约通过发掘模型中并发执行的迁移的交替性，可减少本质上相同的状态，生成足以检验性质的小部分状态空间，从而简化了模型的状态空间。目前几种主要的偏序归约技术包括顽固集技术[87]、睡眠集(sleep set)技术[88]以及覆盖步图(covering step graph)技术[89]等。Bell 实验室的 Holzmann 等人使用偏序归约等技术研制了模型检测工具 Spin[32]，NASA/WVU 软件研究实验室利用 Spin 验证了一个航天容错控制软件，发现了软件中进程优先规则和互斥规则冲突的 3 个异常。

On-the-fly[8, 56, 59, 60, 64, 65, 68]技术事先根据系统的被验证属性构造属性自动机，然后在生成系统模型与属性自动机的同步积自动机时，利用属性自动机去引导同步积自动机的动态构造，在找到违反被验证属性的反例之前，仅需构造同步积自动机的一小部分状态空间，从而避免了对同步积自动机整个状态空间的搜索。目前，基于 LTL 的 on-the-fly 技术已在一些模型检测工具 Spin[32, 33]、Prod[90]、Pep[91]中得以实现。

组合推理[4]是一种基于局部状态空间检测的方法。对于大系统的验证，组合

推理方法利用“分而治之”策略，根据系统的部件组成或模块结构，先分别验证系统各个部件或模块的局部性质，再由各个部件或模块的性质推断整个系统的性质。如果系统满足每个局部性质的要求，且局部性质符合整个规范，那么完整的系统也必定满足整个系统规范。

对称技术主要适用于含有许多对称重复部件的有穷并发系统(如一些协议和硬件)，它采用置换图来定义系统状态空间上的等价关系，并通过划分等价类来简化系统的状态空间。

模型检测能够自动进行验证主要依赖于模型检测工具的支持，有代表性的模型检测工具主要有 SMV、NuSMV、Spin、Verisotf、Java Path Finder(JPF)、PAT、UPPAAL 这几种，与其相应的建模语言与逻辑语言如表 1.2.1 所示。

表 1.2.1 几种有代表性的模型检测工具

模型检测工具	建模语言	逻辑语言
SMV	Kripke 结构	CTL
Spin	Pormela 语言	LTL
NuSMV	Kripke 结构	CTL, LTL
Verisotf	C、C + +、Tel 等程序设计语言	VS_assert 语句
JPF	Java 程序	JPF 断言
PAT	CSP#语言	LTL
UPPAAL	Timed Auomata, C subset	C subset, TCTL

SMV[86]是 Carnegie Mellon University 开发的模型检测工具，用以检测有限状态系统是否满足约定的 CTL 公式。其输入语言以 Kripke 结构作为语义模型，描述有限 Kripke 结构上的状态迁移关系，并支持模块化建模方式。其中模块描述的基本要素包括不确定选择、状态迁移和并行赋值语句，模块间基于共享变量方式进行通信。SMV 模型检测的基本方法是用有序二元决策图(OBDD)表示状态迁移关系，用计算不动点的方法检测状态的可达性和其他的性质。

Spin[32]是 Bell 实验室开发的模型检测工具，用以检测有限状态系统是否满足 LTL 公式及其他一些性质，如可达性和死锁等。其输入语言是语法上类似于 C 语言的 Pormela 语言，其程序由进程、通道和变量构成，其中进程类型(procytpe)用来定义类进程的行为规范，其基本要素包括赋值语句、条件语句、输入/输出语句、不确定选择和循环语句，进程间通过消息传递的方式进行通信。Promela 支持两种通道模式：一种是 FIFO 缓冲区模式，用以实现异步消息传递；另一种是 erndezvous 模式，即容量为零的缓冲区，用以实现同步消息传递。Promela 程序中