



计算机科学

COMPUTER SCIENCE

自动推理技术在求解组合数学难题中的研究进展

黄沛, 刘明昊, 马菲菲, 张健

引用本文

黄沛, 刘明昊, 马菲菲, 张健. 自动推理技术在求解组合数学难题中的研究进展[J]. 计算机科学, 2023, 50(7): 167-175.

HUANG Pei, LIU Minghao, MA Feifei, ZHANG Jian. Automated Reasoning Techniques for Solving Combinatorial Mathematical Problems:A Survey [J]. Computer Science, 2023, 50(7): 167-175.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

基于经验动态建模的因果检测研究综述

Review on Causality Detection Based on Empirical Dynamic Modeling

计算机科学, 2023, 50(6A): 220600194-12. <https://doi.org/10.11896/jsjcx.220600194>

人工智能可解释性: 发展与应用

Explainability of Artificial Intelligence: Development and Application

计算机科学, 2023, 50(6A): 220600212-7. <https://doi.org/10.11896/jsjcx.220600212>

基于冲突搜索的多智能体路径规划研究进展

Research Progress of Multi-agent Path Finding Based on Conflict-based Search Algorithms

计算机科学, 2023, 50(6): 358-368. <https://doi.org/10.11896/jsjcx.220800151>

深度强化学习中的知识迁移方法研究综述

Survey on Knowledge Transfer Method in Deep Reinforcement Learning

计算机科学, 2023, 50(5): 201-216. <https://doi.org/10.11896/jsjcx.220400235>

基于可解释性人工智能的软件工程技术方法综述

Review of Software Engineering Techniques and Methods Based on Explainable Artificial Intelligence

计算机科学, 2023, 50(5): 3-11. <https://doi.org/10.11896/jsjcx.221100159>

自动推理技术在求解组合数学难题中的研究进展

黄沛^{1,3} 刘明昊^{1,3} 马菲菲^{2,3} 张健^{1,3}

1 中国科学院软件研究所计算机科学国家重点实验室 北京 100190

2 中国科学院软件研究所并行软件与计算科学实验室 北京 100190

3 中国科学院大学 北京 100049

(huangpei@ios.ac.cn)

摘要 自动推理是一种以符号演算的方式来自动模拟人类逻辑推理能力的技术,其总体目标是利用计算机构建一个将不同形式的推理机械化的系统。虽然该领域的理论框架尚未实现对人类全部推理能力的模拟,但该领域的发展已经可以帮助研究人员解决一些数学和逻辑领域的开放性问题,并提供了计算机科学中的重要应用。文中简要回顾了利用自动推理技术处理组合数学开放性难题时的代表性方法,重点梳理了该领域的国内外最新进展,分析了各种方法的优势与不足,介绍了近年来出现的增强自动推理结果可信性的技术方法,并探讨了未来的研究方向和面临的挑战。

关键词: 自动推理;组合数学;计算逻辑;人工智能;符号演算

中图分类号 TP181

Automated Reasoning Techniques for Solving Combinatorial Mathematical Problems: A Survey

HUANG Pei^{1,3}, LIU Minghao^{1,3}, MA Feifei^{2,3} and ZHANG Jian^{1,3}

1 State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China

2 Laboratory of Parallel Software and Computational Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China

3 University of Chinese Academy of Sciences, Beijing 100049, China

Abstract Automated reasoning is a symbolic algorithmic technique that aims to simulate the logical reasoning ability of human. The overall goal is to mechanize different forms of reasoning with a computer system. Although the theoretical framework of the field has not yet supported the simulation of the full range of human reasoning capabilities, developments in the field have reached a point where automated reasoning programs are being used by researchers to attack open problems in mathematics and logic and provide important applications in computing science. This paper briefly reviews common approaches of automatic reasoning in dealing with open problems in combinatorial mathematics and highlights the latest developments in this field in China and abroad. Then the strengths and weaknesses of various approaches are analyzed, and reasoning strategies that have emerged in recent years to enhance the trustworthiness of automated reasoning results are introduced. Finally, future research directions and challenges are discussed.

Keywords Automated reasoning, Combinatorial mathematics, Computational logic, Artificial intelligence, Symbolic calculus

1 引言

逻辑作为一种在机械世界观下探究人类思维、计算活动的形式语言,可以被用来描述众多的计算问题。人们可以用逻辑公式将待处理问题描述出来,然后通过判定其公式的可满足性来获得问题的答案。而利用计算机自动化地去判定给定逻辑公式的可满足性的技术,被称为自动推理^[1]。同时,它也是自顶向下符号主义 AI 最具代表性的研究方向,其“自顶向下”的特征体现在无论问题的背景是什么,表达什么含义,

在判定逻辑公式可满足性的层面上看来,所有的计算和推理策略都是作用在逻辑符号上的变换,且这些策略都是根据逻辑公式特征制定的,不会因背景问题本身语义不同而有所不同。如果我们能设计出一个求解器可以快速地判定逻辑公式的可满足性,那么所有可以被逻辑表达的问题都能被高效地解决。然而,这样“美好”的顶层设计却因计算的局限性(可计算、判定性^[2]、NP 屏障^[3])而变得举步维艰。尽管如此,随着自动推理技术几十年来的进步,越来越多的应用问题可以被现代自动推理工具解决,特别是对于许多数学问题的求解^[4]。

到稿日期:2022-10-31 返修日期:2023-03-26

基金项目:国家自然科学基金(61972384,62132020)

This work was supported by the National Nature Science Foundation of China(61972384,62132020).

通信作者:张健(zj@ios.ac.cn)

数学定理自动证明一直都是自动推理领域最为核心的任务,它既是初代 AI 的象征,也是符号主义 AI^[5]最能体现其优势的一个领域。美国心理学家丹尼尔卡尼曼在《Think, Fast and Slow》中把人类的思维活动分为两类:系统 1(System 1)是快速、本能和情绪化的;系统 2(System 2)是更慢的、深思熟虑的、逻辑的。长久以来,人们都将解出数学难题看作人类最具“智慧”的行为,这是因为绝大多数的人类并不擅长系统 2(System 2)的思考方式。自动推理技术就是在模拟人类系统 2 的智能,它以能解决许多令伟大数学家们“竞折腰”的数学难题而吸引着众多学者的研究。特别是组合数学问题,它一直是自动推理领域关注的重点。这一方面是因为利用自动推理技术解决组合数学领域的开放性难题可以促进组合数学的进步和发展;另一方面也是因为组合数学难题中的强组合性质会为自动推理造成巨大的困难,克服这些困难自然也有助于改进推理技术。

近十几年中,越来越多的组合数学难题被自动推理技术所解决,这也标志着自动推理这一符号主义 AI 方法在深度学习统治的时代还在持续不断的更新和进步。本文简述了自动推理技术在求解组合数学领域开放性难题中的相关研究进展,并讨论了未来该领域面临的挑战及研究方向。第 2 节介绍了逻辑公式的一些基本概念和判定其可满足性的基本框架;第 3 节叙述了近些年中自动推理领域所关注的组合数学问题的进展情况以及代表性的方法,并分别介绍了这些代表性方法的设计思路和优势;第 4 节介绍了目前关于自动推理技术在解决数学问题中的可信性问题方面的进展;最后总结全文并展望未来。

2 基本概念

2.1 逻辑公式

本文着重介绍在处理离散数学问题中常用的两种逻辑系统:命题逻辑与一阶逻辑。命题逻辑是最简单的一种形式逻辑系统。其主要研究对象为命题,每个命题有真、假两种可能的取值。通常用 p, q, x_1, x_2 等小写符号来表示命题,称其为命题变元;用 \perp, \top 来表示命题的真假值。通过逻辑联结词可将命题逻辑联结起来构成复杂命题。常见的一元逻辑联结词有:“ \neg ”(又称“非”);二元联结词有:“ \wedge ”“ \vee ”“ \rightarrow ”“ \leftrightarrow ”。一个命题公式,就是由这些符号按一定规则组成的一个符号串。

(1)命题逻辑变元是公式,这样的公式也叫原子公式;

(2) \perp, \top 是公式;

(3)如果 ϕ 是公式,则 $\neg\phi$ 是公式;

(4)若 ϕ_1, ϕ_2 是公式,则 $\phi_1 * \phi_2$ 也是公式,其中 $*$ 是二元逻辑连接符;

(5)所有公式都是有有限次使用(1)–(4)生成的符号串。

我们称原子公式或其否定形式为文字,称原子公式本身为正文字,其否定为负文字。若干个文字的析取被称为子句。子句长度指其所含文字的个数。只有一个文字的子句被称为单子句,没有文字的子句被称为空子句。若干个单子句的合取是一种特殊形式的公式,称为合取范式(Conjunctive Normal Form, CNF),其一般形式为:

$$(L_{11} \vee \cdots \vee L_{1n_1}) \wedge \cdots \wedge (L_{m1} \vee \cdots \vee L_{mm_m})$$

其中, L_{ij} 是文字。

从命题变元集到布尔值集合的函数叫做真值赋值,如果存在一个赋值使得公式为真,则这个公式是可满足的,这个赋值也称为一个公式的模型;否则称该公式是不可满足的。

一阶逻辑是一种比命题逻辑表达能力更强的语言,通常有下列 4 种符号。

(1)常量符号:用小写英文字母 a, b, c 等表示;

(2)变量符号:用小写英文字母 x, y, z 等表示;

(3)函数符号:用小写英文字母 f, g 等表示;

(4)谓词符号:用大写英文字母 P, Q 等表示。

谓词逻辑中的项,被递归定义为:

(1)常量符号是项;

(2)变量符号是项;

(3)若 $f(x_1, \dots, x_n)$ 是 n 元函数符号, t_1, \dots, t_n 是项,则 $f(t_1, \dots, t_n)$ 是项;

(4)所有有限次使用(1), (2), (3)生成的符号串。

若 $P(x_1, \dots, x_n)$ 是 n 元谓词符号, t_1, \dots, t_n 是项,则 $P(t_1, \dots, t_n)$ 是原子。谓词逻辑中的公式被递归定义为:

(1)原子是公式;

(2) \perp, \top 是公式;

(3)如果 ϕ 是公式,则 $\neg\phi$ 是公式;

(4) ϕ_1, ϕ_2 是公式,则 $\phi_1 * \phi_2$ 也是公式,其中 $*$ 是二元逻辑连接符;

(5)若 ϕ 是公式, x 是 ϕ 中的自由变量,则 $\forall x\phi, \exists x\phi$ 是公式;

(6)所有公式都是有有限次使用(1)–(5)生成的符号串。

谓词逻辑中公式 ϕ 的一个解释 I ,是由非空区域 D 和对 ϕ 中常量符号、函数符号及谓词符号以下列规则进行的一组指定组成。

(1)对每个常量符号,指定 D 中的一个元素。

(2)对于每个 n 元函数符号,指定一个函数,即指定 D^n 到 D 的一个映射。

(3)对于每个 n 元谓词符号,指定一个谓词,即指定 D^n 到 $\{0, 1\}$ 的一个映射。

如果存在解释 I ,使公式 ϕ 在 I 下取真值,则 ϕ 称为可满足的,简称 I 满足 ϕ ;若 I 不满足 ϕ ,则称 I 弄假 ϕ ;如果不存在解释 I 满足公式 ϕ ,则 ϕ 称为不可满足的(或者恒假的);如果 ϕ 的所有解释 I 都满足公式 ϕ ,则 ϕ 称为恒真的。

无论是在命题逻辑中还是一阶逻辑中,能使得公式 ϕ 为真的解释或者赋值,被称为模型。

2.2 自动推理方法

通常,待推理任务被形式逻辑语言所表达后,其自动推理的核心就是判定其对应的逻辑公式 ϕ 的可满足性。在计算机中,形式推理方法只分为两大类^[6]。

(1)基于证明论(Proof-theoretic)的方法:该方法用(Deduction)推理的机制来得到最终期望的结果。该机制中依赖的公理(Axioms)和推理规则(Inference Rules)合在一起被称为推理系统。

(2)基于模型论(Model-theoretic)的方法:该方法会在枚举框架下测试所有可能的模型来判定其公式的可满足性。

推理系统通常是由前因(Antecedents)推出后果(Consequents)的推理规则构成,例如双重否定规则: $\neg\neg p \rightarrow p$ 。在现代经常使用的命题逻辑可满足性判定技术中,一条非常重要的规则为:

$$\frac{\neg p \vee C \quad p \vee D}{C \vee D} \quad (1)$$

其中, p 是命题变元, C 和 D 是两个子句。这条规则被称为归结原理(Resolution)^[7]。其一种特殊形式表现为:

$$\frac{\neg p \vee C \quad p}{C} \quad (2)$$

这两条规则都是基于反演推理的,它们也是现代SAT求解器中最重要的两条推理策略。归结原理是以冲突子句学习(Conflict-Driven Clause Learning,CDCL)^[8]的形式出现,而其特殊形式(2)是以单子句传播的形式出现。

另一类基于模型论的方式更加常见,它的基本原理是在有限的空间中枚举或者搜索可能满足公式的所有模型。对于命题逻辑而言,这种方法的主要框架是利用回溯法枚举命题变元真假取值的可能。而对于有限域上的一阶逻辑而言,这种方法也在回溯法的框架中尝试变量的各种赋值以及函数的可能的映射关系。

事实上,在现代判定过程中,演绎和枚举是混合在一起使用的,以追求实践效率的提升。因为对于反演推理(或者反证法)而言,其优势在于证明公式的不可满足性。而对于搜索方法而言,其求解偏好是处理可满足的公式。常见的自动推理工具,如SAT求解器、SMT求解器和有限模型生成器,都采用了枚举法和演绎法混合的求解方式,自动推理专家Bonacina教授也将这种方式称为“半猜半推(Guess and Reasoning)”^[9]。例如,在目前的高效SAT求解器中,主框架是以回溯法展开真值表进行搜索,同时在搜索过程中利用单子句传播和归结原理来避免枚举许多不必要的情况。在各种启发式策略的辅助下,进一步提高其对各种应用问题的求解效率。这种模式实现最好的代表就是CDCL框架,它也是目前自动推理领域已知的能平衡“猜测”和“推理”的最好框架。目前,该框架也被逐步扩展到SMT的各个理论中。

3 组合数学问题自动推理技术的进展

组合设计理论与群论、图论、数论、有限几何以及线性代数理论都有着密切的联系,也常被应用于实验设计、编码、加密等问题;同时,它与计算机科学中的数据结构、算法、人工智能等领域也密切相关^[4]。早在1960年,人们就开始利用计算机来辅助专家处理组合数学中的难题。其中非常著名的一个事件就是Parker等使用计算机否证了数学家欧拉的关于两个正交10阶拉丁方存在性的猜想^[10],随后基于计算机发现的结果,该问题也在历时177年后被数学家们彻底解决^[11]。早期,人们会为特定的问题编写特定的算法;而从1990年开始,人们开始构建通用推理工具来处理这些组合数学的难题。根据文献^[4],Zhang在1991年基于一阶逻辑设计了第一个

可以处理组合数学问题的通用推理工具FALCON^[12],并解决了之前一直悬而未解的拟群相关开放性问题,该工作引起了自动推理社区的广泛关注。随后,在1992年Fujita等开发了定理证明工具MGTP^[13],并解得了大量曾经开放的拟群的问题。1993年,Fujita也在人工智能顶级会议IJCAI上以获奖论文的形式报告了他们的工作^[14]。1994年,Slaney开发了FINDER来推理组合数学的难题^[15]。几乎同时,Stickel,Zhang,McCune基于命题逻辑(SAT)分别开发了LDPP^[16],SATO^[17],MACE^[18]等推理工具,并且也解决了很多与拟群相关的开放性难题。表1列出了近20年中在自动推理领域关注较多的组合数学问题。

表1 自动推理领域关注的组合数学问题

Table 1 Combinatorial mathematics problems concerned in the field of automatic reasoning

| 英文名 | 中文名 |
|-------------------------------|---------|
| Quasigroups and Latin squares | 拟群与拉丁方 |
| Ramsey numbers | 拉姆齐数 |
| Van der Waerden Numbers | 范德瓦尔登数 |
| Covering arrays(CA) | 覆盖数组 |
| Orthogonal arrays | 正交数组 |
| CA numbers of small strength | 小强度覆盖数组 |
| Steiner systems | 斯坦纳系列 |
| Mendelsohn designs | 门德尔森设计 |
| Magic squares | 幻方 |

将常用的推理工具按一阶逻辑和命题逻辑划分,如表2所列。

表2 自动推理工具

Table 2 Automated reasoning tools

| | |
|--------|---|
| 基于一阶逻辑 | FALCON ^[12] ,SEM ^[19] ,SEMD ^[20] MGPT ^[13] ,FINDER ^[15] ,MACE ^[18] |
| 基于SAT | LDPP ^[16] ,SATO ^[17] ,MACE2 ^[18] MiniSAT ^[22] ,Glucose ^[23] |

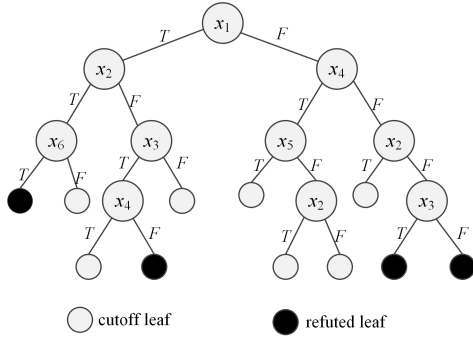
20世纪90年代,许多组合数学问题可直接由此类工具解决。但2000年后,大部分尚未解决的组合数学问题已经很难直接使用这些推理工具求解,人们往往需要根据问题结构、特性来定制化推理算法或者根据问题特点进行逻辑建模的优化。随着近20年深度学习的蓬勃发展,也有一些工作开始探索利用深度学习技术来处理数学难题并取得了一定的进展。后续章节将概述近年来在处理组合数学问题上常见的手段。

3.1 分块攻克法

随着问题编码和自动推理算法的改进,近年来很多离散数学定理在自动推理技术的帮助下得到了证明。然而,仍然有很多问题是在当前条件下难以解决的。在这样的背景下,一些研究人员选择通过提升算力来应对这些难题。这类工作的模式是首先将原问题分解为若干子问题,再将它们分别指派到多台部署了自动推理程序的计算机上进行并行求解,最后整合并返回结果。本文以目前SAT问题上最强大的并行求解方法之一——分块攻克法(Cube-And-Conquer Method)为例,说明其对SAT问题进行并行求解的模式,并介绍通过应用该方法完成的定理证明工作。

图1展示了分块攻克法^[24]的求解流程。首先,将原问题

分解为若干被称为“块”(Cube)的子问题,然后调用SAT求解器对它们进行求解。其中的关键是分块方法的设计,要使得这些子问题的求解难度更小,从而降低整体用时。为此,分块攻克法使用了带有多种启发式策略的前瞻(Look-ahead)求解器,对原公式 F 进行顶层的分支决策,并将每个分支上已决策的文字表示为一条合取式 C ,那么待求解的块可表示为 $F \wedge C$ 。



- $F_1 := F \wedge (x_1 \wedge x_2 \wedge \neg x_6)$
- $F_2 := F \wedge (x_1 \wedge \neg x_2 \wedge x_3 \wedge x_4)$
- $F_3 := F \wedge (x_1 \wedge \neg x_2 \wedge \neg x_3)$
- $F_4 := F \wedge (\neg x_1 \wedge x_4 \wedge x_5)$
- $F_5 := F \wedge (\neg x_1 \wedge x_4 \wedge \neg x_5 \wedge x_2)$
- $F_6 := F \wedge (\neg x_1 \wedge x_4 \wedge \neg x_5 \wedge \neg x_2)$
- $F_7 := F \wedge (\neg x_1 \wedge \neg x_4 \wedge x_2)$

图1 分块攻克法求解流程示意图
Fig. 1 Cube-and-conquer method

由于分块攻克法可以快速地并行求解困难的SAT问题,同时各台计算机之间无需进行任何通信,带来了很大的效率提升,因此它在很多定理自动证明的工作中得到了应用。其中最著名的工作是Heule等于2016年提出的对布尔毕达哥拉斯三元组问题(Boolean Pythagorean Triple problem)的证明。它是组合数学中一个长期未被解决的难题:给定从1到 n 的 n 个自然数,能否将它们划分为两个集合,使得任何一个集合中的元素都不能构成一个毕达哥拉斯三元组 (a, b, c) ,即 $a^2 + b^2 = c^2$? 通过将该问题编码为布尔逻辑公式并应用一些化简规则,可转化为求解包含3700余个变量和14600余个字句的SAT问题。而后,使用分块攻克法将该问题分解为106个块,并在由800个核心组成的计算机集群上运行了35000个CPU时,解决了该问题。本工作的另一个特点是所生成的证明文件十分巨大——超过200TB,仅为该证明运行验证器就花费了16000个CPU时,《Nature》将其称为“全世界最长的数学证明^[25]”。

作为一种通用的并行求解策略,分块攻克法近年来已帮助解决了很多组合数学领域的未解问题。2018年,Heule对Schur数进行探索,首次证明了 $S(5) = 160$,这是自1965年Golomb等证明 $S(4) = 44$ 以来该问题的最新进展^[26]。为了使用SAT求解器计算该问题,将原公式分解为约 10^7 个块,花费约14个CPU年完成全部运算,所生成的证明文件体积也创下了2PB的新纪录。2020年,Brakensiek等对几何学中存在90多年的Keller猜想进行了研究,通过分块攻克法求解

一组SAT问题,从而彻底解决了该猜想^[27]。

3.2 混合策略

3.2.1 一阶逻辑与命题逻辑组合

一阶逻辑和命题逻辑在有限域上表述某一计算问题时各有优势。使用一阶逻辑表达问题时,其表达的方式更加紧凑,保留了背景问题的结构和语义。Zhang等^[20]分析了一阶逻辑在有限域上表达的特点,比如在推理过程中,如果某些常量尚未在解释中被指派,它们的地位可能会是平等的(或者说是对称的),这样我们只需要使用最小的那个元素来完成解释的构建,该方法也被称为“最小取数法(Least Number Heuristic, LNH)”。它是一种代价极小的动态删除大量同构子问题的方法。考虑公式:

$$\forall x, y(f(x, y) = f(y, x)) \wedge (\forall x, y(f(f(x, y), x) = y)) \quad (3)$$

如果想在4元有限域 $\{0, 1, 2, 3\}$ 上构造式(3)的模型,最先对于 $f(0, 0)$ 而言,我们只用考虑为其指派0或者1,其中0是已经出现过的元素,而元素2,3此时可以被看作与1是对称的,或者说我们总能构造出一个置换证明公式在 $f(0, 0) = 1$ 的情况下与 $f(0, 0) = 2$ 的情况下可满足性一致。图2展示了LNH方法对式(3)模型构造过程中产生的剪枝效果,图中左边树结构是不使用LNH策略时需要探索的空间,而右边是在使用LNH策略时产生的剪枝效果。

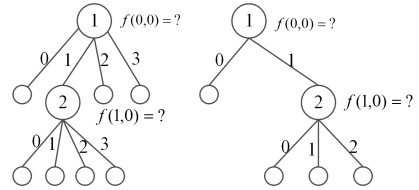


图2 LNH示意图

Fig. 2 Diagram of LNH

与一阶逻辑相比,命题逻辑的优势在于其表达形式简单、统一,有利于实现简单高效的推理机制。SAT求解方法在经历了几十年的发展后,其搜索策略和数据结构都经历了多代专家的精心打磨。最重要的是,它在求解过程中遇到冲突时可以分析冲突产生的原因,并以子句的形式记忆该冲突,这样在后续求解过程中求解算法就可以规避相同的冲突。早在2001年,Zhang^[28]就首先尝试将这两者的优势结合在一起,尽管当时使用的SAT求解方案还是基于Davis-Putnam过程的方法,但其展现出来的效率提升已经揭示了一阶逻辑和命题逻辑混合求解的潜力。2018年,Huang等^[29]将一阶逻辑和基于CDCL的SAT求解方案结合并设计了FOL+SAT的求解模式,该混合策略的框架如图3所示。

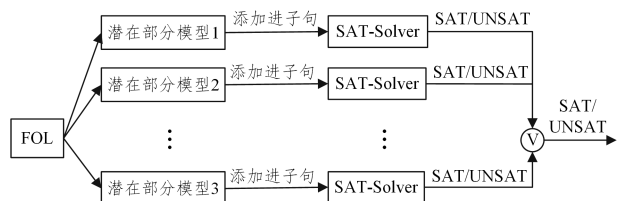


图3 一阶逻辑+命题逻辑

Fig. 3 FOL+SAT

该方法在问题的顶层利用一阶逻辑方便识别对称和打破对称的特性先求出问题的部分模型,然后将这些已经被消除过许多同构情况的潜在模型继续交由命题逻辑求解器处理。基于该方法,Huang 等将拟群大集问题的求解能力从以前的只能处理 8 阶以下的问题改进到了可以处理 13 阶的情况,并发现了新的幂等拟群大集存在性的结果。而这些新结果是使用单一求解方案均无法在一周时间内取得的。

3.2.2 命题逻辑与计算机代数系统组合

计算机代数系统(Computer Algebra System,CAS)在多年的发展和积累中拥有了大量的算法库,因此它很擅长处理具有特定领域知识的代数问题。常见的 CAS 系统有 MAPLE^[30], MATHEMATICA^[31], AGE^[32] 和 MAGMA^[33]。Bright 等认为利用代数系统中的领域知识可以裁剪掉大量的搜索空间并帮助命题逻辑或一阶逻辑加速推理过程,因此他们在 2016 年提出了 SAT+CAS 这种将命题逻辑推理和代数系统混合的求解模式^[34]。SAT+CAS 的工作原理如图 4 所示。

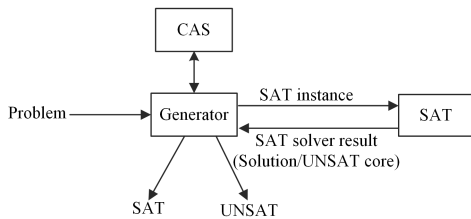


图 4 命题逻辑+代数系统

Fig. 4 SAT+CAS

图 4 中,中心的生成器由两个接口构成:一个接口联结 CAS 系统并将问题以 CAS 函数的方式交由代数系统并生成可以修剪组合空间的引理(Lemma);另一个接口是将生成的组合引理转变为命题逻辑公式并交由 SAT 处理并获取最终结果。基于该方法,Bright 等成功验证了 Hadamard 猜想小于 156 阶的情况,并且验证了 Williamson 矩阵不存在 35 阶这一情形,这一结果也断言了 35 是 Williamson 矩阵不存在的最小阶数。随后,他们又用 SAT+CAS 的模式解决了长度小于 25 的 Golay Pairs 问题^[35]、Good Matrices^[36] 问题以及 Lam's 问题中的编码问题^[37]。

3.2.3 启发式搜索与逻辑推理混合

局部搜索是一种以经验主义和灵感为指导的搜索方法。尽管启发式方法缺少严谨的逻辑指导,但它往往会使得一些问题的解决变得更加高效。人类在解决数学问题时,经常也是将启发式和逻辑推理相混合的。例如,在证明某一数学结论时,需要将某一元多项式写成两项乘积的形式:

$$x^2 + 5x + 6 \Rightarrow (x + 2)(x + 3)$$

这样的字符串改写既是形式主义的,又是启发式的。因为公式 $x^2 + 5x + 6 = 0$ 的两个解(模型) $x = -3$ 和 $x = -2$ 是我们根据经验“猜”出来的,并不是基于严谨的形式主义推理获得的。此类方法虽然高效,但多半是非完备的,我们很难对任何表达式都“猜”出它的两个解。但是在推理过程中加入一些“猜测”的成分,可以极大地促进求解效率的提升。鉴于此观察,Huang 等采用了一种将启发式与逻辑推理混合的方式

来处理组合数学相关问题^[38],他们的基本想法可以由图 5 来概括。

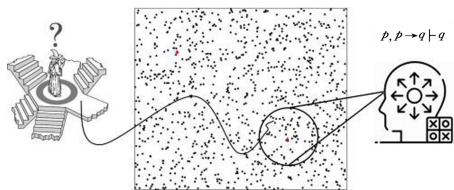


图 5 启发式+命题逻辑

Fig. 5 Heuristic+SAT

该方法的主要思想是在公式求解的上层基于启发式方法优先确定一些变量的赋值或者函数的解释,此时公式化简后就对应着由这些变量确立下来的子空间。这样的一个子空间可以看作是由启发式策略的一次尝试所选定的,然后可以将这一子空间对应的逻辑公式交由完备的逻辑求解器进行推理。Huang 等基于该方法找到了关于正交 Golf 设计的新结果,改进了 17 阶以下的已知情形。

3.3 公式重写方法

同一问题以不同的逻辑公式描述,其求解难度也会不同,因此如何巧妙地将问题编码为易于推理和计算的逻辑公式,在求解实际组合数学难题时也是非常值得关注的。

最常见的一种公式重写方法就是通过添加公式(或约束)来消除问题中所存在的同构情形。形式上来说,若我们想要判定公式 ϕ 的可满足性,则我们有时可以根据问题的语义找到一个公式 ω 使得公式 $\phi \wedge \omega$ 的可满足性不变,但是 ω 在几乎不增加计算复杂度的同时又能消除 ϕ 公式中大量潜在的可能模型。因为组合问题通常都存在着大量的对称情形,因此这种公式改写方式在求解组合数学问题中十分常见。

另一种方式是直接将公式 ϕ 中的一部分或者全部转变为语义等价的公式 ϕ' ,其中 ϕ' 与 ϕ 的可满足性等价,但是 ϕ' 更易于推理和计算。例如,对于经典的关于两个正交拉丁方是否存在的欧拉猜想问题,当我们使用横截建模的方式去寻找一个拉丁方的正交伙伴时,会节省大量的时间。正如图灵奖获得者 Knuth 在《The Art of Computer Programming》一书第 4A 卷中描述的那样:使用横截建模的方式去解决 10 阶欧拉猜想的情况,可以让搜索算法节省出一个 10^{12} (!) 因子的计算量。Ma 等^[39]在她们的作品中首次展示了横截建模的方式可以以几十倍地加速一阶逻辑求解器在证明正交拉丁方相关问题时的效率。随后, Jin 等使用这种建模方式改写公式,并成功求解出 10 阶以下带有特殊性质的 Costas 拉丁方的未解情形^[40]。

3.4 基于深度学习的方法

机器学习特别是深度学习技术在感知智能方面已经取得了巨大的成功,这也促使人们思考如何将它应用于求解高度逻辑化的数学问题上。其中具有代表性的工作就是 2021 年 Deep Mind 团队提出了一种机器学习框架并几乎证明了一个卡兹丹·卢思提格多项式的古老猜想;除此之外,他们还基于该框架观察到代数和几何不变量之间的惊人联系,这也被称为利用机器学习做出的第一个重大发现^[40],该成果也被刊登

于著名杂志《Nature》上^[41]。在组合数学方面,Adam 等提出一种深度交叉熵方法的算法来构造一些图结构并反驳一些数学猜想。在该方法中,神经网络只学习预测给定状态下最佳的移动路径,而不学习状态或者状态-动作下的值函数。给定任意一个状态作为神经网络的输入,然后输出该状态下所有可能移动的概率分布,概率最好的代表最佳路径。举例说明,假设有关于图的匹配值与最大特征之和猜想:设 G 是 $n \geq 3$ 个顶点的连通图, λ_1 是最大特征值, μ 是匹配数,那么它们满足不等式:

$$\lambda_1 + \mu \geq \sqrt{n-1} + 1$$

值得注意的是,它对于 $n \leq 18$ 都是正确的,最小反例是 $n \leq 18$ 都是正确的,最小反例是 $n = 19$ 。如图 6 所示的最后一个图结构,它有最大特征值 $\sqrt{10}$ 和匹配数 2,所以 $\lambda_1 + \mu \approx 5.16 < 5.24 \approx \sqrt{(19-1)} + 1$ 。

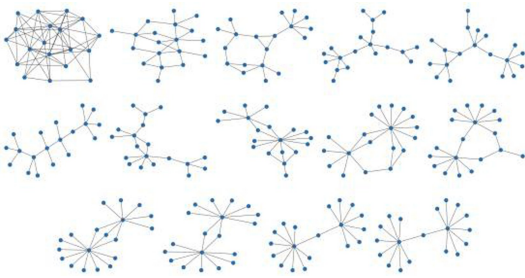


图 6 19 个顶点的反例演化过程

Fig. 6 Evolution of counterexample of 19 vertices

图 6 中过程显示了最佳构图是如何随着学习过程迭代次数增加而变化的,这个过程中神经网络很快就发现了树状是最好的,然后它会逐渐收敛到这一情形。该工作基于这种方法求解了:1)关于图的最大特征值和匹配数之和的猜想^[42]; 2)Aouchiche 和 Hansen 提出的关于图的距离谱和邻近性的猜想^[43]; 3)Graham 和 Lov'asz 关于距离测度的猜想^[44]; 4)距离拉普拉斯算子的余谱下,图的传输正则性不保持^[45]; 5)可以用很少的超平面覆盖超立体的某些子集^[46]。

4 自动推理的可信性

通过编写和运行计算机程序进行自动定理证明,固然有其优越性。然而,由于程序设计技术的复杂性,如何确认编写的程序正确地表示了证明目标,实际上是非常困难的。如果程序中出现了逻辑错误,其运行结果可能对定理证明产生误导,甚至会直接导致伪证。事实上,1976 年 Appel 和 Haken 对四色定理的证明发表后就曾经被指出存在错误^[47],这也导致通过计算机进行的定理证明工作长期以来无法被很多数学家所信任。近年来,自动推理研究人员做出了大量努力来提高自动定理证明的可信度,除了对主流求解器进行大量测试和验证工作之外,还构建了一系列证明系统(Proof System),使得工具的运行结果也可以通过统一算法得到检验,并逐渐成为自动定理证明工作的标准配置。

4.1 证明系统

由于证明系统在 SAT 问题的求解方面发展最为完善,本文主要以该问题为例进行介绍。对于一个命题逻辑公式,

SAT 求解器会返回可满足和不可满足两种结果之一。对于可满足的结果,很容易通过将每个变量的赋值代入公式进行检验;而对于不可满足的结果,验证其正确性则会困难得多。历史上,很多 SAT 求解器都曾经被发现存在重大缺陷^[48],这对于定理证明和安全性验证等对可靠性要求很高的应用而言是不可接受的。为此,研究人员提出了证明系统的概念,在求解器返回不可满足结果的同时,输出一个被称为证明(Proof)的文件,它可以被用来验证该问题的不可满足性。根据前文介绍,当前的 SAT 求解器主要基于 CDCL 算法,其判定问题不可满足的依据主要是在子句(包括学习到的子句)间产生了矛盾(空子句)。因此,证明文件的主要内容就是对原始子句到最终矛盾的过程性记录。具体而言,就是将求解中学习到的与最终矛盾的产生相关的子句均保留下来,使得验证器可以根据这些子句逐步地从原问题推导至矛盾,从而确认该算法的正确性。由于验证器的算法实现相对求解器要简单得多,而且这样的证明形式更加方便第三方独立验证,因此证明系统对于高可信的自动定理证明是不可或缺的。

4.2 证明系统的演进

SAT 求解器的证明系统最早于 2003 年在 zChaff 中被实现^[49],其每条学习子句和最终矛盾的产生过程都被详细记录在日志文件中。根据这些信息,验证器可以很自然地生成从原始子句到空子句的一条归结链(Resolution Chain),因此能够验证结果的正确性。此类方法也被称为归结证明系统(Resolution Proof System, RES)。但是,由于日志文件中包含了大量冗余信息,因此对于数学定理证明等比较复杂的问题,其所需的存储空间将会非常巨大。

为了使证明系统的表示更加紧凑且易于验证,研究人员近年来提出了多种重要改进。反向单元传播(Reverse Unit Propagation, RUP)证明系统显著地提升了证明的紧凑程度^[50]。其主要思想是只按照顺序保留学习子句,而不再记录它们的产生条件。在验证阶段,验证器按顺序依次验证每条学习子句的正确性。假设待验证子句为 C ,原公式为 F ,为了验证 $F \models C$,RUP 先令 C 为假,得到若干单子句,将它们加入 F 后进行单元传播,如果产生了矛盾则表示子句 C 通过验证,将 C 加入 F 后继续验证下一条子句。可以发现,对于同样的问题,RUP 占用的存储空间相比 RES 明显降低,但每一步都可能需要在整个公式上进行推理。RUP 证明系统较好地平衡了证明紧凑程度和验证成本的关系,使得人们可以为更大规模的 SAT 问题生成证明。此外,还有带删除信息的反向单元传播>Delete Reverse Unit Propagation, DRUP)证明系统。它利用学习子句可能蕴含现有子句的性质,将被蕴含的子句标记为删除,以动态地减少公式中的子句数,提高后续验证的效率。

然而,近些年很多先进的 SAT 求解器在标准 CDCL 算法以外还会采用一些特殊的求解技术,例如有界变量增加(Bounded-Variable Addition, BVA^[51])、阻塞子句增加(Blocked-Clause Addition, BCA^[52])和扩展归结(Extended Resolution, ER^[53-54])等,通过这些技术产生的子句不能由 RUP 验证。为此,研究人员总结出归结非对称重言式(Reso-

lution Asymmetric Tautology, RAT) 的概念。目前, SAT 求解算法中使用的预处理、中处理和子句学习技术所产生的子句都可以通过 RAT 进行验证,也正是基于这一原理产生了 RAT 证明系统。对于每条待验证子句 C , RAT 首先使用 RUP 对其进行验证。如果未能通过,则设 C 的首个文字为 l , 并从公式 F 中选出含有 $\neg l$ 的子句构成集合 F' 。对于每个子句 $C' \in F'$, 首先对 C 和 C' 进行一步归结得到子句 R , 而后使用 RUP 对 R 进行验证。如果所有的 C' 都能通过验证, 则表示子句 C 通过验证, 将 C 加入 F 后继续验证下一条子句。同样地, 也有带删除信息的归结非对称重言式 (Deletion Resolution Asymmetric Tautology, DRAT) 证明系统, 其中以字符 “d” 开头的子句意义与 DRUP 相同。DRAT 的表达能力强于上述所有证明系统, 并且具有很好的兼容性, 因此目前最先进的 SAT 验证器一般均是基于 DRAT 证明系统构建, 例如工具 DRAT-trim^[55] 已被广泛用于国际 SAT 竞赛的结果验证。

5 挑战及未来趋势

自动推理技术的进步, 可以辅助数学家们发现新的结果并不断促进数学研究的进步, 同时数学家们精心构造的组合数学问题也为自动推理领域创造了更多的挑战并促进了自动推理技术的发展。目前, 组合数学领域还有众多的开放性问题有待解决, 并且其中一些问题恰好处在自动推理技术处理能力的边界地带, 比如现在的自动推理技术可以在几百秒内找到两个正交的 10 阶拉丁方, 但是对 3 个相互正交的拉丁方是否存在这一开放性问题却无能为力。未来在攻克这些开放性问题的同时, 新的自动推理技术也会应运而生。目前, 利用自动推理技术解决数学开放性难题有以下几个值得关注的方面:

(1) 证明生成及验证。自动推理工具理应被要求生成一个证明文件, 特别是对于不可满足的公式逻辑公式, 它应当能生成一个可以被验证的证明文件来保证其结果的可信性。目前, SAT 求解器已经可以基于 DRAT 生成证明文件, 但其证明文件规模巨大, 检查耗时依然是个问题。未来, 改进证明系统以压缩证明文件和提高验证效率为发展趋势。另外, 如何将这种 CDCL 框架下的证明系统应用于一阶逻辑推理过程, 以及如何产生人类易读的证明文件, 也是非常重要的发展方向。

(2) 并行推理方法。随着近年 GPU 和云计算的发展, 实现高效的并行自动推理算法和工具是非常有前景的方法。特别是对于组合数学难题, 简单的并行方法对求解效率的提升微乎其微, 尚未充分发挥并行计算的优势。因此, 如何在并行环境下分解问题, 以及各个进程之间如何交换冲突、推理信息以充分发挥并行优势, 是未来的又一研究趋势。

(3) 神经符号方法。传统的自动推理完全基于符号方法, 具有很强的演绎推理能力。然而, 现实中的很多问题往往都是基于特定背景, 为了提高求解这些问题的效率, 人们需要进行观察并提出合适的推理策略, 而现有的方法尚不具备这种归纳能力。近年来, 深度学习技术在多个领域极大地增强了人工智能系统的能力, 能否将其与符号方法结合, 以提高自动

推理的求解效率, 成为了国内外的研究热点。例如, 已有工作基于图神经网络等模型对 SAT 等问题进行表示学习, 预测其可满足性、变量赋值、决策优先级等属性^[56-57]。然而, 此类方法目前尚处于早期发展阶段, 如何更好地结合神经和符号方法的优势, 提出更加高效的求解方法, 仍有待进一步探索。

结束语 将自动推理这一计算机技术用于组合数学领域的开放性问题求解, 一直受到计算机领域和数学领域的关注。对近年来该领域的求解技术和发展进行了回顾与总结, 总结了该领域关注的主要数学问题与主流推理工具; 按求解模式将近些年出现的求解技术分为 3 种类型, 并介绍了其主要思想和方法优势; 还对自动推理技术处理数学难题时的可信性问题进行了介绍与总结, 指出了自动推理技术所面临的难点及未来研究方向, 这对于利用自动推理技术处理组合数学难题的研究与发展具有一定的借鉴意义。

参考文献

- [1] ROBINSON, ALAN J A, VORONKOV A, et al. Handbook of Automated Reasoning[M]. Elsevier, 2001: 17-18.
- [2] TURING A M. On computable numbers, with an application to the Entscheidungsproblem[J]. Journal of Mathematics, 1936, 42: 230-265.
- [3] COOK S A. The complexity of theorem-proving procedures [C] // Proceedings of the Third Annual ACM Symposium on Theory of Computing, 1971: 151-158.
- [4] ARMIN B, HEULE M, MAREN H V, et al. Handbook of satisfiability[M]. IOS Press, 2009.
- [5] SMOLENSKY P. Connectionist AI, symbolic AI, and the brain [J]. Artificial Intelligence Review, 1987, 1(2): 95-109.
- [6] KROENING D, STRICHMAN O. Decision procedures [M]. Berlin: Springer, 2016.
- [7] LUCKHAM D, ROBINSON J A. A Machine-Oriented Logic based on the Resolution Principle[J]. Journal of Symbolic Logic, 1966, 31(3): 515.
- [8] MOSKEWICZ M W, MADIGAN C F, ZHAO Y, et al. Chaff: Engineering an efficient SAT solver [C] // Proceedings of the 38th annual Design Automation Conference. 2001: 530-535.
- [9] BONACINA M P, GRAHAM-LENGRAND S, SHANKAR N. Satisfiability modulo theories and assignments [C] // International Conference on Automated Deduction. Cham: Springer, 2017: 42-59.
- [10] BOSE R C, SHRIKHANDE S, PARKER E T. Further results on the construction of mutually orthogonal Latin squares and the falsify of the Euler's conjecture [J]. Canadian Journal of Mathematics, 1960, 12: 189-203.
- [11] OSMUNDSEN J A. Major mathematical conjecture propounded 177 years ago is disproved: 3 mathematicians solve old puzzle [J]. The New York Times, 1959, 26: 1.
- [12] ZHANG J. Constructing finite algebras with FALCON [J]. Journal of automated reasoning, 1996, 17(1): 1-22.
- [13] HASEGAWA R, FUJITA H, KOSHIMURA M, et al. A model generation-based theorem prover MGTP for first-order logic

- [M]//Computational Logic: Logic Programming and Beyond. Berlin: Springer, 2002; 178-213.
- [14] FUJITA M, SLANEY J, BENNETT F. Automatic generation of some results in finite algebra[C]//IJCAI. 1993; 52-57.
- [15] SLANEY J. FINDER: Finite domain enumerator system description[C]// International Conference on Automated Deduction. Berlin: Springer, 1994; 798-801.
- [16] ZHANG H, STICKEL M. Implementing the davis-putnam method[J]. *Journal of Automated Reasoning*, 2000, 24(1): 277-296.
- [17] ZHANG H. SATO: An efficient propositional prover[C]// International Conference on Automated Deduction. Berlin: Springer, 1997.
- [18] MCCUNE W. MACE 2.0 reference manual and guide [J]. arXiv:0106042, 2001.
- [19] ZHANG J, ZHANG H. SEM: a system for enumerating models [C]//Proceedings of the 4th International Joint Conference on Artificial Intelligence. 1995; 298-303.
- [20] JIA X, ZHANG J. A powerful technique to eliminate isomorphism in finite model search[C]//International Joint Conference on Automated Reasoning. Berlin: Springer, 2006; 318-331.
- [21] MCCUNE W. Mace4 reference manual and guide [J]. arXiv: 0310055, 2003.
- [22] EÉN N, SÖRENSSON N. An extensible SAT-solver[C]//International conference on theory and applications of satisfiability testing. Berlin: Springer, 2003; 502-518.
- [23] AUDEMARD G, SIMON L. Predicting learnt clauses quality in modern SAT solvers[C]//Twenty-first International Joint Conference on Artificial Intelligence. 2009.
- [24] HEULE M J H, KULLMANN O, WIERINGA S, et al. Cube and conquer. Guiding CDCL SAT solvers by lookaheads[C]//Haifa Verification Conference. Berlin: Springer, 2011; 50-65.
- [25] LAMB E. Two-hundred-terabyte maths proof is largest ever [J]. *Nature*, 2016, 534(7605).
- [26] HEULE M. Schur number five[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2018.
- [27] BRAKENSIEK J, HEULE M, MACKEY J, et al. The Resolution of Keller's Conjecture[C]//Automated Reasoning: 10th International Joint Conference (IJCAR 2020). 2020; 48-65.
- [28] ZHANG J. Automatic symmetry breaking method combined with SAT[C]//Acm Symposium on Applied Computing. ACM, 2001.
- [29] HUANG P, MA F, GE C, et al. Investigating the existence of large sets of idempotent quasigroups via satisfiability testing [C]//International Joint Conference on Automated Reasoning. Cham: Springer, 2018; 354-369.
- [30] CHAR B W, FEE G J, GEDDES K O, et al. A tutorial introduction to Maple[J]. *Journal of Symbolic Computation*, 1986, 2(2): 179-200.
- [31] WOLFRAM S. The Mathematica Book[M]. Wolfram Research, Inc., 2003.
- [32] STEIN W. Sage: Open Source Mathematical Software[J/OL]. The Cloud, 2008. <https://www.williamstein.org/talks/2010-01-14-ams/2010-01-14-ams-sage/sage-slides.pdf>.
- [33] BOSMA W, CANNON J, PLAYOUST C. The Magma algebra system I: The user language[J]. *Journal of Symbolic Computation*, 1997, 24(3/4): 235-265.
- [34] BRIGHT C, GANESH V, HEINLE A, et al. MathCheck 2: A SAT+CAS verifier for combinatorial conjectures[C]//International Workshop on Computer Algebra in Scientific Computing. Cham: Springer, 2016; 117-133.
- [35] BRIGHT C, KOTSIREAS I, HEINLE A, et al. Enumeration of complex Golay pairs via programmatic SAT[C]//Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation. 2018; 111-118.
- [36] BRIGHT C, ĐOKOVIĆ D Ž, KOTSIREAS I, et al. A SAT + CAS approach to finding good matrices: New examples and counterexamples[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2019; 1435-1442.
- [37] BRIGHT C, CHEUNG K, STEVENS B, et al. Unsatisfiability Proofs for Weight 16 Codewords in Lam's Problem[C]//Twenty-Ninth International Joint Conference on Artificial Intelligence and Seventeenth Pacific Rim International Conference on Artificial Intelligence (IJCAI). 2020.
- [38] HUANG P, LIU M, GE C, et al. Investigating the existence of orthogonal golf designs via satisfiability testing [C]//Proceedings of the 2019 International Symposium on Symbolic and Algebraic Computation. 2019; 203-210.
- [39] MA F F, ZHANG J. Finding orthogonal latin squares using finite model searching tools [J]. *Science China Information Sciences*, 2013, 56(3): 1-9.
- [40] JIN J, LV Y, GE C, et al. Investigating the existence of costas latin squares via satisfiability testing [C]//International Conference on Theory and Applications of Satisfiability Testing. Cham: Springer, 2021; 270-279.
- [41] DAVIES A, VELIĆKOVIĆ P, BUESING L, et al. Advancing mathematics by guiding human intuition with AI[J]. *Nature*, 2021, 600(7887): 70-74.
- [42] AOUCHICHE M, HANSEN P. A survey of automated conjectures in spectral graph theory[J]. *Linear Algebra and Its Applications*, 2010, 432(9): 2293-2322.
- [43] AOUCHICHE M, HANSEN P. Proximity, remoteness and distance eigenvalues of a graph[J]. *Discrete Applied Mathematics*, 2016, 213: 17-25.
- [44] COLLINS K L. On a conjecture of Graham and Lovász about distance matrices[J]. *Discrete applied mathematics*, 1989, 25(1/2): 27-35.
- [45] HOGBEN L, REINHART C. Spectra of variants of distance matrices of graphs and digraphs: a survey [J]. *La Matematica*, 2022, 1(1): 186-224.
- [46] AARONSON J, GROENLAND C, GRZESIK A, et al. Exact hyperplane covers for subsets of the hypercube[J]. *Discrete Mathematics*, 2021, 344(9): 112490.
- [47] APPEL K, HAKEN W. The four-color proof suffices[J]. *The Mathematical Intelligencer*, 1986, 8(1): 10-20.

- [48] BRUMMAYER R, LONSING F, BIERE A. Automated testing and debugging of SAT and QBF solvers[C]//International Conference on Theory and Applications of Satisfiability Testing. Berlin:Springer,2010:44-57.
- [49] ZHANG L, MALIK S. Validating SAT solvers using an independent resolution-based checker, Practical implementations and other applications[C]//2003 Design, Automation and Test in Europe Conference and Exhibition. IEEE,2003:880-885.
- [50] VAN GELDER A. Verifying RUP Proofs of Propositional Unsatisfiability[C]//International Symposium on Artificial Intelligence and Mathematics. Springer,2008.
- [51] MANTHEY N, HEULE M J H, BIERE A. Automated reencoding of boolean formulas[C]//Haifa Verification Conference. Berlin:Springer,2012:102-117.
- [52] JÄRVISALO M, HEULE M J H, BIERE A. Inprocessing rules [C]//International Joint Conference on Automated Reasoning. Berlin:Springer,2012:355-370.
- [53] AUDEMARD G, KATSIRELOS G, SIMON L. A restriction of extended resolution for clause learning SAT solvers[C]//Twenty-Fourth AAAI Conference on Artificial Intelligence. 2010.
- [54] KULLMANN O. On a generalization of extended resolution[J]. Discrete Applied Mathematics,1999,96:149-176.
- [55] WETZLER N, HEULE M J H, HUNT W A. DRAT-trim: Efficient checking and trimming using expressive clausal proofs

[C]//International Conference on Theory and Applications of Satisfiability Testing. Cham:Springer,2014:422-429.

- [56] SELSAM D, LAMM M, BENEDIKT B, et al. Learning a SAT Solver from Single-Bit Supervision[C]//International Conference on Learning Representations. 2018.

- [57] YOLCU E, PÓCZOS B. Learning local search heuristics for boolean satisfiability[C]//Proceedings of the 33rd International Conference on Neural Information Processing Systems. 2019:7992-8003.



HUANG Pei, born in 1992, Ph.D. His main research interests include automated reasoning and trustworthy AI.



ZHANG Jian, born in 1969, Ph.D, professor, Ph.D supervisor, is a member of China Computer Federation. His main research interests include software engineering and automated reasoning.

(责任编辑:柯颖)