

文章编号: 2023-0850

支持复杂规范条文解译的基元函数提取与分析

逯静洲¹, 曹心瑜¹, 郑哲², 陆新征², 林佳瑞²

(1. 烟台大学土木工程学院, 山东烟台 264005; 2. 清华大学土木工程系, 土木工程安全与耐久教育部重点实验室, 北京 100084)

摘 要: 规则条文解译(Rule interpretation)是智能审图(Automated rule checking, ARC)面临的关键挑战之一, 目前依赖领域专家完成规则解译, 耗时耗力。其根本原因是规范条文包含大量复杂逻辑与领域隐含知识, 计算机缺乏相应的基本常识与逻辑规则, 因此难以有效理解和解译复杂设计规范条文。因此, 本研究以《建筑设计防火规范》关键内容为例, 通过系统整理、分析与拆解有关规范条文, 首次提出了基元函数概念, 以表征不同规范条文规则推理与逻辑计算过程共用的基础算法单元; 并面向复杂规范条文解析, 建立了有关基元函数的识别与提取方法, 识别、定义了存在性检查、数量检查、几何检查、距离检查、面积检查、空间位置检查、窗墙比检查、属性检查共 8 类 66 个基元函数, 构建了首个针对我国建筑结构设计审查的基元函数库。同时, 研究从审查对象以及基元函数复杂度、使用频次、共现特征与复杂条文表示能力等方面进行了详细分析, 结果表明(1) 基元函数使用频率呈长尾分布; (2) 属性类函数与其他各类函数都有较高的共现次数; (3) 所建立的基元函数具有强大的复杂逻辑表示能力, 可大幅提升复杂条文解译范围与推理能力。

关键词: 建筑规范; 基元函数; 文本分析; 规则解译; 自动合规性审图; 智能设计; 智能建造

中图分类号: TU17 **文献标志码:** A **doi:**

ATOMIC FUNCTION EXTRACTION AND ANALYSIS FOR INTERPRETING COMPLEX RULES FROM BUILDING CODES

LU Jing-zhou¹, CAO Xin-yu¹, ZHENG Zhe², LU Xin-zheng², LIN Jia-rui²(1. School of Civil Engineering, Yantai University, Yantai 264005, China;
2. Department of Civil Engineering, Tsinghua University, Beijing 100084, China)

Abstract: Automated rule checking (ARC) is revolutionizing in various domains, while its key stage, rule interpretation, still remains a bottleneck. Currently, the process relies heavily on domain experts, making it both time-consuming and labor-intensive. This challenge arises due to the complex computational logics and hidden domain knowledge embedded within the rules. These complexities are compounded by the computer's inability to inherently understand complex design rules, given its lack of basic common sense and logical rules. Therefore, our study takes the key contents of "Building Design Fire Code" as an example, and through systematically organizing, analyzing, and disassembling the relevant rules, the concept of "atomic function" is firstly proposed. This concept aims to characterize the basic algorithmic units shared by different rules when it comes to rule reasoning and logical computation. Then, for the analysis of complex rules, the method to identify and extract relevant atomic functions is established. As a result, we've identified 66 atomic functions, and further categorized them into 8 groups for the checking of existence, quantity, geometry, distance, area, spatial location, window-wall ratio, and property respectively, thereby forming the first atomic function database for the checking of Chinese building design codes. Finally, we scrutinize the complexity, utilization frequency, co-occurrence patterns, and complex logic representation capabilities of the checking objects and atomic functions, and the results

收稿日期: 2024-03-05; 修改日期:

基金项目: 国家自然科学基金(72091512, 52378306, 51978596)

通讯作者: 林佳瑞(1987—), 男, 山东人, 副研究员, 博士, 主要从事智能建造、数字孪生与数字防灾技术研究(E-mail: lin611@tsinghua.edu.cn)

作者简介: 逯静洲(1973—), 男, 山西人, 教授, 博士, 硕士生导师, 主要从事混凝土损伤特性研究(E-mail: lujingzhou@sina.com);

曹心瑜(1999—), 女, 山东人, 硕士生, 主要从事建筑信息化研究(E-mail: cccnel@163.com);

郑哲(1997—), 男, 四川人, 博士生, 主要从事建筑信息化研究(E-mail: zhengz19@mails.tsinghua.edu.cn);

陆新征(1973—), 男, 安徽人, 教授, 博士, 博士生导师, 主要从事数值模拟与防灾减灾研究(E-mail: luxz@tsinghua.edu.cn);

demonstrate: (1) Atomic functions' usage showcases a long-tailed distribution. (2) Property functions frequently co-occur with other function types. (3) The established atomic functions have a strong capacity for representing complex logics, which can greatly improve the interpretation scope and reasoning ability of complex design rules.

Key words: Building Codes; atomic functions; text analysis; rule interpretation; automated rule checking (ARC); intelligent design; intelligent construction

建筑结构规范是建筑设计、施工乃至运维必须遵守的基本要求。设计图纸或模型的审查,或审图,是复核有关设计是否满足规范要求的关键环节^[1-2]。而人工图纸审查的模式在规范性、时效性、全面性等方面面临诸多挑战。在行业发展要求和国家政策的推动下,自动设计审查技术成为近几年的研究热点^[3]。

自动设计审查通常包括四个阶段:(1)规则的解译和逻辑表达;(2)模型准备;(3)规则提取;(4)输出审查报告^[4]。其中,规则解译旨在解释和理解规则、标准或招标书等文本中的要求和规定,以使计算机理解自然语言表达的条文,并实现设计模型或图纸实体的自动识别与校核。建筑结构工程领域国家规范和行业标准等均由自然语言编写而成,其条文包含大量领域知识和复杂逻辑。计算机由于缺乏相应的领域基本常识和逻辑推理能力,对自然语言表达的条文和领域隐含知识的理解能力有限,无法对规范条文进行准确解译。因此,如何让计算机自动理解相关条文已成为智能审图乃至智能设计、建造领域亟待解决的关键难题。

针对以上问题,本文通过分析 IFC(Industry Foundation Class)标准^[5]和《建筑信息模型语义字典(征求意见稿)》^[6]等资料,面向规范条文规则推理与逻辑表达需求,首次提出了基元函数概念。并通过消防规范有关设计条文的解译分析,构建了首个针对我国建筑结构规范的基元函数库。最后,研究从审查对象以及基元函数复杂度、使用频次、共现特征与复杂条文表示能力等方面对基元函数库进行了详细分析。结果表明,本文提出的基元函数库可以有效提升规范条文解译的质量和效率,从而使自动规则解译的应用范围进一步扩大,以支持更快速的规则库开发与智能审查。

1 规范条文解译研究现状

1.1 人工规则解译研究现状

目前,业界最常用的规则解译方法主要通过领域专家逐条解释规范,并由开发者将规范转化为计

算机代码嵌入到审查引擎中^[7-8]。该方式高度依赖领域专家经验及其与开发者的高效协同,规则库开发效率较低^[9]。此外,此类方法难以根据规范的更迭或用户的需求及时修改和调整^[10],用户只能使用规则库中的既有规则,无法自定义审查条文^[11]。

1.2 半自动规则解译

半自动规则解译方法旨在通过一定的自动化手段对规范语义或语句进行初步解析或标记来降低规范解译的成本。

以 RASE (Requirement, Applicability, Selection, Exception)方法^[12]为例,该方法引入四种运算符的标记辅助规范文本标注,以提升逻辑语句生成效率;孙澄宇等人通过文本分类将条文分为可数字化条文与不可数字化条文,以辅助可数字化条文计算机解译^[9];半自动解译的方法具有较好的可解释性,推动了 ARC(Automated rule checking)的发展,但仍需要大量的手工标注作为应用基础,自动化程度不高。

1.3 自动规则解译方法

NLP 等人工智能技术的引入大幅降低人工条文解译的成本,让自动规则解译成为可能^[13]。例如,Bimchecker 利用 NLP 技术从规范文本中自动抽取领域知识,实现了规范条文逻辑的结构化及快速解译^[14]。此外,Zhang 等人基于 NLP、Express 数据处理方法、逻辑推理法等技术提出了自动规则解译与规则检查方法^[15],Guo 等人利用 NLP 提取规范中信息和逻辑并自动生成 SPARQL 语句^[16],Zhou 等人提出了基于深度预训练模型的规则解译方法,可以将复杂规范条文自动转换为规则检查树(RCTree)^[17]。在此基础上,Zheng 等人提出了一种基于无监督学习的语义对齐方法和基于知识的冲突消解方法以进一步提高规则解译的准确性^[18]。

1.4 规则解译研究难点

有关文献表明,规则解译过程通常可划分为预处理、审查对象识别、计算逻辑提取与约束条件提取四个步骤,如图 1 所示。

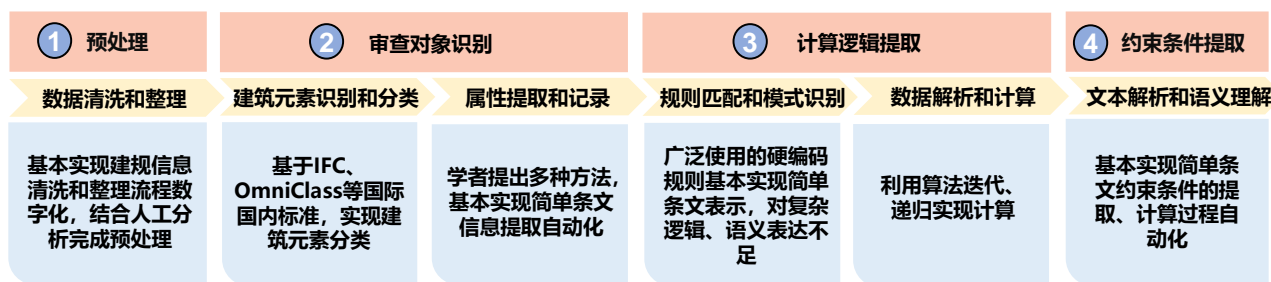


图1 规则解译流程图

Fig.1 The flowchart of rule interpretation

力。

在预处理阶段, 计算机需要对收集到的建筑结构规范、技术文件等信息进行清洗和整理, 以确保数据的规范化保存; 识别审查对象阶段中, 计算机需遍历整合后的规范条文文本, 并从中识别出以名词形式出现的各类建筑元素及其属性, 即为规则解译的审查对象; 在提取计算逻辑阶段, 利用一阶逻辑表示方法或 NLP 等技术将规范和准则中的计算逻辑进行提取, 并确定相应的计算参数与计算方法等, 该过程条文中复杂计算逻辑提取困难是当前规则解译面临的关键挑战; 最后提取约束条件阶段, 即为对规范中约束条件所涉及的属性、要素和数值等的限制规则进行提取。

尽管既有基于 NLP 的自动规则解译方法提高了规则解译的效率、准确性和自动化程度^[3]。但当前规则表示通常采用一阶逻辑表示方法, 对条文内容的表达能力有限, 仅能表示条文中的简单逻辑, 难以支持领域隐式属性的准确提取和计算。然而, 规范条文中存在大量结构复杂的长难句, 且不同条文之间往往存在大量具有类似计算逻辑的部分, 需要依赖更为强大的逻辑表示法来进行提取。

针对条文复杂逻辑表达问题, 既有研究引入了 DSL (domain-specific language) 和高阶函数库两种方法。围绕 DSL 方法, Sydora 等人基于 DSL 实现了规范强条解译与 BIM 模型评估、检查^[19]; Solihin 等人提出了基于 BIMRL 的简化 DSL 方法, 支持使用空间运算符对模型几何信息进行查询和校核^[20-21]。有关 DSL 方法探索了复杂计算逻辑的表达, 但往往与数据模型(IFC、IFCOWL 等)深度绑定、难以迁移与扩展, 同时其学习应用成本较高, 因此应用范围较局限。此外, 韩国团队探索了基于高阶函数的规则构建方法, Uhm^[22]和 Lee 等人^[23]提出了一系列高阶函数, 用于拓扑关系检查、几何检查、实体位置关系等检查, 体现出较好的复杂逻辑表示能

综上所述, 当前复杂条文及规则解析仍面临以下挑战:

(1) 既有方法只能描述简单条文, 难以对内容、结构复杂的条文进行准确性表达^[24], 其表达能力有限;

(2) 对函数库的探索刚刚起步, 亟需探索抽取与构建面向我国建筑结构设计规范函数库, 支持复杂条文解译。

2 条文拆解与基元函数提取

根据前期研究, 建筑结构规范条文往往隐含大量领域知识和复杂计算逻辑, 并具有大量共性的计算逻辑单元。

例如, 表格 5.2.2 中提到“建筑类别为一、二级的高层民用建筑之间的防火间距不大于 13m;”

例如, 条文 4.2.3 中提到“甲、乙类液体立式储罐之间的防火间距不应小于 2m。”

上述两条规范条文均需检查对象间的防火间距, 因此可将类似共性的计算逻辑进行统一抽象, 定义为计算函数。受此启发, 本研究提出将此类共性的计算逻辑单元定义为基元函数, 作为不同规范条文规则解析的基本逻辑单元。在此基础上, 进一步明确相应的识别、定义与分类方法, 以构建复杂结构审查与计算分析的基元函数库, 具体流程如图 2 所示。

基元函数提取旨在发现不同规范条文中具有共性的计算逻辑单元, 经过分析、整理后形成基元函数库。识别与提取基元函数的过程包含以下几个步骤: (1) 数据搜集与条文预处理(2.1 节), 旨在搜集规范并进行条文转化与过滤, 将规范中计算机可解译的表格与复杂的句子拆分为能够完整表达要求的短句; (2) 审查对象提取和基元函数识别(2.2

节),旨在对条文进行分析,并对条文中的审查对象和基元函数进行提取;(3)基元函数分类(2.3节),旨在根据基元函数的复杂度、检查类型等对其进行分类和整理,从而建立针对我国建筑规范的基元函数库。

在构建上述函数库的基础上,本文对条文与函数进行了进一步的统计分析,包括审查对象以及基元函数复杂度、使用频次、共现特征与复杂条文的

表示能力分析,旨在寻找基元函数在规范条文间的使用规律,以此来分析基元函数对条文中复杂逻辑的表达程度。

本文以《建筑设计防火规范》(GB 50016-2014)^[25]为例展示所提出的方法。通过搜集分析《建筑设计防火规范》(GB 50016-2014)^[25]中的三、四、五章的条文和表格信息,来实现本文提出的框架。

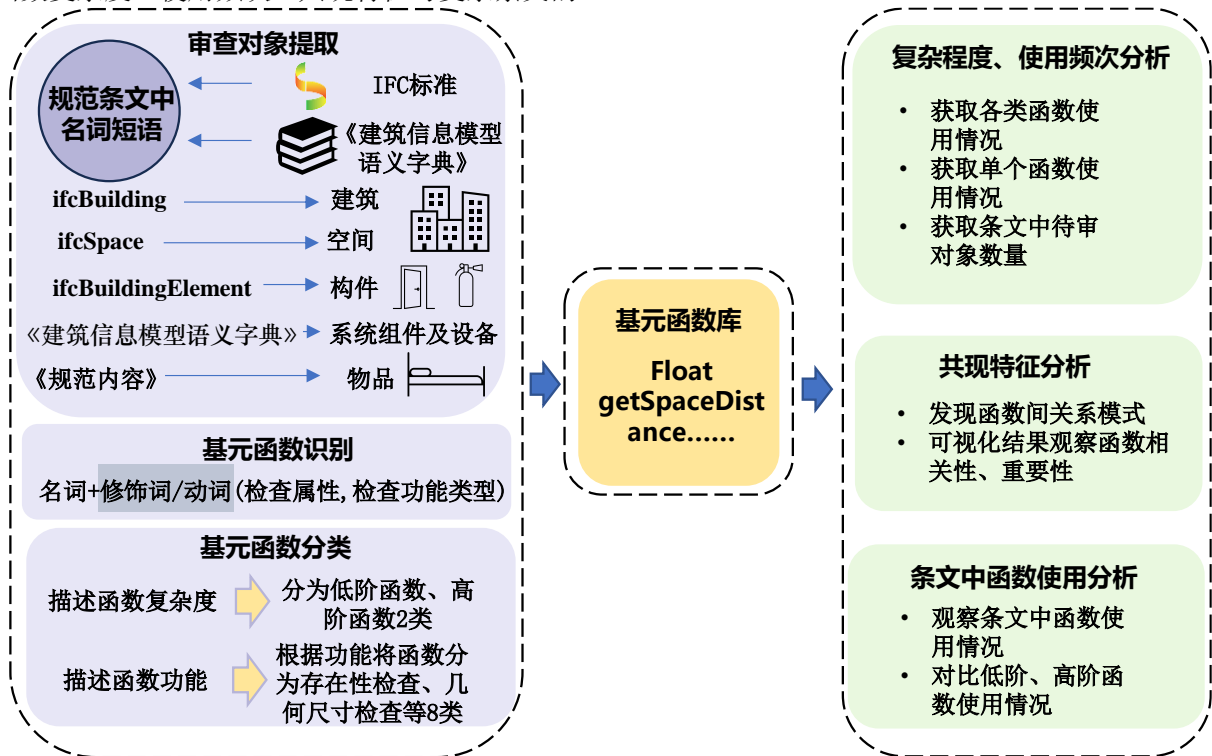


图2 研究流程图

Fig.2 The workflow of this study

2.1 数据搜集与条文预处理

2.1.1 条文数据收集和整理

火灾作为影响城市的“风险源”之一^[26],是建筑设计需要考虑的重要因素。《建筑设计防火规范》(GB 50016-2014)^[25]适用于各类新建、扩建和改造的建筑,其涵盖了多样的建筑类型、丰富的建筑构件、多样且复杂的建筑要求(如构件的几何要求、可见性等),且规范中描述了厂房仓库、可燃物堆场、民用建筑等不同场景下的防火要求。

因此,选取《建筑设计防火规范》(GB 50016-2014)^[25]用于分析研究,可以建立较为全面的基元函数库。本文所选取的章节为三、四、五章,第三章为仓库和厂房的相关规定,包括但不限于耐火等级、防火间距和安全疏散等问题;第四章为甲、乙、丙类液体、气体储罐(区)和可燃材料堆场的

相关规定,多集中为防火间距问题;而第五章则以民用建筑为审查对象,对其平面布局、防火分区和疏散避难等相关问题进行了防火层面上的规定。

2.1.2 条文预处理

由于规范中的单个条文可能包含多个针对不同情形的设计要求子句,其结构过于复杂、不利于分析与解译。因此,为使计算机能够从逻辑上判断一条语句的结构,宜将条文拆分成具备完整的检查对象、检查条件和检查要求的短句,从而便于计算机对各成分进行分析、提取和解译^[22]。此外,为扩大有关条文数量,本研究通过对审查对象及其属性限制条件的整合梳理,将规范中的表格内容转化为文本形式的格式化条文。

以《建筑设计防火规范》中表格 3.2.1 为例,将其中内容转化为文本形式,格式为: x 级耐火等

级的 x 构件的燃烧性能为 x ，且其耐火极限不应低于 $x(h)$ 。

构件名称		耐火等级			
		一级	二级	三级	四级
墙	防火墙	不燃性 3.00	不燃性 3.00	不燃性 3.00	不燃性 3.00
	承重墙	不燃性 3.00	不燃性 2.50	不燃性 2.00	难燃性 0.50
	楼梯间和前室的墙 电梯井的墙	不燃性 2.00	不燃性 2.00	不燃性 1.50	难燃性 0.50
	疏散走道两侧的隔墙	不燃性 1.00	不燃性 1.00	不燃性 0.50	难燃性 0.25
	非承重外墙房间隔墙	不燃性 0.75	不燃性 0.50	难燃性 0.50	难燃性 0.25

表 1 《建筑设计防火规范》(GB 50016-2014)表 3.2.1 (部分)

Table1 《Code for fire protection design of buildings》(GB 50016-2014) table3.2.1 (partial)

1.一级耐火等级的厂房和仓库墙构件中防火墙的燃烧性能为不燃性，耐火极限不低于 3.0h。

2.二级耐火等级的厂房和仓库墙构件中承重墙的燃烧性能为不燃性，耐火极限不低于 2.0h。

.....

2.1.3 条文过滤

在上述整理成短句的条文中，只能选取具备规范检查意义和计算机可处理条文可以用来进一步分析。因此，需要首先定义计算机不可处理条文，并在此基础上对条文进行过滤。本文将计算机不可处理的情况包括以下三种：(1) 定义类条文，即句子中没有直接审查要求的短句，只起到介绍工程对象特性等作用；(2) 定性要求条文，即包含模糊性设计要求的短句^[22]，该类句子没有明确的检查标准，需要由设计师根据经验对设计图纸进行评估，以确认合理与否；(3) 外部参考性条文：短句中包含需要从其他规范章节间引用信息的句子，包括章节间的相互引用，和一个大条件下需要引用多个相关约束的句子。基于以上分类，本研究基于团队此前条文自动分类算法^[27]对规范条文进行了自动分析和分类，以辅助识别计算机可处理的条文。

2.2 审查对象提取和基元函数识别

2.2.1 审查对象提取

本文参考 IFC 标准^[5]和《建筑信息模型语义字典(征求意见稿)》^[6]，将条文中名词即建筑元素分为五类：建筑类、空间类、构件类、系统组件及设备类和其他物品类。

分类根据如下，本文定义的建筑等价于 IFC 中的 ifcBuilding，空间和构件等价于 IFC 中的 ifcSpace 和 ifcBuildingElement；建筑中的各类机电设备、家具、功能性系统（喷水灭火系统等），在 IFC 中没有能够直接完全对应的分类。因此，参考《建筑信息模型语义字典》的定义，本文采用系统组件及设备类名词定义上述系列概念。此外，因为厂房等系

列条文中涉及大量生产、储存所用到的原材料（可燃气体、液体等），本文还定义了物品类来描述上述概念。在后续分析条文的过程中，涉及到的审查对象都属于上述五个类型。下文表 2 总结了防火规范中名词类型的分布。

2.2.2 基元函数识别

由于建筑规范中多数关键词具有领域专用性，如耐火极限、火灾危险性等级等，计算机难以对其进行准确分割并理解，其含义也因国家而异。并且关键词所表达的含义也具有多样性，如相邻、建筑间距离等，计算机难以对其所表达语义正确解译并准确定位，故基元函数的识别需要领域专家在原始条文基础上进行人工解译，从而所使识别出的函数功能与短句中修饰词、动词所表达的含义相同。

具体的，在确定了审查对象后，需要继续获取审查对象的属性要求及逻辑关系，即提取短句句法。进而，基元函数的识别可根据规范短句中具备一定审查功能的关键词所表达的语义信息进行理解、归纳、总结，以形成一系列功能函数，并通过共性特征识别基元函数。

首先是条文中功能函数的识别。比如当条文中出现“距离”、“间距”等词时，可以人工识别出该条文是对审查对象距离的检查，从而得到一系列距离检查函数；当条文中出现“面积”、“建筑面积”时，可以从条文中识别出对审查对象面积要求的规定，得到面积检查函数；类似的，当条文中出现“邻近”、“连通”、“通向”等关键词时，根据关键词在条文中的语义便可以识别出对应的相邻、连通关系检查函数等等。

对上述识别出的功能函数进行罗列，可以发现不同条文审查时所需的函数具有共性。不同关键词识别出的函数或针对不同对象审查的函数可能具有相似的审查内容、相同的审查逻辑，针对这种具

有共性特征的功能函数,可归纳提取形成基元函数。例如,针对条文1“当住宅建筑疏散楼梯的户门不能通至屋面时,应设置2个安全出口”及条文2“住宅建筑单元之间的疏散楼梯应能通过屋面连通”,尽管“通至”“连通”两词在条文中用词形式不同,但均需获取与检查对象所连通的空间,因此,可归纳为基元函数 `isAccessibleWith(element a,direction b,type c)`,用来对条文中具备该含义的内容进行计算。

类似的,可针对 2.2.1 节五类审查对象,逐一识别定义各基元函数。

2.2.3 基元函数定义

函数名称采用驼峰命名法定义。具体的,函数名称以动词开头,动词后一般是名词或形容词来对函数功能进行描述。动词一般包括 `get`, `is`, `has` 三类,其中 `get` 开头的函数返回一系列待审对象、字符串或者数值等类型;`is` 开头的函数主要用于判断对象是否属于特定类型,并返回布尔值;`has` 开头的函数则判断对象是否由特定对象构成,并返回布尔值。最后,函数的形参包括审查对象的具体名称(`obj`)和以何种类型被检查(`type`)。

例如基元函数 `Float getSpaceDistance(space a, space b, type c)`,表示获取两个空间的距离,并返回值是数值型(`float`)。函数的形参是两个空间对象 `a`、`b`, `type` 表示距离的测量方式。距离测量方式包括门间距离、最远距离、最近距离和疏散距离等方式。

2.3 基元函数分类

针对函数审查内容的复杂程度,基元函数分为低阶函数和高阶函数两类。低阶函数是具有较为基础和具体功能的函数,主要用于处理能够直接从条文或模型中直接获取的属性信息,如建筑的用途属性、构件的材料属性等。因此,低阶函数往往只能用于表达条文中的简单语义信息,与一阶逻辑描述等价,其复杂逻辑表征能力有限。而高阶函数是在低阶函数的基础上构建的能够实现更为抽象和复杂功能的函数,旨在对空间距离、联通关系、机理特征等复杂计算逻辑进行封装和表征,支持根据相应的审图需求构建自定义的审图逻辑,更好支持复杂条文的解析能力。

例如,对条文“医院和疗养院的病房楼内相邻护理单元之间应采用耐火极限不低于 2.00h 的防火分隔墙”,如果仅考虑低阶函数,只能确定病房楼的用途、墙构件耐火等级等属性,而无法表示病房

楼与护理单元的存在性关系、相邻关系等,因此,只有同时考虑低、高阶两类基元函数,才能实现计算机对条文的准确解译和进一步审查。

此外,本文还进一步按照所实现的功能类别对基元函数进行了划分,根据各函数在检查对象、检查规则及计算方式上的不同,分为属性检查、空间位置检查、存在性检查、数量检查、几何尺寸检查、距离检查、窗墙比检查和面积检查 8 类,如下所示:

(1) 属性检查:属性检查类函数可以用于检查建筑元素的多种属性,例如获取墙构件的材料属性、房间的类型等。

(2) 空间位置检查:空间位置检查函数主要用于确定建筑各元素间的位置及关系,例如空间可见性、相邻等关系的检查等。

(3) 存在性检查:存在性检查函数用于检查建筑元素是否存在于给定的空间中,例如空间是否包含设备及系统组件等。

(4) 数量检查:数量检查函数可以统计建筑元素的数量,例如获取建筑中某类空间的数量等。

(5) 几何尺寸检查:几何尺寸检查类函数用于执行建筑元素之间的简单几何尺寸检查,例如获取空间的高度、宽度等。

(6) 距离检查:距离检查类函数用于计算建筑元素间的距离或与给定位置之间的距离,例如计算两个安全出口之间的距离、获取建筑间防火间距等。

(7) 窗墙比检查:窗墙比检查类函数用于检查建筑物中窗户与墙壁之间的比例关系,可以用于验证窗户设计是否符合建筑规范,是否满足通风、采光的要求。

(8) 面积检查:面积检查类函数用于计算建筑元素的面积并进行相应的检查,例如计算房间、楼层或建筑的面积,并检查是否符合相关建筑规范的要求。

基于以上过程,本研究共针对 5 类审查对象,梳理了 8 类 66 个基元函数。附录 1 展示了各函数的类别、返回值、函数名称以及函数对应的描述。其中高阶函数 61 个、低阶函数 5 个(蓝色)。

3 规范条文分析结果

为验证以上成果可行性及有效性,本节通过一系列分析和统计对基元函数特点及其在增强规范条文自动解译能力方面的价值进行了分析。

3.1 计算机可解译条文统计

对于《建筑设计防火规范》(GB 50016-2014)^[25]的三、四、五章内容,本文共搜集到文本形式条文451条,表格形式条文960条,共1411条;再将文本形式条文拆为679条,表格形式条文拆为1048条,最终共1727条;随后经条文过滤,共去除235条不可解译条文,剩余可解译条文1492条。

3.2 审查对象类型及其数量统计

表2为上述条文中审查对象的数量及其分布。其中,第三章的检查对象主要为建筑类(45.86%),其次是空间类(29.17%)的名词,对于其他三个层次名词的检查则相对较少;第四章的检查对象同样主要集中在建筑类(48.35%)上,但与第三章不同的是,第四章对系统组件及设备类名词(30.5%)的检查较多,其他三类的名词检查较少;第五章则侧重于空间类和构件类名词的检查,分别占比为30.57%和37.51%,其次是针对建筑类名词(24.27%)的检查,其余两类的名词检查较少。可见,由于各章的检查重点不同,章节内条文所针对的检查对象也不同。

表2 审查对象分布统计表

Table2 Distribution of object types

空间		构件		设备及组件		建筑		物品	
/	%	/	%	/	%	/	%	/	%
528	29.1%	256	14.1%	48	2.6%	830	45.9%	148	8.2%
201	13.8%	39	2.7%	443	30.5%	703	48.3%	68	4.7%
568	30.6%	697	37.5%	128	6.9%	451	24.3%	14	0.8%

对于条文中审查对象的数量,如图4所示。其中,80%的条文均包含至少2个审查对象;而只有20%的条涉及单个审查对象。由此可见,规范条文对多个审查对象的逻辑关系高度关注,对复杂条文的解译需求较高。

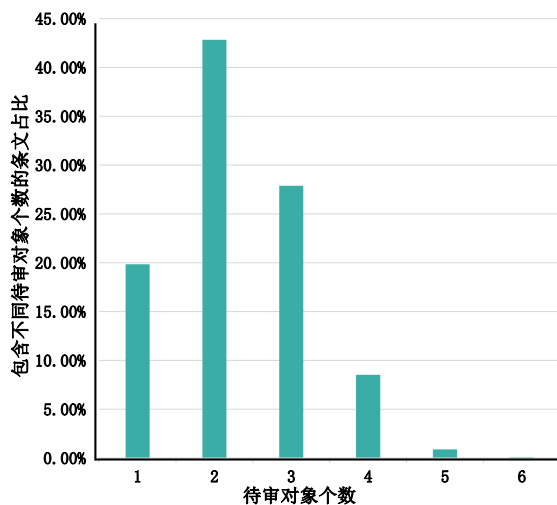


图4 包含不同待审对象数量的条文占比

Fig.4 Percentage of regulatory rules containing different numbers of checking objects

3.3 条文可解译性分析

本文进一步统计了仅采用低阶函数就可以解译的条文的占比。如图5所示,仅采用低阶函数可进行计算机解译的条文仅占总可解译条文的7.77%,表明单纯自动解译简单条文难以满足未来智能审图需求。而采用本文定义的基元函数库(包括高阶和低阶)可以基本实现剩余92.23%条文的自动解译,具备《建筑设计防火规范》3、4、5章接近100%的条文表示能力。因此,构建基元函数库对实现所有规范条文的计算机解译意义重大。

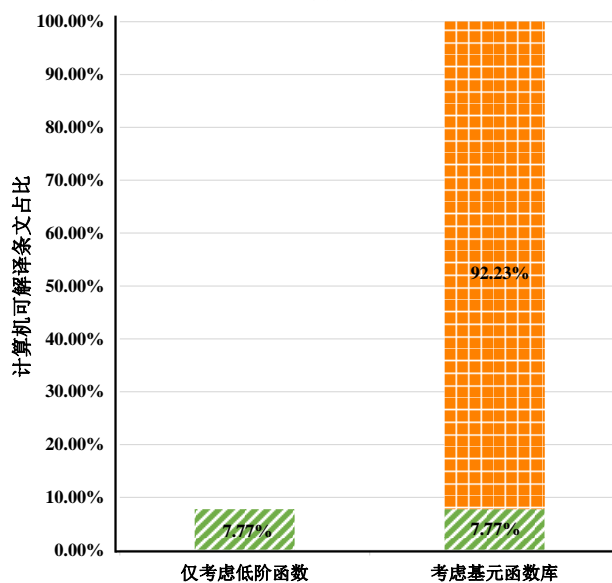


图5 计算机可解译的规范条文占比

Fig.5 Percentage of regulatory rules that can be interpreted by computers

3.4 基元函数使用频率统计

本文进一步统计了函数库中函数用于条文解译过程的使用频率分布,如图6所示。考虑篇幅限制,仅展示使用频次大于10函数的分布情况,红色表示低阶函数,紫色表示高阶函数。

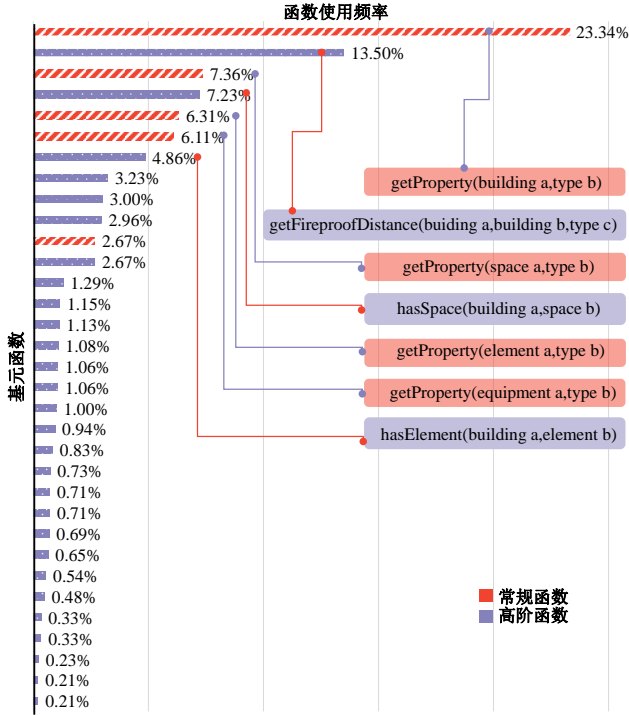


图6 基元函数使用频次统计图

Fig.6 Usage distribution of atomic functions

可以看出基元函数的使用情况呈现长尾分布。使用最多的五个函数中三个为低阶函数(getProperty(buiding a,type b), getProperty(element a,type b), getProperty(element a,type b)),两个是高阶函数(getFireproofDistance(buiding a,buiding b,type c)和 hasSpace(buiding a,space b))。在高阶函数中,最常用的五个函数为(getFireproofDistance(buiding a,buiding b,type c), hasSpace(buiding a,space b), hasElement(buiding a,element b), hasElement(space a,element b), hasEquipment(buiding a,equipment b))。

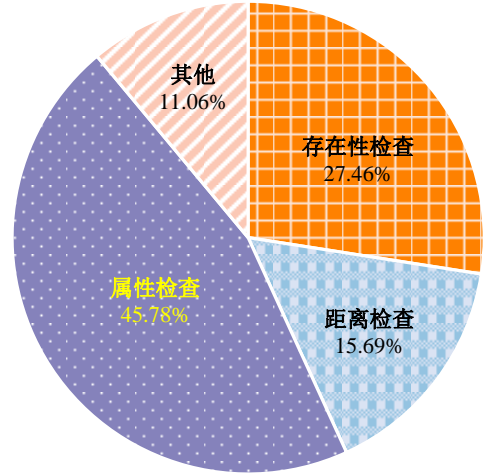


图7 各类函数使用频率统计图

Fig.7 Usage proportion of functions of different categories

因为获取检查对象的属性是最常用的操作,因此属性检查类的函数使用频率最高,如图7。此外,在高阶函数的使用中,条文中对检查对象存在性逻辑的需求最为频繁,因此,与其他功能性函数相比,存在性检查类的函数使用频率最高。

3.5 基元函数共现分析

本研究同时对有关函数在条文解译过程中的共现关系进行了分析,以发现函数间的关联模式(图8),以更好地指导未来审图规则库研发及应用。图8中的节点表示基元函数,节点间的连线表示两函数的共现情况。节点越大表示该函数出现的次数越多,连线越粗表示共现的次数越多。

由图8可知,在函数大类上,属性类与其他类型函数间的节点连线较粗,表明属性类函数与其他类型函数间存在大量共现情况,因此需将其作为基础类函数进行优先开发。而从单个函数的角度看,同类待审对象的函数间共现次数较多,如获取防火间距与获取建筑属性函数共现次数较多,因其审查对象都为建筑类。从条文逻辑上来看,空间往往存在于建筑内部,因此,获取空间属性与获取建筑属性的共现次数也较高。属性间函数、属性与存在性检查类函数间也有较多共现情况的发生,可以发现共现关系的分析与上述使用频率的统计具有相似的趋势,即使用频率高的函数之间同时具有较高的共现次数。

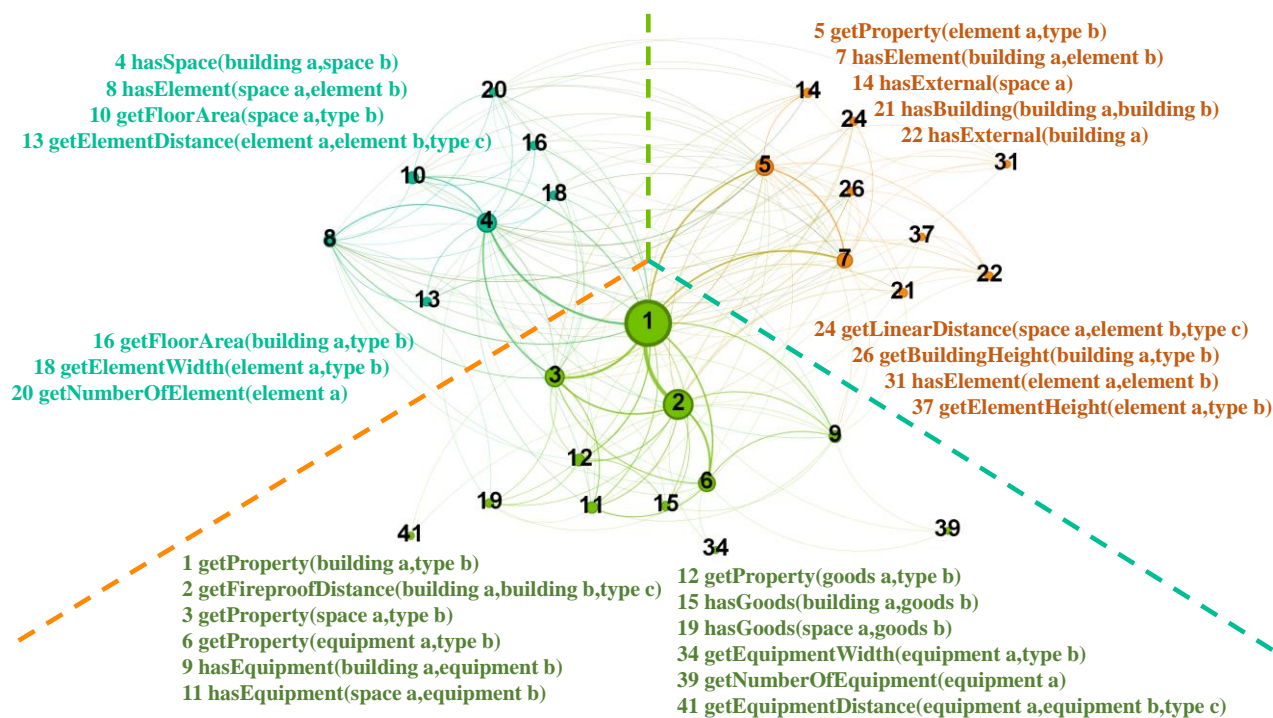


图8 函数共现关系网络图

Fig.8 Co-occurrence relationship of atomic functions

对共现次数较多的基元函数,与上述函数使用频次的分析相结合,可以为计算机进行规范函数识别提供一定参考。如 hasExternal(space a)函数的总体使用率只有 1.15%,但通过聚类分析图可知,其 92%的使用情形都与 getProperty(element a,type b)函数共同使用,故在工程师开发程序或应用时可以着重关注共现和使用频率的分析结果,对共现次数多、联系紧密的函数进行同时开发与实现。同时,在开发规范条文自动解译算法时,可根据函数的共现关系匹配相关基元函数,共现频率越高的函数越有可能在解析同一规范条文时出现。

3.6 规范条文中基元函数的使用情况分析

如图 9 所示,当条文解译所需的函数数量小于 3 时,高低阶函数的使用情况基本各占一半。对于较为复杂的规范条文,解译需要的基元函数大于 3 时,高阶函数的使用数量大于低阶函数。因此,含有复杂语义、领域知识的条文解译往往要依赖高阶函数来实现。

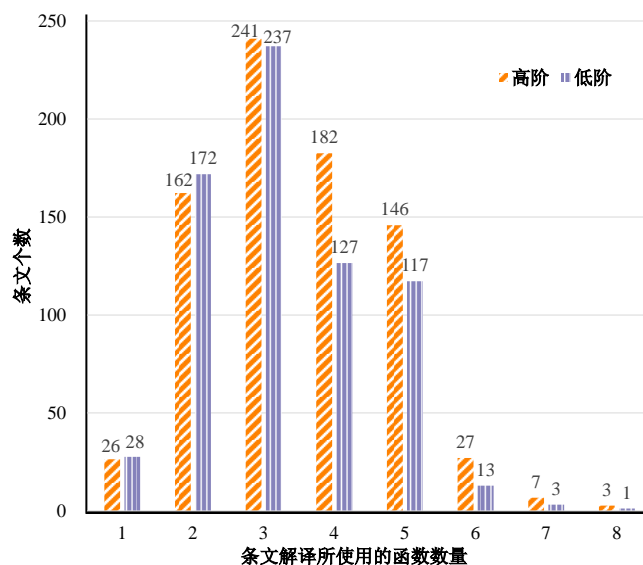


图9 条文解译所使用的函数数量统计

Fig.9 Numbers of atomic functions used for rule interpretation

根据上述规范条文与函数使用情况的分析可以得到以下结论:

(1) 有 20%左右的条文只包含一个审查对象,但仅使用单个函数进行检查的条文数量又较少,因此,针对同一审查对象往往会有多项审核需求。

(2) 有 93%左右的条文包含 2~5 个函数,而有 85%左右的条文包含 1~3 个高阶函数,对比可得,

低阶函数即属性类检查函数基本出现在每个条文的解译中,为基础函数。这与上述的使用频率分析、共现分析得出的结论基本相同。

(3) 仅有 8%左右的条文不包含高阶函数,剩余 92%的条文都需要至少一个高阶函数来完成复杂逻辑的解译,因此,本文提出的基元函数库中高阶函数的部分不可缺少。

(4) 综上所述,在本文提出针对我国规范的基元函数库中,低阶函数为条文解译提供基本支持,引入高阶函数可以更好支持复杂规则解译与审查。低、高阶函数的结合可有效提升条文解译的范围。

4 结论

本文面向规范条文自动解译,首先通过分析《建筑设计防火规范》(GB 50016-2014)^[25]中三、四、五章规范内容,提取整理了 1492 条建筑结构设计相关的条文;其次,通过相关条文的拆解、分析,首次引入基元函数的概念,作为审图规则的基础单元,用于表示不同规范条文之间共性的规则推理和复杂逻辑计算过程;同时,研究建立了相应基元函数的识别和提取方法,并通过系列分析验证了基元函数在复杂规范条文解译方面的价值。

基于上述方法,本研究共识别并定义了 8 类 66 个基元函数,形成了针对我国建筑结构设计审查的首个基元函数库,覆盖存在性检查、数量检查、几何检查、距离检查、面积检查、空间位置检查、围墙比检查和属性检查等不同审查需求。对函数使用频率、共现关系等的分析表明,本文提出的基元函数库有效提升了我国建筑规范复杂条文的表征与解译能力,可基本实现计算机可处理条文的 100% 解析。

研究表明,本研究提出的方法普适性较好,可推广至其他规范的基元函数提取过程,且本文构建的基元函数库也可用于解译其他有关规范。本研究同时为规则解译过程的自动化及函数集构建方向提供了指导。未来,在大规模建筑领域语料库和特定领域预训练模型的基础上^[28],可将该方法与 LLM (Large Language Model) 等技术结合实现复杂规则的自动解析^[29-30],实现我国设计规范的完全数字化解析与计算机自动推理技术,推动智能设计发展。

参考文献:

[1] 林佳瑞,周育丞,郑哲,等. 自动审图及智能审图研究与应用综述[J]. 工程力学, 2023, 40(07): 25-38.

LIN J R, ZHOU Y C, ZHENG Z, et al. RESEARCH AND APPLICATION OF INTELLIGENT DESIGN REVIEW [J]. ENGINEERING MECHANICS, 2023, 40(07): 25-38. (in Chinese)

[2] Nawari, Nawari O. Building information modeling: Automated code checking and compliance processes [OL]. CRC Press, 2018.

[3] 陆新征,廖文杰,顾栋炼,等. 从基于模拟到基于人工智能的建筑结构设计方法研究进展[J]. 工程力学, 2023, 40: 1-18.

LU X Z, LIAO W J, GU D L, et al. RESEARCH PROGRESS ON BUILDING STRUCTURAL DESIGN METHODS: FROM SIMULATION-BASED TO ARTIFICIAL INTELLIGENCE-BASED [J]. ENGINEERING MECHANICS, 2023, 40: 1-18. (in Chinese)

[4] EASTMAN C, LEE J M, JEONG Y S, et al. Automatic rule-based checking of building designs [J]. Automation in Construction, 2009, 18(8): 1011-1033.

[5] International Organization for Standardization. (2018). Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries (ISO 16739-1:2018)

[6] Housing and Construction Bureau of Shenzhen (2023). Standard for building information modeling semantic data dictionary (building fascicule) (SJGXXX-2023). (in Chinese)

[7] HOSSAIN M M, AHMED S. Developing an automated safety checking system using BIM: a case study in the Bangladeshi construction industry [J]. International Journal of Construction Management, 2022, 22(7): 1206-1224.

[8] 周逸苇,王广斌,曹冬平. 基于 BIM 的自动合规性审查研究综述 [J]. 土木工程学报. <https://doi.org/10.15951/j.tmgcxb.22121210>

ZHOU Y W, WANG G B, CAO D S. A Review on BIM-based Automated Compliance Checking [OL]. <https://doi.org/10.15951/j.tmgcxb.22121210> (in Chinese)

[9] 孙澄宇,柯勋. 建筑设计中 BIM 模型的自动规范检查方法研究[J]. 建筑科学, 2016, 32(4): 140-145.

- SUN C Y, KE X. Method of Automatic Design Code Compliance Checking for BIM Models [J]. BUILDING SCIENCE, 2016, 32(4): 140-145. (in Chinese)
- [10] XU R, SOLIHIN W, HUANG Z Y. Code Checking and Visualization of an Architecture Design [OL]. (2004-10).
- [11] PREIDEL C, BORRMANN A. Automated Code Compliance Checking Based on a Visual Language and Building Information Modeling [OL]. (2015-07).
- [12] Hjelseth E, Nisbet N. Capturing normative constraints by use of the semantic mark-up RASE methodology [C]// Proceedings of the 28th CIB W78 Conference. France: Sophia Antipolis, 2011: 1-10
- [13] SONG J, KIM J, LEE J K. NLP and Deep Learning-based Analysis of Building Regulations to support Automated Rule Checking System [J]. 35th International Symposium on Automated and Robotics in Construction (ISARC2018), 2018, 35: 1-7.
- [14] 张荷花, 顾明. BIM 模型智能检查工具在审查平台及消防审查中的应用[J]. 土木建筑工程信息技术, 2021, 13(1): 1-7.
- ZHANG H H, GU M. Application of the Intelligent BIM Model Checking Tool in the BIM Platform and Fire Protection Code Checking [J]. Journal of Information Technology in Civil Engineering and Architecture, 2021, 13(1): 1-7. (in Chinese)
- [15] ZHANG J, EI-GOHARY N M. Integrating semantic NLP and logic reasoning into a unified system for fully-automated code checking [J]. Automation in Construction, 2017, 73: 45-57.
- [16] GUO D, ONSTEIN E, LA ROSA A D. A Semantic Approach for Automated Rule Compliance Checking in Construction Industry [J]. IEEE Access, 2021, 9: 129648-129660.
- [17] ZHOU Y C, ZHENG Z, LIN J R, et al. Integrating NLP and context-free grammar for complex rule interpretation towards automated compliance checking [J]. Computers in Industry, 2022, 142, 103746.
- [18] ZHENG Z, ZHOU Y C, LU X Z, et al. Knowledge-informed semantic alignment and rule interpretation for automated compliance checking [J]. Automation in Construction, 2022, 142, 104524.
- [19] SYDORA C, STROULIA E. Rule-based compliance checking and generative design for building interiors using BIM [J]. Automation in Construction, 2020, 120, 103368.
- [20] SOLIHIN W, DIMYADI J, LEE Y C, et al. Simplified schema queries for supporting BIM-based rule-checking applications [J]. Automation in construction, 2020, 117: 103248.
- [21] SOLIHIN W, DIMYADI J, LEE Y C, et al. The critical role of accessible data for BIM-based automated rule checking systems [C]// Proceedings of the Joint Conference on Computing in Construction (JC3).Heraklion, Crete, Greece: Heriot-Watt University, 2017:53-60.
- [22] UHM M, LEE G, PARK Y, et al. Requirements for computational rule checking of requests for proposals (RFPs) for building designs in South Korea [J]. Advanced Engineering Informatics, 2015, 29: 602-615.
- [23] LEE J K, CHO K, CHOI H, et al. High-level implementable methods for automated building code compliance checking [J]. Developments in the Built Environment, 2023, 15, 100174.
- [24] 林佳瑞, 郭建锋. 基于 BIM 的合规性自动审查[J]. 清华大学学报 (自然科学版), 2020, 60(10): 873-879.
- LIN J R, GUO J F. BIM-based automated compliance checking [J]. Journal of Tsinghua University (Sci&Technol), 2020, 60(10): 873-879. (in Chinese)
- [25] The Ministry of Public Security of the People's Republic of China (2014). Code for fire protection design of buildings (GB 50016-2014). (in Chinses)
- [26] 岳清瑞, 陆新征, 许镇, 等. 城市安全“风险源-承灾体-减灾力”理论框架[J]. 工程力学, 2023, 40: 1-9.
- YUE Q R, LU X Z, XU Z, et al. THE “RISK SOURCE-RISK EXPOSURE-MITIGATION FORCE” THEORETICAL FRAMEWORK FOR URBAN SAFETY[J]. ENGINEERING MECHANICS, 2023, 40: 1-9. (in Chinese)
- [27] ZHENG Z, ZHOU Y C, CHEN K Y, et al. A Text Classification-based Approach for Evaluating and

- Enhancing the Machine Interpretability of Building Codes [J]. Engineering Applications of Artificial Intelligence, 2024,127: 107207.
- [28] ZHENG Z, LU X Z, CHEN K Y, et al. Pretrained domain-specific language model for natural language processing tasks in the AEC domain [J]. Computers in Industry, 2022, 142, 103733.
- [29] ZHENG Z, CHEN K Y, CAO X Y, et al. (2023). LLM-FuncMapper: Function Identification for Interpreting Complex Clauses in Building Codes via LLM. arXiv preprint arXiv: 2308.08728.
- [30] 蒋灿, 郑哲, 梁雄, 等. 大语言模型驱动的交互式建筑设计新范式—基于 Rhino7 的概念验证[OL]. 图学学报.
- JIANG C, ZHENG Z, LIANG X, et al. A new interaction paradigm for architectural design driven by large language model: proof of concept with Rhino7 [OL]. JOURNAL OF GRAPHICS. (in Chinese)

附录 1

类别	待审对象	输出值	函数名称	函数解释
存在性检查	空间	Boolean/Collection	hasSpace(space a,space b)	空间 a 包含空间 b;
	空间、构件	Boolean/Collection	hasElement(space a,element b)	空间 a 包含构件 b;
	空间、设备	Boolean/Collection	hasEquipment(space a,equipment b)	空间 a 包含系统或设备 b;
	空间、建筑	Boolean/Collection	hasBuilding(space a,building b)	空间 a 包含建筑 b;
	空间、物品	Boolean/Collection	hasGoods(space a,goods b)	空间 a 包含物品 b;
	空间	Collection	hasExternal(space a)	获取空间 a 外围构件;
	构件	Boolean	hasElement(element a,element b)	构件 a 包含构件 b;
	构件、设备	Boolean	hasEquipment(element a,equipment b)	构件 a 包含系统或设备 b;
	设备、构件	Boolean	hasElement(equipment a,element b)	系统或设备 a 包含构件 b;
	设备、物品	Boolean	hasGoods(equipment a,goods b)	系统或设备 a 包含物品 b;
	建筑、空间	Boolean	hasSpace(building a,space b)	建筑 a 包含空间 b;
	建筑、构件	Boolean	hasElement(building a,element b)	建筑 a 包含构件 b;
	建筑、设备	Boolean	hasEquipment(building a,equipment b)	建筑 a 包含系统或设备 b;
	建筑	Boolean	hasBuilding(building a,building b)	建筑 a 包含建筑 b;
数量检查	建筑、物品	Boolean	hasGoods(building a,goods b)	建筑 a 包含物品 b;
	建筑	Collection	hasExternal(building a)	获取建筑 a 外围构件;
	空间	Integer	getNumberOfSpace(space a)	获取空间 a 的数量;
	构件	Integer	getNumberOfElement(element a)	获取构件 a 的数量;
简单几何检查	设备	Integer	getNumberOfEquipment(equipment a)	获取系统组件或设备 a 的数量;
	建筑	Integer	getNumberOfBuilding(building a)	获取建筑 a 的数量;
	空间	Float	getSpaceWidth(space a, type b)	获取空间 a 在测量方法为 b 时的宽度;
	空间	Float	getSpaceHeight(space a, type b)	获取空间 a 在测量方法为 b 时的高度;
	空间	Float	getSpaceLength(space a, type b)	获取空间 a 在测量方法为 b 时的长度;
	构件	Float	getElementWidth(element a, type b)	获取构件 a 在测量方法

	构件	Float	getElementHeight(element a, type b)	为 b 时的宽度； 获取构件 a 在测量方法为 b 时的高度；
	构件	Float	getElementLength(element a, type b)	获取构件 a 在测量方法为 b 时的长度；
	构件	Boolean	isOpened(element a)	构件 a 是否开口；
	构件	Boolean	isEvenOpened(element a)	构件 a 的开口是否均匀；
	设备	Float	getEquipmentWidth(equipment a, type b)	获取系统或设备 a 在测量方法为 b 时的宽度；
	设备	Float	getEquipmentHeight(equipment a, type b)	获取系统或设备 a 在测量方法为 b 时的高度；
	设备	Float	getEquipmentLength(equipment a, type b)	获取系统或设备 a 在测量方法为 b 时的长度；
	建筑	Float	getBuildingWidth(building a, type b)	获取建筑 a 在测量方法为 b 时的宽度；
	建筑	Float	getBuildingHeight(building a, type b)	获取建筑 a 在测量方法为 b 时的高度；
	建筑	Float	getBuildingLength(building a, type b)	获取建筑 a 在测量方法为 b 时的长度；
距离检查	空间	Float	getSpaceDistance(space a, space b, type c)	获取空间 a 与空间 b 在测量方法为 c 时的距离；
	空间、构件	Float	getLinearDistance(space a, element b, type c)	获取空间 a 中一点与构件 b 在测量方法为 c 时的距离；
	构件	Float	getElementDistance(element a, element b, type c)	获取构件 a 与构件 b 在测量方法为 c 时的距离；
	构件、建筑	Float	getDistance(element a, building b, type c)	获取构件 a 与建筑 b 在测量方法为 c 时的距离；
	构件、设备	Float	getDistance(element a, equipment b, type c)	获取构件 a 与系统组件或设备 b 在测量方法为 c 时的距离；
	设备	Float	getEquipmentDistance(equipment a, equipment b, type c)	获取系统组件或设备 a 与系统组件或设备 b 在测量方法为 c 时的距离；
	建筑	Float	getBuildingDistance(building a, building b, type c)	获取建筑 a 与建筑 b 在测量方法为 c 时的距离；
面积检查	建筑	String	getFireproofDistance(building a, building b, type c)	获取建筑 a 与建筑 b 在测量方法为 c 时的防火间距；
	空间	Float	getFloorArea(space a, type b)	获取空间 a 在测量方法为 b 时的面积；
	构件	Float	getFloorArea(element a, type b)	获取构件 a 在测量方法为 b 时的面积；
	建筑	Float	getFloorArea(building a, type b)	获取建筑 a 在测量方法为 b 时的面积；
空间位置检查	空间	Boolean	isAccessible(space a, space b, type c, direction d)	空间 a 可以通过 c 方式, d 方向到达空间 b；
	空间	Boolean	isVisibleFrom(space a, space b, type c)	空间 b 可以通过方式 c 看到空间 a；
	空间	Boolean	isAdjacent(space a, space b, direction c)	空间 a 在 c 方向上与空间 b 相邻；
	空间、构件	Boolean	isAdjacent(space a, element b,	空间 a 在 c 方向上与构件

	空间、设备	Boolean	direction c) isAdjacent(space a, equipment b, direction c)	b 相邻; 空间 a 在 c 方向上与系统组件或设备 b 相邻;
	构件	Boolean	isOpenDirection(element a, type b)	构件 a 的开启方向时类型 b;
	构件	Boolean	isFacedDirectly(element a, element b)	构件 a 与构件 b 是否正对开设;
	构件	Collection	getAccessibleWithSpace(space a, direction b, type c)	获取与构件 a 连通的空间;
	设备	Boolean	isConnectedTo(equipment a, equipment b)	设备 a 与设备 b 是否相连;
	设备	Boolean	isGroupArranged(equipment a, equipment b)	获取设备 a, 设备 b 是否成组布置;
	建筑	Boolean	isAccessible(building a, building b, type c, direction d)	建筑 a 可以通过 c 方式, d 方向到达建筑 b;
	建筑	Boolean	isAdjacent(building a, building b, direction c)	建筑 a 在 c 方向上与建筑 b 相邻;
	建筑、空间	Boolean	isAdjacent(building a, space b, direction c)	建筑 a 在 c 方向上与空间 b 相邻;
	建筑、设备	Boolean	isAdjacent(building a, equipment b, direction c)	建筑 a 在 c 方向上与系统或设备 b 相邻;
	建筑	Boolean	isGroupArranged(building a, building b)	获取建筑 a, 建筑 b 是否成组布置;
窗墙比检查	构件	Float	getWindowWallRatio(element a, element b, orientation c)	获取构件 a 与构件 b 在四个主要方向上的比例;
属性检查	空间	String/Float/ Boolean/ Collection	getProperty(space a, type b)	获取空间 a 的类型 b 属性;
	构件	String/Float/ Boolean/ Collection	getProperty(element a, type b)	获取构件 a 的类型 b 属性;
	设备	String/Float/ Boolean/ Collection	getProperty(equipment a, type b)	获取系统组件或设备 a 的类型 b 属性;
	建筑	String/Float/ Boolean/ Collection	getProperty(building a, type b)	获取建筑 a 的类型 b 属性;
	物品	String/Float/ Boolean/ Collection	getProperty(goods a, type b)	获取物品 a 的类型 b 属性;