



中华人民共和国国家标准

GB/T 45393.3—2025

信息技术 建筑信息模型(BIM)软件 第3部分:模型视图定义

Information technology—Building information model (BIM) software—
Part 3: Model view definition

2025-04-25 发布

2025-11-01 实施

国家市场监督管理总局
国家标准化管理委员会 发布

目 次

前言	III
引言	IV
1 范围	1
2 规范性引用文件	1
3 术语和定义	1
4 缩略语	3
5 总体要求	4
5.1 MVD 的组成	4
5.2 MVD 与信息模式之间的依赖关系	4
5.3 MVD 语言	4
5.4 MVD 开发和执行过程	5
6 MVD 开发	5
6.1 MVD 开发步骤	5
6.2 编制 IDM	5
6.3 根据交换需求创建 MVD	7
7 MVD 执行	8
7.1 完备性检查	8
7.2 模型视图抽取	10
8 MVD 管理机制	11
8.1 交换需求管理	11
8.2 MVD 成果管理	11
8.3 MVD 软件应用管理	11
9 MVDLite 语言	12
9.1 基本规定	12
9.2 词法规则	12
9.3 句型结构规则	18
9.4 标记规则	24
附录 A(规范性) MVDLite 语法	27
A.1 MVDLite 词法规则	27
A.2 MVDLite 句型结构与标记规则	30
附录 B(资料性) 适用于 IFC 的常用 MVDLite 规则模板	37
B.1 定义类规则模板	37
B.2 约束类规则模板	39

GB/T 45393.3—2025

B.3 模型视图抽取规则模板	53
B.4 标记规则模板	54
参考文献	56



前 言

本文件按照 GB/T 1.1—2020《标准化工作导则 第 1 部分：标准化文件的结构和起草规则》的规定起草。

本文件是 GB/T 45393《信息技术 建筑信息模型(BIM)软件》的第 3 部分。GB/T 45393 已经发布了以下部分：

- 第 1 部分：通用要求；
- 第 2 部分：参数化模型；
- 第 3 部分：模型视图定义；
- 第 4 部分：网格模型；
- 第 5 部分：数据接口。

本文件由全国信息技术标准化技术委员会(SAC/TC 28)提出并归口。

本文件起草单位：清华大学、数云科际(深圳)技术有限公司、中国建筑标准设计研究院有限公司、中国电子技术标准化研究院、北京市建筑设计研究院股份有限公司、上海市隧道工程轨道交通设计研究院、上海天华建筑设计有限公司、东风设计研究院有限公司、广联达科技股份有限公司、中南大学、雄安雄创数字技术有限公司、中国建筑西南设计研究院有限公司、华润置地控股有限公司、上海建工四建集团有限公司、中国铁路设计集团有限公司、上海城建信息科技有限公司、中交第一公路勘察设计研究院有限公司、深圳比目鱼平方科技有限公司、中设数字技术有限公司、深圳前海贾维斯数据咨询有限公司、北京鸿鹄云图科技股份有限公司。

本文件主要起草人：顾明、高歌、刘寒、路子轩、王明彦、王茜、王瑶瑶、毛羽丰、胡筋、李文俊、陈鹏、孙冉、张楠、李会涛、姚旭康、乔世范、刘思铖、杜欣怡、王巍、黄爽、马继生、孙旭、陈亚军、王佐、王凯军、苏林、石磊、康永君、余芳强、许铮铭、韩斌、朱康田、李刚、只飞。



引 言

BIM 软件以三维化、结构化、语义化模型的建立与运用为特征,通过提供 BIM 应用所需的建模、计算、表现、管理、交付、检查、协同功能,在建筑全生命周期不同阶段、不同专业、不同应用中发挥着支撑作用。

GB/T 45393《信息技术 建筑信息模型(BIM)软件》旨在指导从业人员对 BIM 软件形成一致认识,在 BIM 软件开发中,采用本系列标准,实现 BIM 软件的互联互通、BIM 模型的交换共享,提高 BIM 软件的可用性、适用性和成熟度,促进 BIM 软件生态的形成,助力 BIM 模型的资产化。

GB/T 45393 拟由 5 部分构成。

- 第 1 部分:通用要求。明确 BIM 软件的基本概念、功能和数据的要求,目的是协助 BIM 软件的需求定义、开发实现。
- 第 2 部分:参数化模型。给出采用标准化参数集表示几何对象的模型格式,接收方通过解析与修改几何对象参数来显示、修改 BIM 模型中的对象,目的是实现 BIM 软件间可编辑的模型交换。
- 第 3 部分:模型视图定义。提供对模型中的内容进行形式化定义的语言,目的是根据特定需求对提交的模型进行完备性检查或模型视图抽取。
- 第 4 部分:网格模型。给出基于三角形网格表示几何对象不可参数化编辑的模型格式,目的是用于 BIM 模型可视化的表现、沟通、管理。
- 第 5 部分:数据接口。给出不同 BIM 软件与平台间的数据调用方式、发送与返回信息,目的是实现不同的 BIM 软件以接口的方式共享 BIM 数据。

本文件的发布机构提请注意,声明符合本文件时,可能涉及 7.2.2~7.2.6 相关的专利的使用。

本文件的发布机构对于该专利的真实性、有效性和范围无任何立场。

该专利持有人已向本文件的发布机构承诺,他愿意同任何申请人在合理且无歧视的条款和条件下,就专利授权许可进行谈判。该专利持有人的声明已在本文件的发布机构备案。相关信息可以通过以下联系方式获得:

专利持有人:清华大学。

地址:北京市海淀区清华园 1 号。

请注意除上述专利外,本文件的某些内容仍可能涉及专利。本文件的发布机构不承担识别专利的责任。

信息技术 建筑信息模型(BIM)软件

第3部分:模型视图定义

1 范围

本文件规定了模型视图定义(MVD)的开发、执行和管理机制等要求,提供了一个轻量级的模型视图定义语言。

本文件适用于建筑信息模型的数据交换中所需的模型视图定义的开发和执行。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中,注日期的引用文件,仅该日期对应的版本适用于本文件;不注日期的引用文件,其最新版本(包括所有的修改单)适用于本文件。

GB/T 13000 信息技术 通用编码字符集(UCS)

GB/T 36456.3 面向工程领域的共享信息模型 第3部分:测试方法

GB/T 51447 建筑信息模型存储标准

GB/T 45393.1—2025 信息技术 建筑信息模型(BIM)软件 第1部分:通用要求

3 术语和定义

下列术语和定义适用于本文件。

3.1

信息模型 information model

反应现实世界特定领域对象的形式、特性、关系的模式化数据。

[来源:GB/T 36456.1—2018,3.1]

3.2

数据交换 data exchange

数据的存储、访问、传输和归档。

[来源:GB/T 16656.1—2008,3.2.17]

3.3

模型视图 model view

信息模型中的由节点与边组成的满足特定交换需求的子集。

3.4

模型视图定义 model view definition

描述模型视图期望符合的交换需求的计算机可执行规则集。

3.5

业务需求 business requirement

对数据交换的一个业务场景的自然语言描述。

注：业务需求通常包括对数据交换的目的、参与方、流程等信息的描述，以及对涉及的数据内容的概述。

3.6

交换需求 exchange requirement

针对特定业务，对所需要交换的信息单元与相应信息约束的描述。

3.7

信息交付指南 information delivery manual

描述数据交换的业务需求与交换需求的文档，以文本、表格、图表等自然语言形式表示。

3.8

信息单元 information unit

交换需求中的用自然语言描述的信息项。

注：一个信息单元通常为一个单独的对象或一个对象与特性、关系、表现之一的组合。单独的对象，如墙、门、窗等；对象与特性、关系、表现之一的组合，如墙的厚度、墙与门之间的连接关系等。

3.9

信息约束 information constraint

对信息单元的内容限制与使用限制的自然语言描述。

注：内容限制，如节点类型、值域、单位、精度等；使用限制，如节点存在与否、是否必填、数量、唯一性等。

3.10

信息模式 information schema

描述信息模型的数据分类与数据结构约束的计算机可执行规则集。

3.11

数据分类 data classification

描述信息模型允许具有的节点类型与边类型，以及规定节点类型之间、边类型之间的父子类关系的规则。

3.12

数据结构约束 data structure constraint

描述信息模型中节点与边之间允许的连接方式与连接数量的规则。

3.13

子图模板 subgraph template

组成 MVD 的一类规则，描述信息模式中的多种节点类型与边类型组成的图结构。

3.14

子图表达式 subgraph expression

组成 MVD 的一类规则，针对一个子图模板制定，表示对子图实例执行的逻辑判定。

3.15

定义规则 definition rule

一类子图表达式，用于定义扩展节点类型。

3.16

约束规则 constraint rule

一类子图表达式，用于描述一项信息约束。



3.17

固有节点类型 intrinsic node type

在 MVD 引用的信息模式中规定的节点类型。

3.18

扩展节点类型 extended node type

在 MVD 中扩展规定的节点类型。

注：扩展节点类型继承自固有节点类型，扩展节点类型在信息模型中对应的节点集是其继承的固有节点类型对应的节点集的一个子集。

3.19

子图路径 subgraph path

子图模板中两个节点位置之间的一条有向通路，由多个节点类型与边类型交替组成。

3.20

子图实例 subgraph instance

在信息模型中与子图模板匹配的图结构，由若干节点实例与边实例组成。

注：子图实例中节点实例与边实例的组合方式应与子图模板中节点类型与边类型的组合方式一致。

3.21

根概念 root concept

在子图表达式中作为图结构的根节点类型。

3.22

根节点集 root node collection

根概念在信息模型中对应的节点集。

3.23

MVD 开发 MVD development

编制 IDM 文档并根据 IDM 中的交换需求编制 MVD 的过程。

3.24

MVD 执行 MVD execution

计算机程序针对输入的信息模型执行 MVD 并输出结果的过程。

3.25

MVD 语言 MVD language

具有描述子图模板与子图表达式的能力的计算机语言。

3.26

完备性检查 completeness checking

关于信息模型中的组件、属性和关系与全部规定要求相一致的检查。

[来源：GB/T 36456.3—2018,3.2]

3.27

模型视图抽取 model view extraction

计算机程序根据 MVD 从信息模型中抽取一个模型视图，并输出为新的信息模型的过程。新的信息模型符合原信息模型的信息模式。

4 缩略语

下列缩略语适用于本文件。

BIM: 建筑信息模型(Building Information Model)

IDM: 信息交付指南(Information Delivery Manual)

IFC: 工业基础类(Industry Foundation Classes)

LOD:多细节层次(Level Of Details)

MVD:模型视图定义(Model View Definition)

MVDLite:轻量型模型视图定义语言(Model View Definition Lite Language)

URI:统一资源标识符(Uniform Resource Identifier)

5 总体要求

5.1 MVD 的组成

MVD 是 GB/T 45393.1—2025 中 7.5 的内容。MVD 包括子图模板与子图表达式两类规则。

- a) 子图模板用于描述信息模型中可能具有的一种图结构,该结构以一个节点类型作为图结构的根,若干子图路径指向与此节点类型相关的各项信息在信息模型中的位置。
- b) 子图表达式用于描述信息模型中与子图模板匹配的子图实例执行的规则,分为定义规则与约束规则:
 - 1) 定义规则用于定义扩展节点类型,以信息模型中符合定义规则的子图实例的根节点组成的集合作为扩展节点类型对应的节点集;
 - 2) 约束规则用于描述针对子图实例的一项信息约束。

5.2 MVD 与信息模式之间的依赖关系

MVD 引用一个信息模式,基于此信息模式编制 MVD 中的所有规则。MVD 与信息模式之间的依赖关系如图 1 所示。

- a) 子图模板依赖信息模式中的数据分类规则与数据结构约束规则。
- b) 子图表达式依赖子图模板。
- c) 约束规则依赖定义规则。

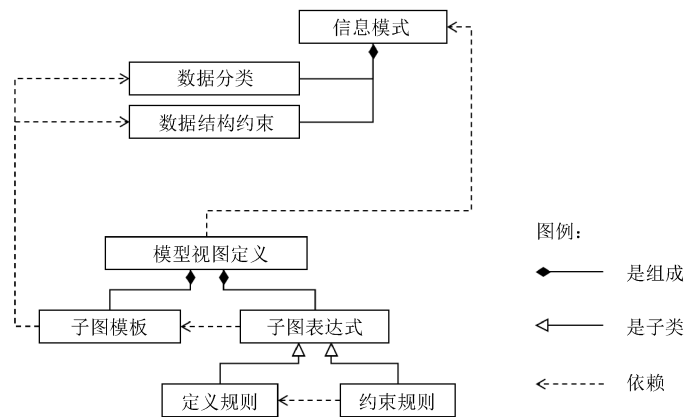


图 1 MVD 与信息模式之间的依赖关系

5.3 MVD 语言

用于描述 MVD 的计算机语言应具备以下能力：

- a) 描述子图模板与子图表达式的能力；
- b) 为规则附加自然语言标记的能力。

第 9 章给出的 MVDLite 语言是一种满足上述要求的、专门用于描述 MVD 的语言。

5.4 MVD 开发和执行过程

MVD 开发和执行过程见图 2。

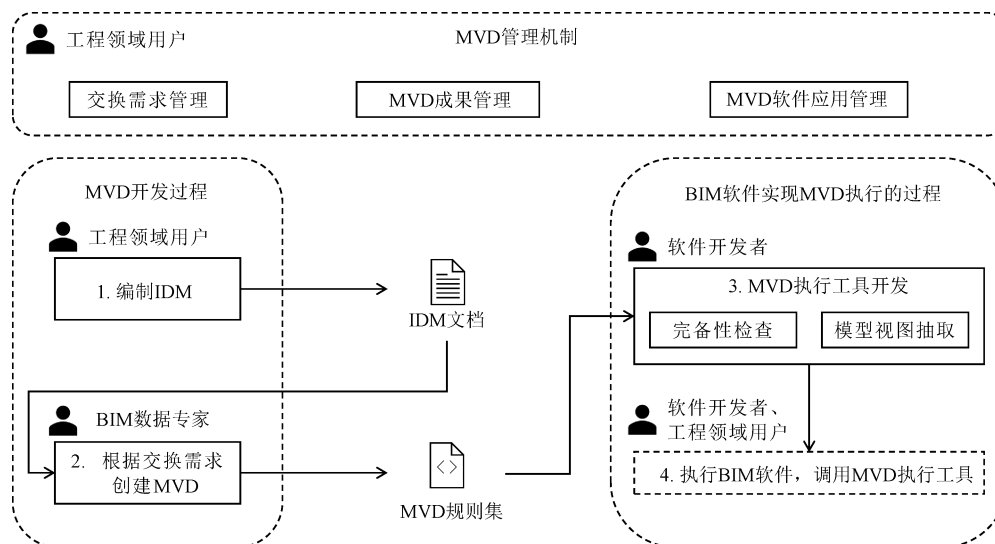


图 2 MVD 开发和执行过程

6 MVD 开发

6.1 MVD 开发步骤

MVD 的开发包括以下步骤：

- 编制 IDM: 描述工程领域中的业务需求, 并明确列举数据的交换需求, 形成 IDM 文档;
- 根据交换需求创建 MVD: 针对特定的信息模型, 将 IDM 文档中的交换需求转换为计算机可执行的 MVD。

注: MVD 的开发通常需要工程领域从业者与信息领域从业者的协作, 对工程领域中的业务场景与业务需求进行分析, 明确描述交换需求中的信息单元与信息约束, 并将其转换为计算机可执行的规则集。

6.2 编制 IDM

6.2.1 IDM 的内容

6.2.1.1 IDM 的组成

IDM 应对业务场景与交换需求进行详细描述, 使目标读者能明确地识别出各项业务场景对应的交换需求, 以及各项交换需求中的信息单元与信息约束。IDM 的主要组成见图 3。

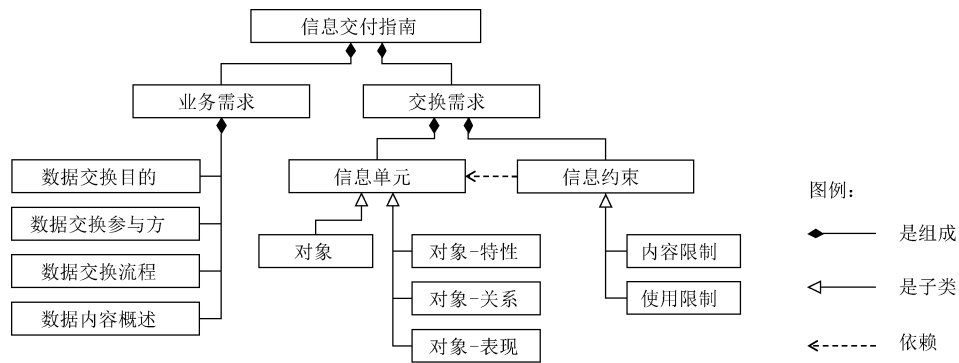


图 3 IDM 的主要组成

6.2.1.2 IDM 业务需求

IDM 中的业务需求描述应包括以下内容：

- a) 数据交换目的：数据交换的目的与应实现的目标；
- b) 数据交换参与方：数据发送方与接收方的角色，以及双方的交互；
- c) 数据交换流程：工程项目中数据交换的流程图，以及流程图中每个数据交换场景的概况；
- d) 数据内容概述：数据交换涉及的数据格式与内容范围的概况。

6.2.1.3 IDM 交换需求

IDM 中的交换需求描述应包括对信息单元与信息约束的列举。

- a) 列举单一对象信息单元：将业务需求中涉及的对象按照其功能、用途、形态等特征进行分类，说明将某一类对象与其他类型对象区分开的特征。
- b) 列举对象与特性、关系、表现组合的信息单元：列出每种对象涉及的特性、关系、表现信息，以及上述信息与业务需求中的项目的对应关系。
- c) 列举信息约束：针对每个信息单元列出内容限制与使用限制描述，以及上述约束与业务需求中的项目的对应关系。其中：
 - 1) 内容限制可包括对象命名要求、对象分类标识要求、特性类型要求、特性取值要求、几何表现要求、颜色表现要求、材质表现要求等；
 - 2) 使用限制包括对象、特性、关系、表现是否存在的要求，以及特性、关系、表现应当满足的数量限制、唯一性限制等。

6.2.2 IDM 引用语义字典

在实施对象分类以及描述对象类型的识别特征的过程中，应引用针对工程领域对象的语义字典标准或分类标准。当 IDM 中的对象名称与所引用的分类标准不一致时，应按照语义字典的规定进行领域规范术语映射。

6.2.3 IDM 文档成果

IDM 文档是编制 IDM 的成果。IDM 文档成果中除了涵盖各种局部对象的业务需求与交换需求外，还应涵盖模型文件要求与模型全局要求。

- a) 模型文件要求：信息模型的文件格式、文件命名规则等要求。

- b) 模型全局要求:信息模型中全局唯一的或具有全局影响的信息单元与信息约束,如模型的基点坐标、计量单位、空间组织结构等要求。

6.3 根据交换需求创建 MVD

6.3.1 MVD 的创建步骤

根据交换需求创建 MVD 的过程包括以下步骤:

- a) 选择 MVD 引用的信息模式:MVD 应针对特定的信息模式制定,应根据业务需求,确定一个或多个引用的信息模式;
- b) 转译信息单元与信息约束:针对信息模式,使用 MVD 语言将信息单元与信息约束转译为 MVD 规则;
- c) MVD 校验模型准备:创建一个或多个信息模型,并声明完备性检查或模型视图抽取的预期结果,用于校验 MVD 的正确性;
- d) MVD 规则正确性校验:在 MVD 校验模型上执行完备性检查或模型视图抽取,将实际结果与预期结果进行比较,并修正 MVD 规则。

6.3.2 选择 MVD 引用的信息模式

MVD 引用的信息模式应满足以下方面的使用要求:

- a) 信息模式应支持描述信息单元中的对象、特性、关系、表现,数据交换上下游软硬件系统支持输入、输出、转换过程中的信息损耗情况能满足使用要求;
- b) 信息模式的可扩展性、可编辑性、可维护性能满足使用要求;
- c) 用于编辑模型的工具的授权、使用期限等能满足使用要求;
- d) 信息模式符合数据交换双方角色的其他约定。

6.3.3 转译信息单元与信息约束

使用 MVD 语言将信息单元与信息约束转译为 MVD 规则应包括:

- a) 从信息单元中的对象类型到信息模式中的根节点类型的映射;
- b) 对象类型的识别特征在 MVD 定义规则中的表达;
- c) 从信息单元中的特性、关系、表现到子图模板中的子图路径的映射;
- d) 针对某信息单元的信息约束在 MVD 约束规则中的表达。

6.3.4 MVD 校验模型准备

6.3.4.1 完备性检查的预期结果声明

MVD 校验模型准备中,完备性检查的预期结果声明应包括:

- a) 信息单元中的对象类型与信息模型中的根节点集的对应关系;
- b) 根节点集中每个元素的特性、关系、表现的存在性与数量的声明;
- c) 根节点集中每个元素对某条信息约束的符合性的声明。

6.3.4.2 模型视图抽取的预期结果声明

MVD 校验模型准备中,模型视图抽取的预期结果声明应包括:

- a) 信息单元中的对象类型与信息模型中的根节点集的对应关系;
- b) 根节点集中每个元素是否应包含在被抽取的模型视图中的声明;

- c) 根节点集中每个被包含在被抽取的模型视图中的元素是否应具有某项特性、关系、表现的声明。

6.3.5 MVD 规则正确性校验

6.3.5.1 MVD 规则正确性校验的目标

MVD 规则正确性校验应在 IDM 编制者、MVD 创建者、MVD 执行应用开发者三者之间进行,以保证以下方面的正确性:

- a) IDM 中的描述以及 IDM 读者对其的理解与 IDM 编制者的真实意图符合;
- b) MVD 中的计算机可执行规则与 IDM 中的描述符合,且与其引用的信息模式兼容;
- c) MVD 校验模型已按照 IDM 的描述声明了预期的 MVD 执行结果;
- d) 执行 MVD 的结果与声明的预期结果符合。

6.3.5.2 MVD 规则正确性校验的实施方法

应按照第 7 章的要求实施 MVD 规则正确性校验,其中用于完备性检查的 MVD 规则的正确性校验按照 7.1 所述方法执行,用于模型视图抽取的 MVD 规则的正确性校验按照 7.2 所述方法执行。

7 MVD 执行

7.1 完备性检查

7.1.1 完备性检查的目的

基于 MVD 对信息模型执行完备性检查,以确认信息模型是否满足特定的交换需求。

7.1.2 完备性检查的流程

MVD 完备性检查的流程应按照 GB/T 36456.3 所述方法执行。

7.1.3 子图表达式与测试项的对应

在基于 MVD 的完备性检查中,MVD 中的一条定义规则对应一个根概念,MVD 中的一条约束规则对应一个测试项。在完备性检查执行时,先执行定义规则,后执行约束规则。

7.1.4 子图表达式的执行步骤

一条 MVD 子图表达式的执行步骤包括根节点集获取、子图匹配、子图过滤、结果输出四个步骤,如图 4 所示。



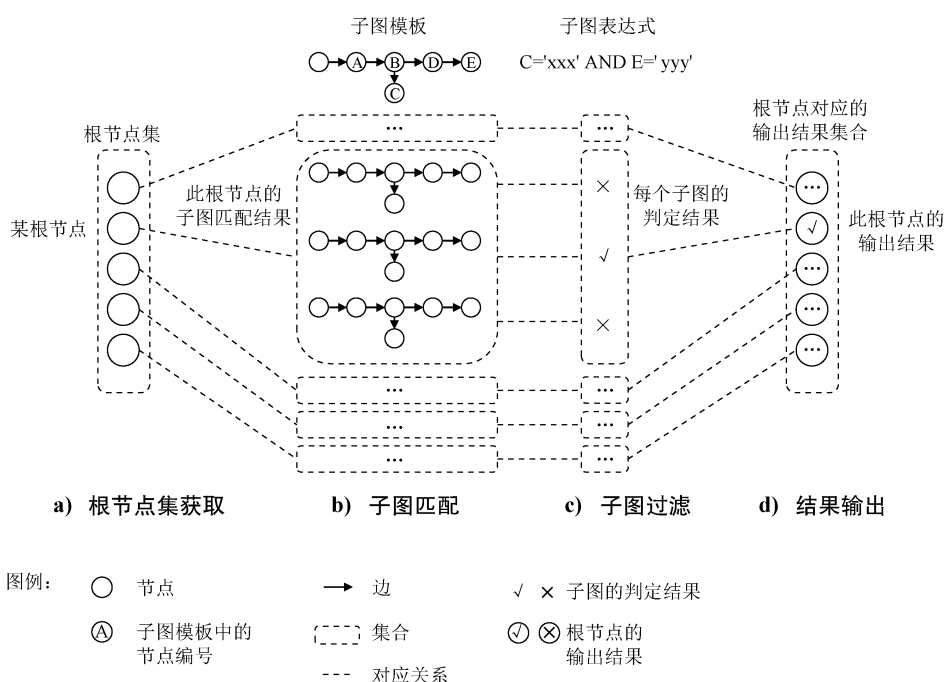


图 4 子图表达式执行步骤

- a) 根节点集获取: 根节点集获取步骤中, 从信息模型中获取一个节点集作为约束规则的根节点集。
- 1) 如果约束规则不引用定义规则而直接引用信息模式中的根节点类型, 则使用信息模型中根节点类型对应的节点集作为根节点集。
 - 2) 如果约束规则引用定义规则, 则根据定义规则获取根概念对应的节点集作为根节点集。
- b) 子图匹配: 子图匹配步骤中, 根据约束规则引用的子图模板, 在信息模型中查找以根节点集中的每个元素为起始节点的所有匹配的子图实例。匹配子图实例包括与子图模板中的所有节点类型与边类型均匹配的完全匹配子图实例, 以及无法找到与子图模板中某个位置的边类型与节点类型匹配的边与节点, 从而自此位置之后的节点位置留空的部分匹配子图实例。
- c) 子图过滤: 子图过滤步骤中, 针对子图实例集合中的每个子图实例使用子图表达式进行测试, 并根据结果将子图实例标记为“符合”或“不符合”。
- d) 结果输出: 结果输出步骤中, 输出的结果为根节点集中每个元素的“符合”与“不符合”的结果。如果存在以某根节点为起始节点的且标记为“符合”的子图实例, 则此根节点的结果为“符合”, 否则结果为“不符合”。其中, 对于定义规则, 结果为“符合”的根节点组成定义规则对应根概念的节点集; 对于约束规则, 结果为“符合”的根节点组成通过测试项的节点集, 结果为“不符合”的根节点组成不通过测试项的节点集。

示例:

定义
 养老院 继承 IfcBuilding
 养老院->类型标签 = '养老院'
 约束
 养老院->实例属性集->实例属性(->属性名称='建筑高度')->属性值! = 空

执行步骤如下:



- 根据定义规则,获取所有“类型标签 = ‘养老院’”的 IfcBuilding 作为根节点集;
- 根据约束规则“->实例属性集->实例属性->属性值”搜索每个根节点对应的匹配或部分匹配子图;
- 在上一步的搜索结果中,根据“属性名称 = ‘建筑高度’”及“属性值! =空”过滤子图,将符合该条件的子图标记为“符合”,反之标记为“不符合”;
- 对根节点集中的每个根节点,如果存在标记为“符合”的子图,则此根节点输出结果为真;否则此根节点输出结果为假。

7.2 模型视图抽取

7.2.1 目的

基于 MVD,可对信息模型执行模型视图抽取,从信息模型中抽取出一个子集,这个子集含有来自于原信息模型的若干节点与边。这些节点与边组成的新的信息模型,应符合原信息模式,且符合 MVD 的约束条件,满足 MVD 对应的业务需求。

7.2.2 模型视图抽取的输出

模型视图抽取生成的新的信息模型应符合下列要求:

- a) 符合原信息模型的信息模式的数据分类与数据结构约束要求;
- b) 交换需求中声明需要的对象、特性、关系、表现的信息包含在此信息模型中;
- c) 交换需求中要求排除的对象、特性、关系、表现的信息不包含在此信息模型中。

7.2.3 模型视图抽取的规则集

模型视图抽取使用的规则集由 MVD 中的定义规则、约束规则以及信息模式中的关键边保留规则组成。其中:

- a) MVD 中的一条定义规则对应一个根概念,其用途与 6.2 中所述相同;
- b) MVD 中的一条约束规则表示一项信息单元存在性声明,分为根节点保留规则、根节点排除规则、路径保留规则、路径排除规则四种类型;
- c) 关键边保留规则是由信息模式中规定为应存在的边类型组成的规则,其目的是使模型视图抽取生成的新的信息模型符合原信息模型的信息模式的数据分类与数据结构约束要求。例如,当使用 GB/T 51447 作为信息模式时,关键边保留规则宜保留其中的所有未标记为“OPTIONAL”的边类型;可根据应用需求添加其他边类型作为关键边,例如,将 IfcProduct 类型的 Representation 边作为关键边,以保留产品对象的几何信息。

7.2.4 执行模型视图抽取的软件功能

执行模型视图抽取的软件模块应包括以下功能:

- a) 根概念定义规则执行与根节点查找:根据定义规则,查找每个需要抽取的根概念在信息模型中对应的根节点集;
- b) 信息单元存在性标记:根据信息单元存在性声明与关键边保留规则,对信息模型中的节点与边进行存在性标记,将其标记为“保留”或“排除”;
- c) 数据抽取与输出:抽取标记为“保留”的节点与边,并输出为新的信息模型。

7.2.5 规则执行结果

信息单元存在性标记与数据抽取与输出的执行结果应符合以下要求:

- a) 根节点保留规则:符合保留规则的根节点集中的节点标记为“保留”,并包含在输出的信息模型中;
- b) 根节点排除规则:符合排除规则的根节点集中的节点标记为“排除”,并在输出的信息模型中不出现;
- c) 路径保留规则:从标记为“保留”的根节点集出发的、符合路径保留规则的子图实例,路径中沿途所有节点与边标记为“保留”,并包含在输出的信息模型中;
- d) 路径排除规则:从标记为“保留”的根节点集出发的、符合路径排除规则的子图实例,路径的最后一段边标记为“排除”,并在输出的信息模型中不出现;
- e) 关键边保留规则:与标记为“保留”的节点与边相关的、符合关键边保留规则的节点与边标记为“保留”,并包含在输出的信息模型中。

7.2.6 规则优先级

在信息单元存在性标记中,当出现规则之间的冲突时,排除规则的优先级高于保留规则,根节点规则的优先级高于路径规则。

8 MVD 管理机制

8.1 交换需求管理

交换需求管理符合如下要求:

- a) 数据交换的需求方应整理描述业务需求与交换需求所需的术语概念,在 IDM 中统一术语定义;
- b) 数据交换的需求方应编制适于描述 IDM 的文档模板;
- c) MVD 所涵盖的工程信息模型交换场景,应符合工程建设的实际流程需要,涵盖设计协同、设计算量、设计施工深化、竣工验收等用例流程。各场景的应用要求标准,应符合相关国家标准或行业标准要求。

8.2 MVD 成果管理

MVD 成果管理符合如下要求:

- a) MVD 成果管理应覆盖 IDM 文档与 MVD 规则文件;
- b) 针对同一需求、由不同方编制的 MVD 规则文件,应在统一的评审机制下,完成去重、冲突解决和合并等工作,最终形成统一的 MVD 规则文件存档;
- c) 应对 MVD 成果进行分类和索引,备注来源、历史、继承或变更信息;
- d) MVD 规则文件中,同一交换需求可针对不同交换场景对应多个不同的约束规则;此时应使用标记项明确规定其对应的场景,并在 MVD 执行中据此筛选适用于当前场景的规则。

8.3 MVD 软件应用管理

MVD 软件应用管理符合如下要求:

- a) BIM 软件调用 MVD 执行时,应确保当前使用的 BIM 信息模式、交换场景与 MVD 规则文件声明的信息模式、适用场景一致;
- b) 针对 MVD 的内容,宜配套形成面向主要 BIM 软件的配置文件,以提高用户建模效率;
- c) BIM 软件宜支持客户端与云平台之间的 MVD 规则同步与更新。

9 MVDLite 语言

9.1 基本规定

9.1.1 MVDLite 是一种 MVD 语言,其语法由词法、句型结构、标记三部分规则组成。其中:

- a) 词法规则规定了 MVDLite 语言中的标识符、数值、字符串、符号、关键字等成分表示方法,应符合附录 A 中 A.1 的规定;
- b) 句型结构规则规定了 MVDLite 语言表示子图模板、定义规则、约束规则的方法,应符合 A.2 的规定;
- c) 标记规则规定了在 MVDLite 句型结构基础上附加标记信息的方法,应符合 A.2 的规定。

9.1.2 MVDLite 语言使用的字符集为 GB/T 13000 字符集。

9.1.3 MVDLite 规则文件应使用“.mvd lite”或“.mdl”作为扩展名。

9.1.4 适用于 IFC 信息模式的常用 MVDLite 模板见附录 B。

9.2 词法规则

9.2.1 标识符

标识符是 MVDLite 语言中用于表示节点、边、子图路径、标记的名称,类似于编程语言中的变量名。

标识符由大小写字母(A-Z、a-z)、数字(0-9)、汉字字符、下划线(_)组成,且首字符不能为数字。其中汉字字符为 GB/T 13000 字符集中从\u4e00 到\u9fa5 之间的字符。

当工程领域术语中含有超出上述字符范围的特殊符号,然而应使用此带有特殊符号的术语作为标识符时,应使用标识符字符串表示,以代替通常的标识符。

9.2.2 整数

整数由 1 至多个数字,以及可选的符号前缀组成。

符号前缀包括正号(+)与负号(-),其中正号可以省略。

9.2.3 浮点数

浮点数由以小数点分隔的两组 0 至多个数字,以及可选的符号前缀与指数后缀组成。其中至少有一组数字的长度不为 0。

普通浮点数由以小数点分隔的两组 0 至多个数字,以及可选的符号前缀组成,其中至少有一组数字的长度不为 0。

科学计数法浮点数表示 10 进制科学计数法,由一个整数或普通浮点数作为前缀,以字母 e 或 E 后接表示指数的另一个整数作为后缀。



9.2.4 值域

值域表示实数轴上的一个连续闭区间。值域表示分为上下界型值域与中心误差型值域两种,具体内容如下。

- a) 上下界型值域由波浪线(~)连接的左右两个浮点数或整数组成,分别表示闭区间的下界与上界,其中左侧的数值应更小。如:0.1~0.9。
- b) 中心误差型值域由正负号(+ -)连接的左右两个浮点数或整数组成,左侧为中心值,右侧为误

差值,其中右侧值应为正数。如:3.5+-0.01。

9.2.5 字符串与转义字符

字符串由一对半角双引号(")或一对半角单引号(')包围的 0 至多个字符组成。字符串中的内容不能超过一个物理行,其中的换行符(\n)与回车符(\r)应使用转义表示。

字符串中的转义字符由反斜杠(\)后接特定字符组成,用于表示特殊字符。需要转义的字符如表 1 所示。

表 1 MVDLite 转义字符

符号名称	符号表示
双引号	\"
单引号	\'
反斜杠	\\
换行符	\n
回车符	\r
制表符	\t

9.2.6 部分匹配字符串

部分匹配字符串是在字符串前增加表示部分匹配的符号,用于表示 MVD 规则中对字符串值进行部分匹配判定的运算。

部分匹配的符号类型如表 2 所示。

表 2 MVDLite 部分匹配字符串

部分匹配类型	部分匹配符号	示例	满足部分匹配条件的字符串样例
前缀匹配	s	s"app"	"apple", "apply", "append"
后缀匹配	e	e"ing"	"doing", "interesting", "boring"
包含匹配	c	c"ly"	"lyrics", "happily", "flying"

9.2.7 正则匹配字符串

正则匹配字符串是 MVD 规则中对字符串值进行正则表达式匹配判定的运算。

正则匹配字符串以表示正则匹配的符号(r)开头,后接由一对半角双引号(")或一对半角单引号(')包围的正则表达式。

正则匹配字符串中的功能字符如表 3 所示。这些功能字符在正则表达式中具有特殊功能,当需要匹配这些字符本身时,需要使用转义表示。

注:功能字符用于“[]”包围的字符集合内部时不必转义。

表 3 MVDLite 正则匹配字符串中的功能字符

符号	符号功能	转义表示
^	表示当前位置对应字符串的开头	\^
\$	表示当前位置对应字符串的结尾	\\$
()	表示正则表达式中的一组子表达式	\(\)
[]	其中的若干字符组成一个字符集合	\[\]
{ }	其中的数字表示当前子表达式的匹配次数	\{ \}
?	表示当前子表达式匹配 0 次或 1 次	\?
+	表示当前子表达式匹配 1 次或 1 次以上	\+
*	表示当前子表达式匹配 0 次或 0 次以上	*
	表示在左右两个子表达式中匹配任意一项	\
.	表示匹配除了换行符 \n 以外的任何单一字符	\\.

除了表 1 所示的转义字符以外,正则匹配字符串中可使用的额外的转义字符如表 4 所示。

表 4 MVDLite 正则匹配字符串中的额外转义字符

符号功能	转义表示
匹配一个数字	\d
匹配一个非数字	\D
匹配一个空白字符,包括空格、制表符、回车符、换行符	\s
匹配一个非空白字符	\S
匹配一个单词边界,即当前位置介于空白字符与非空白字符之间	\b
匹配一个非单词边界	\B
匹配数字、大小写字母、下划线中的一个字符	\w
匹配除了数字、大小写字母、下划线之外的一个字符	\W

正则匹配字符串的示例如表 5 所示。

表 5 MVDLite 正则匹配字符串

示例	满足部分匹配条件的字符串样例
r"F\d\d \$"	"F01", "F13"
r"F\d{1,2} \$"	"F1", "F01", "F13"
r"F\d+ \$"	"F1", "F01", "F13333"
r"F\d? \$"	"F", "F1"

表 5 MVDLite 正则匹配字符串 (续)

示例	满足部分匹配条件的字符串样例
<code>r"~F\d* \$"</code>	"F", "F1", "F13333"
<code>r"~[BF]\d+ \$"</code>	"F1", "B01"
<code>r"~(B F RF)\d+ \$"</code>	"F1", "B01", "RF1"

9.2.8 标识符字符串

标识符字符串是在字符串前增加表示标识符的符号(\$),在 MVDLite 句型结构中的作用等价于标识符。

由于标识符中不允许出现空白字符(空格、换行符等)与除下划线以外的特殊符号,也不允许标识符的首字符为数字,因此可以使用标识符字符串来表示不满足上述条件的标识符。

例如:\$ "钢管-树脂管连接件"、\$ "出入口(应急)"、\$ "300 °C 防火阀"。

9.2.9 逻辑值及其他特殊值

MVDLite 中的逻辑值及其他特殊值如表 6 所示。

表 6 MVDLite 逻辑值及其他特殊值

英文表示	中文表示
TRUE	真
FALSE	假
UNKNOWN	未知
NULL	空
SELF	自身

9.2.10 度量符号

MVDLite 中的度量符号如表 7 所示。

表 7 MVDLite 度量符号

英文度量符号	中文度量符号
[Value]	[值]
[Type]	[类型]
[Size]	[数量]
[Exists]	[存在]
[Unique]	[唯一]

9.2.11 比较符号

MVDLite 中的比较符号如表 8 所示。

表 8 MVDLite 比较符号

符号表示	描述
=	等于
! =	不等于
>	大于
>=	大于或等于
<	小于
<=	小于或等于

对于各种数据类型的[Value]度量,比较符号的含义如表 9 所示。

表 9 MVDLite[Value]度量比较符号的含义

左侧数据类型	右侧数据类型	比较符号含义
字符串	字符串	按字典序排序的字符串顺序比较
整数、浮点数	整数、浮点数	按实数轴排序的数值大小比较
	值域	左侧值与右侧值域在实数轴上的位置关系
逻辑值	逻辑值	只支持“=”与“! =”符号

对于[Type]度量,比较符号的含义如表 10 所示。

表 10 MVDLite[Type]度量比较符号的含义

符号表示	含义
$A = B$	A 与 B 是同一个类型
$A \neq B$	A 与 B 是不同类型
$A > B$	A 是 B 的子类
$A < B$	A 是 B 的父类
$A \geq B$	A 是 B 的子类或者 A 与 B 相同
$A \leq B$	A 是 B 的父类或者 A 与 B 相同

注: 此处将概念的分类系统抽象为横向的树结构,“ $A < B$ ”表示“A 作为父类有若干子类分支,其中一个分支是 B”;“ $A > B$ ”表示“B 作为父类有若干子类分支,其中一个分支是 A”。

9.2.12 逻辑运算符号

MVDLite 中的逻辑运算符号如表 11 所示。

表 11 MVDLite 逻辑运算符号

英文表示	中文表示	符号表示
AND	且	&
OR	或	
XOR	异或	^
NOT	不、非	!

9.2.13 其他符号

MVDLite 中的其他符号如表 12 所示,其中字符均为半角符号字符。

表 12 MVDLite 其他符号

符号名称	符号表示
指向	->
左括号	(
右括号)

9.2.14 关键字

MVDLite 中的关键字如表 13 所示。

表 13 MVDLite 关键字

英文关键字	中文关键字
definition	定义
constraint	约束
as	作为
extends	继承

9.2.15 空格

在 MVDLite 语言中,词法规则中所述各成分之间可以使用 1 至多个空格分隔。当两个词法成分连写不会造成歧义时,空格可以省略。

空格包括空格符、制表符、换行符、回车符四种。

空格不参与语法解析,且不能出现在除字符串、部分匹配字符串、正则匹配字符串、标识符字符串以外的其他词法成分中。

在字符串、部分匹配字符串、正则匹配字符串、标识符字符串中,空格符正常书写,制表符、换行符、回车符应使用转义字符表示。

9.2.16 注释

在 MVD Lite 语言中,注释内容不参与语法解析,仅在人类直接阅读 MVD Lite 时提供辅助说明。注释分为单行注释与多行注释两种。

- a) 单行注释以双斜杠(//)开始,直到行尾结束。
- b) 多行注释以斜杠星号(/ *)开始,以星号斜杠(* /)结束。

9.3 句型结构规则

9.3.1 规则文件

规则文件由头部的预定义缩写与若干正文块组成,其中预定义缩写可省略。

9.3.2 预定义缩写

预定义缩写位于规则文件头部,由若干缩写表达式构成。

9.3.3 正文块

正文块是规则文件中用于组织 MVD 规则的主要结构。

一个规则文件可以具有 0 至多个正文块。

正文块分为定义块与约束块两种,每个正文块由关键字“定义”或“约束”开头,直到下一个“定义”或“约束”关键字之前,或文件结尾之前结束。

9.3.4 定义块

定义块是表示 MVD 定义规则的正文块,用于规定在信息模型中查找某根概念对应的节点集的规则。

定义块由关键字“定义”开头,其内部可含有若干概念表达式与规则表达式。

9.3.5 约束块

约束块是表示 MVD 约束规则的正文块,用于规定信息模型中与子图模板匹配的子图实例应当满足的约束规则。

约束块由关键字“约束”开头,其内部可含有若干规则表达式。

9.3.6 缩写表达式

缩写表达式用于将 MVD Lite 的规则链中的一些常用的规则段序列缩写为一个标识符,以便于其在规则表达式中引用。

- a) 缩写表达式由以关键字“as”连接的前后两部分组成。前部分为一个标识符,是缩写表达式的名称;后部分以括号包围的根节点类型开头,后接一个由若干中间规则段组成的序列。其中根节点类型可为空,此时表示不限制根节点类型。
- b) 在正文的规则表达式中,可以使用缩写表达式的标识符代替其后部分的规则段序列。
- c) 在缩写表达式的后部分的规则段序列中,也可以嵌套使用其他缩写标识符,但是不允许形成循环引用。



9.3.7 概念表达式

概念表达式用于声明一个新的扩展节点类型,作为已有的节点类型的一个子类。

- a) 概念表达式由以关键字“extends”连接的前后两部分组成。前部分为一个标识符,是新的扩展节点类型的名称;后部分为一个已有的节点类型,或者多个已有的节点类型的并集,用“或”分隔(可以为汉字“或”,大写英文“OR”,或者竖线“|”)。
- b) 概念表达式只能出现在定义块中,并与定义块中的其他以此节点类型为根概念的规则表达式共同组成此节点类型的定义规则。
- c) 以一个节点类型为根概念的规则表达式可出现在与概念表达式所处的定义块不同的其他定义块中。

9.3.8 规则表达式

规则表达式是正文块中表示定义规则或约束规则的主要结构。

- a) 规则表达式可以为一条单一的规则语句,也可以为若干条规则语句、逻辑连接符号、括号的组合。
- b) 当规则表达式中含有多条规则语句时,每条规则语句的根概念应相同。
- c) 当使用逻辑连接符号为“XOR”时,当前语句(或当前括号内的子句)只能有两项。
- d) 当逻辑连接符号“NOT”只能位于当前语句(或当前括号内的子句)的开头,且此句只能有一项。

9.3.9 规则语句

规则语句是规则表达式中的基本单元。

- a) 一个规则语句由一个根概念后接一个度量规则链组成,表示根概念对应的节点集上的一个过滤条件。
- b) 规则语句执行的结果是根概念对应的节点集中满足度量规则链的节点子集。

9.3.10 规则链

规则链是由若干规则段串联组成的链条,是 MVDLite 句型结构中用于表示一段子图路径以及这段子图路径上的规则的结构。

规则链按照功能分为度量规则链与路径规则链两种,以其结尾规则段的类型作为区分。

9.3.11 度量规则链

度量规则链表示对此规则链之前的节点集的过滤条件,应以度量段、对偶度量段或复合度量段结尾。

9.3.12 路径规则链

路径规则链表示以此规则链之前的节点集为起点的子图路径,应以属性段或复合属性段结尾。

9.3.13 规则段

规则段分为属性段、度量段、对偶度量段、复合属性段、复合度量段五种。

其中,度量段只能出现在规则链的结尾,不能出现在规则链的中间;其他四种规则段可以出现在规则链的任意位置。

如需以度量段作为规则链结尾,应使用括号将度量段封装为复合度量段。

如需将一个以复合度量段结尾的规则链作为路径规则链使用,应在其后添加一个“->SELF”属性段。

9.3.14 属性段

属性段是表示子图路径中的一个边类型的规则段,形如“—>边类型标识符:目标节点约束标识符”。其中“:目标节点约束标识符”可省略,表示不限制此边类型的目标节点类型。

属性段的目标节点集为目标节点的无序不重复集合。具体分为以下情况:

- a) 当边指向单一目标节点时,将目标节点加入目标节点集;
- b) 当边指向包含多个节点的节点集时,将其中的每个节点加入节点目标节点集;
- c) 当边指向包含两层以上嵌套的节点集时,仅展开最外层集合,将其内部的每个嵌套集合看作一个目标,加入目标节点集。

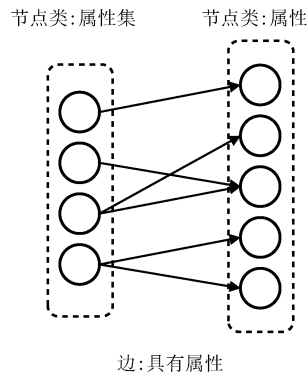


图 5 属性段示例

注:边类型标识符使用一个缩写表达式名称,此时使用此缩写对应的规则段序列替换此属性段;如果存在“:”表示的目标节点类型约束,则将此约束应用于规则段序列的末尾。

9.3.15 度量段



度量段是表示对当前节点集的一个过滤条件的规则段,由一个比较符号与左侧的度量符号、右侧的值组成,如“[Size] > 3”。其中,当度量符号为“[Value]”时,度量符号可以省略。

根据度量段的行为的不同,度量段分为单度量段与集合度量段两类。

9.3.16 单度量段

单度量是可以由单一节点确定的度量,其输出的节点集是对当前输入节点集的过滤。

单度量段的符号为“[Value]”与“[Type]”两种。

- a) [Value]度量段根据当前输入节点集的每个节点的值与比较符号右侧的值的比较结果,对当前输入节点集进行过滤。当度量符号为“[Value]”,且值为部分匹配字符串或正则匹配字符串时,比较符号只能为“=”或“!=”,示例参见图 6。
- b) [Type]度量段根据当前输入节点集的每个节点的类型与比较符号右侧的类型在节点的分类体系中的关系进行过滤。当度量符号为“[Type]”时,比较符号右侧只能为字符串值,表示节点类型名称;此时比较符号的示例参见图 7。

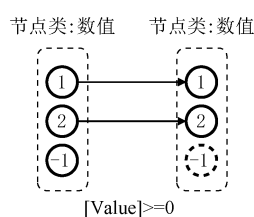


图 6 [Value]度量段示例

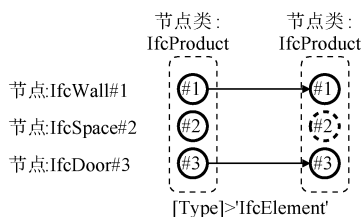


图 7 [Type]度量段示例

9.3.17 集合度量段

集合度量段是由当前节点集与其以属性段或复合属性段关联的上一级节点集的引用关系进行判定的度量,其输出的节点集是对上一级节点集的过滤。

集合度量段符号为“[Size]”“[Exists]”“[Unique]”三种。

- [Size]度量段根据上一级节点集中的每个节点指向当前节点集中节点的数量,对上一级节点集进行过滤。当度量符号为“[Size]”时,值只能为整数。[Size]度量段的示例见图 8。
- [Exists]度量段根据上一级节点集中的每个节点是否存在指向当前节点集中节点,对上一级节点集进行过滤。当度量符号为“[Exists]”时,比较符号只能为“=”或“!=”,值只能为逻辑值。[Exists]度量段的示例见图 9。
- [Unique]度量段根据上一级节点集中的每个节点指向当前节点集中的节点是否与其他上一级节点互斥,对上一级节点集进行过滤。当指向存在且互斥的情况下[Unique]度量为 TRUE,否则为 FALSE。当度量符号为“[Unique]”时,比较符号只能为“=”或“!=”,值只能为逻辑值。[Unique]度量段的示例见图 10。

注 1: 由于度量段只能出现在规则链的结尾,因此上一级节点集是当前规则链中的最后一个属性段或复合属性段的输入节点集。

注 2: 如规则链中不存在属性段或复合属性段,则上一级节点集为规则链的根节点集;但是此规则不适用于复合属性段或复合度量段内部的规则链,也不适用于[Unique]度量。

注 3: 当上述最后一个属性段或复合属性段与集合度量段之间存在其他复合度量段时,将这些复合度量段的过滤结果作为当前节点集,并进行集合度量的判定。

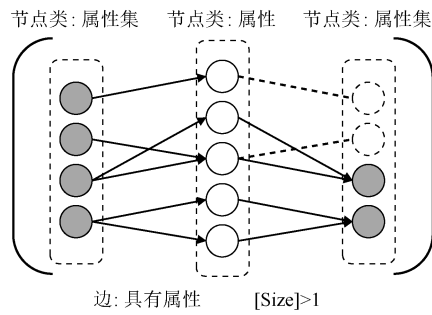


图 8 [Size]度量段示例

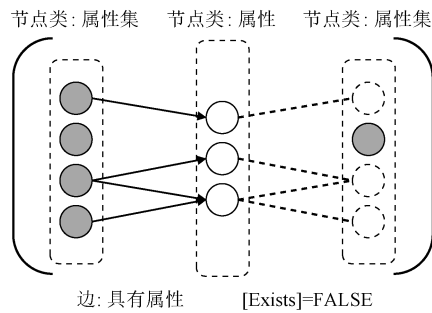


图 9 [Exists]度量段示例

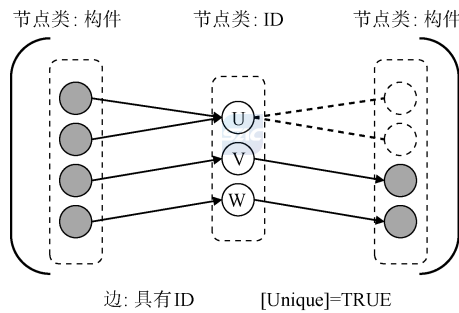


图 10 [Unique]度量段示例

9.3.18 对偶度量段

对偶度量段是表示根据当前节点集关联的两个值之间的比较对当前节点集进行过滤的规则段。

对偶度量段以括号“()”作为起止字符,其内部依次为左侧路径规则链、左侧度量符号、比较符号、右侧路径规则链、右侧度量符号,如“(->Name[Value] = ->ObjectType[Value])”。其中,当左侧度量符号或右侧度量符号为“[Value]”时,此度量符号可以省略。

对偶度量段执行时,从当前节点集中的每个节点出发,根据左侧路径规则链与左侧度量符号获取左侧值,根据右侧路径规则链与右侧度量符号获取右侧值,并根据比较符号计算左侧值与右侧值的比较结果确定当前节点是否保留。若查找到多个左侧值或右侧值,则当其中存在一组值使比较结果为真时保留当前节点。

对偶度量段示例见图 11。

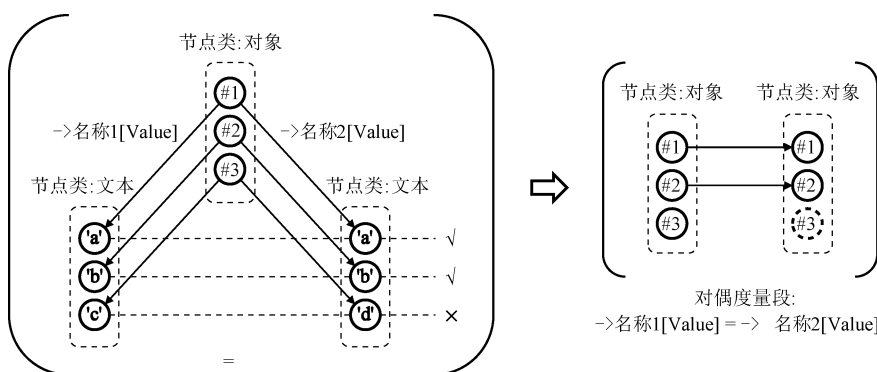


图 11 对偶度量段示例

9.3.19 复合属性段

复合属性段是将 1 至多个路径规则链的并联用括号“()”封装为一个规则段,其作用等价于一个属性段,表示从输入节点集出发到输出节点集的多条可选的子图路径。其中,多个路径规则链之间使用“OR”或“或”连接,表示对多条可选的子图路径的搜索结果求并集。

复合属性段的示例见图 12。

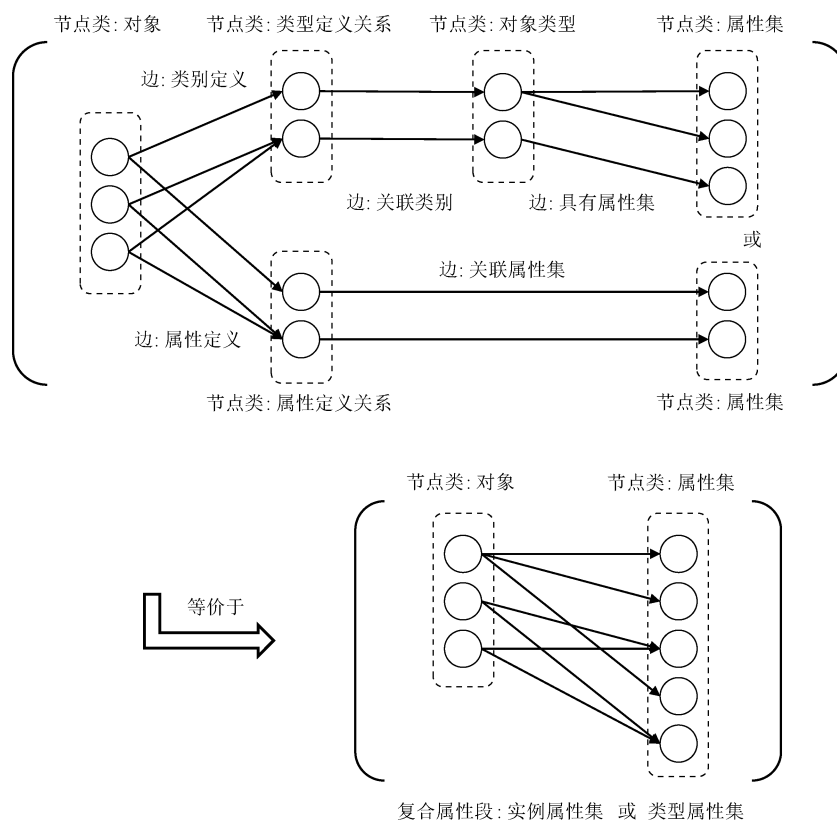


图 12 复合属性段示例

9.3.20 复合度量段

复合度量段是将 1 至多个度量规则链用括号“()”封装为一个规则段,其作用等价于一个度量段,表

示对输入节点集的一个复合的过滤条件。其中,多个度量规则链之间使用逻辑连接符号连接。复合度量段的示例见图 13。

在复合度量段内部的度量规则链的执行中,根据度量规则链末尾的度量段的过滤结果,回溯能通过过滤条件的子图路径,这些路径的起始节点的集合为输入节点集的一个子集,以此作为根据这条度量规则链对输入节点集过滤的结果。

复合度量段与度量段的一个区别是,度量段只能出现在规则链的结尾,而复合度量段可以出现在规则链的任意位置。

如果复合属性段内部的一条规则链以集合度量段符号结尾,那么规则链中此集合度量段之前的位置应存在属性段或复合属性段,否则集合度量段无法执行。

注:复合度量段中的逻辑连接符号的使用方法与约束同规则表达式(见 9.3.8),而区别是,规则表达式中每一个叶子项是一个具有根概念的规则语句,而复合度量段中每一个叶子项是一个不含根概念的度量规则链。

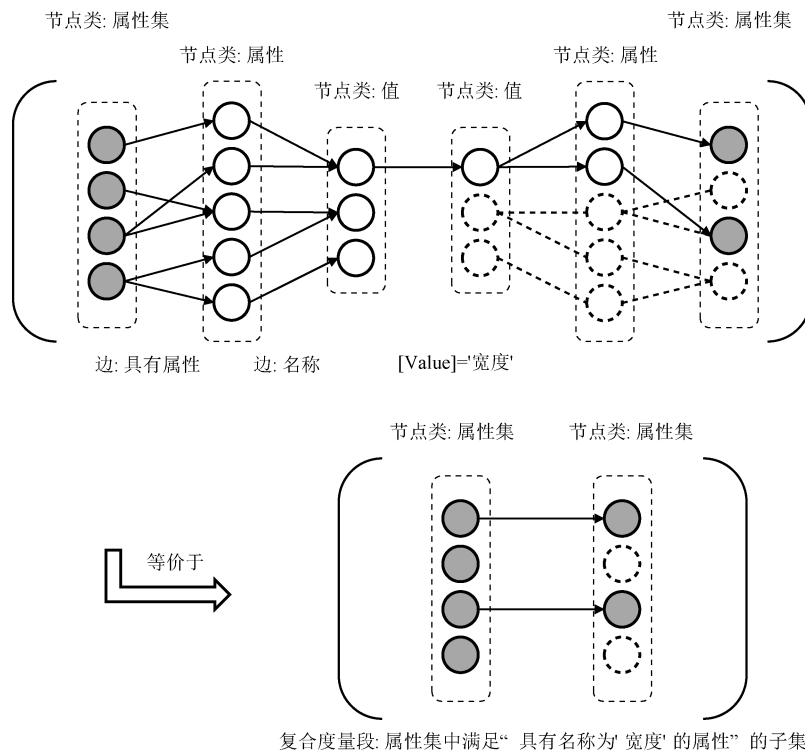


图 13 复合度量段示例

9.3.21 名称语法糖

名称语法糖是一种特殊的复合度量段。当复合度量段用于根据默认的名称路径查找名称值并过滤节点集时,可以使用名称语法糖的方式对此复合度量段进行简写,省略其中的“->默认名称[值]”部分。例如:复合度量段“(→默认名称[值]=‘ABCD’)”可简写为“(‘ABCD’)”。

为了使用名称语法糖,在规则集的头部应以缩写表达式的形式规定“默认名称”对应的规则段序列,例如:“默认名称 as ()->Name”。

9.4 标记规则

9.4.1 规则文件分块

规则文件分块是规则文件中的一个可选的结构,用于将规则文件分割为多个部分,以便于管理或

使用。

规则文件分块以分块起始标记“# BLOCK”或“# 块”开始,以分块终止标记“# ENDBLOCK”或“# 块结束”结束,其中可以包含若干正文块。

分块终止标记可以省略,此时以下一个分块起始标记之前或文件结尾作为此分块的结束。

规则文件分块应完整地覆盖内部的正文块,即分块起始标记与分块终止标记不能把定义块或约束块从中间截断。

9.4.2 标记项

标记项是一个由标记项标识符与一个标记值组成的语句,用于表示附加于 MVD 句型结构的标记信息,这些信息不会用于 MVD 执行,但是可用于提供关于便于管理的说明性信息,例如规则集的作者、目的、使用场景等信息,以及规则条目的分类、分级等信息。

- a) 标记项形如“# 标记项标识符 = 标记值”。其中标记值可以为字符串值、数值或逻辑值。例如: # 规则描述 = "消防系统的配色规则"。
- b) 当标记项标识符为“名称”或“Name”时,可省略标记项标识符,将标记项简写为“# 标记值”。此时当标记项的位置为头部标记时,表示整个规则集的名称;位置为表达式标记时,表示其后紧接的表达式的名称;位置为分块标记时,表示当前规则文件分块的名称。

9.4.3 标记项的位置

标记项按照位置分为头部标记、表达式标记、分块标记。

- a) 头部标记是位于规则文件开头的若干标记项,表示对整个规则集的描述信息。
- b) 表达式标记是出现在概念表达式或规则表达式之前的若干标记项,表示对此规则的标记信息。
- c) 分块标记是紧接于分块起始标记之后的若干标记项,表示对此规则文件分块的标记信息。

9.4.4 标记项的使用要求

标记标识符与标记值应具有准确性和易读性,能够准确表达规则集及其内部条目对应的场景或功能,便于用户理解。

当在同一个位置有多个标记项时,不可出现重复的标记标识符。

9.4.5 必需的标记项

以下标记项为规则集必需的标记信息,应当在头部标记中写入下列标记项,标记项标识符应采用中文标识符或英文标识符之一:

- a) 语言版本(Language_Version):规则集的语言版本。符合本文件所述语法的规则集版本应标记为"GBT 45393_3_2025";
- b) 标题(Title):规则集的标题;
- c) 信息模式(Information_Schema):规则集适用的信息模式及其版本,如:"IFC4"。当规则集适用于多个信息模式时,应在字符串内使用竖线(|)分隔,如:"IFC4|IFC2X3";
- d) 版本(Version):当前规则集的版本号;
- e) 最后编辑时间(Last_Edition_Time):当前规则集最后修改的时间,格式为“[年]-[月]-[日]T[时]:[分]:[秒][时区]”,如:"2023-02-22T17:17:54+08:00"。

9.4.6 推荐的标记项

9.4.6.1 一般规定

为了便于规则集以及其中的规则条目的管理与使用,规则集内宜加入下列推荐的标记项。当使用这些推荐的标记项时,标识符应采用中文标识符或英文标识符之一,其中与规则集整体相关的标记项应写入头部标记,与规则条目相关的标记项应写入表达式标记或分块标记。

9.4.6.2 推荐的头部标记项

推荐的头部标记项包括:

- a) 作者(Author):作者名称;
- b) 组织(Organization):编制规则集的组织机构;
- c) 邮箱(Email):作者联系方式邮箱;
- d) 描述(Description):规则集的说明文字;
- e) 状态(Status):当前规则集的发布状态;
- f) URI:当前规则集的唯一识别符或访问网址;
- g) 编辑工具(Authoring_Tool):用于编辑或发布此规则集的软件工具名称及其版本;
- h) 模型导出工具(Model_Exporting_Tool):规则集针对的 BIM 数据导出工具及其版本;
- i) 模型导入工具(Model_Importing_Tool):规则集针对的 BIM 数据导入工具及其版本;
- j) 创建时间(Creation_Time):规则集创建的时间;
- k) IDM 引用(IDM_Reference):引用规则集关联的 IDM;
- l) 用例引用(Use_Case_Reference):引用规则集关联的用例。

9.4.6.3 推荐的表达式标记项或分块标记项

推荐的表达式标记项或分块标记项包括:

- a) 名称(Name):规则的名称;
- b) 描述(Description):规则的说明文字;
- c) 用例(Use_Case):表示规则的使用场景的文字或编号;
- d) 等级(Level):表示规则的严重程度的文字,如“严重”“重要”“不要求”等;
- e) 来源(Source):规则的来源文本或引用;
- f) LOD:表示规则适用的 LOD 等级的文字或编号。

附 录 A
(规范性)
MVDLite 语法

A.1 MVDLite 词法规则

```
lexer grammar MVDLite_Lexer_CH;
```

```
BLOCK_COMMENT
```

```
: /*'.*?'/* -> skip;
```

```
LINE_COMMENT
```

```
: // ~[\r\n]* -> skip;
```

```
WS
```

```
: ('\t'|\n'|\r')+ -> skip;
```

```
START_BLOCK
```

```
: 'BLOCK' | '块';
```

```
END_BLOCK
```

```
: 'ENDBLOCK' | '块结束';
```

```
DEFINITION
```

```
: 'definition' | '定义';
```

```
CONSTRAINT
```

```
: 'constraint' | '约束';
```

```
AS
```

```
: 'as' | '作为';
```

```
EXTENDS
```

```
: 'extends' | '继承';
```

```
ITS
```

```
: '->' | '的';
```

```
AND
```

```
: 'AND' | '&' | '且';
```

```
OR
```

```
: 'OR' | '|' | '或';
```

```
XOR
```

```
: 'XOR' | '~' | '异或';
```

```
NOT
```

```
: 'NOT' | '!' | '不' | '非';
```

```
FALSE
```

```
: 'FALSE' | '假';
```

```
TRUE
```



```

:   'TRUE' | '真';
UNKNOWN
:   'UNKNOWN' | '未知';
NULL
:   'NULL' | '空';
SELF
:   'SELF' | '自身';

M_VALUE
:   [ ('Value' | '值') ];
M_SIZE
:   [ ('Size' | '数量') ];
M_TYPE
:   [ ('Type' | '类型') ];
M_EXISTS
:   [ ('Exists' | '存在') ];
M_UNIQUE
:   [ ('Unique' | '唯一') ];

EQUAL
:   '=';
NOT_EQUAL
:   '!=';
GREATER_THAN
:   '>';
GREATER_THAN_OR_EQUAL
:   '>=';
LESS_THAN
:   '<';
LESS_THAN_OR_EQUAL
:   '<=';

LPAREN
:   '(';
RPAREN
:   ')';
QUOTATION
:   '\ ' | '"';

FLOAT
:
SIMPLE_FLOAT
|

```




```

: ' STRING ;
STARTS_WITH_STRING
: ' STRING ;
ENDS_WITH_STRING
: ' STRING ;
NAME_STRING
: '$' STRING ;

NAME
: NAME_START_CHAR NAME_CHAR * ;

fragment
NAME_CHAR
: NAME_START_CHAR | DIGIT ;

fragment
NAME_START_CHAR
: UPPER | LOWER | '_' | CHCHAR ;

fragment
UPPER
: 'A..'Z' ;

fragment
LOWER
: 'a..'z' ;

fragment
CHCHAR
: '\u4e00..\u9fa5' ;

```

A.2 MVDLite 句型结构与标记规则

```

grammar MVDLite_Grammar;

import MVDLite_Lexer_CH;

file // 规则文件
:
tag *
abbr_expression *
(body | block) *
EOF
;

abbr_expression // 缩写表达式
:
identifier

```

```

AS
LPAREN (identifier)? RPAREN
mid_segment +
;

block // 规则文件分块
:
start_block
tag *
body *
end_block?
;

tag // 标记项
:
'#' identifier
|
'#' STRING
|
'#' identifier '=' single_value
;

start_block // 分块起始标记
:
'#' START_BLOCK
(identifier | STRING)?
;

end_block // 分块终止标记
:
'#' END_BLOCK
;

body // 正文块
:
body_definition | body_constraint
;

body_definition // 定义块
:
DEFINITION
(
tag *

```

```

    (concept_expression | rule_expression)
  ) *
;

body_constraint    // 约束块
:
CONSTRAINT
(
    tag *
    rule_expression
) *
;

concept_expression // 概念表达式
:
identifier EXTENDS identifier ( OR identifier ) *
;

rule_expression    // 规则表达式
:
rule_term (logical_operator rule_term) *
;

rule_term          // 规则语句
:
identifier metric_rule_chain
|
LPAREN rule_expression RPAREN
|
NOT rule_term
;

rule_chain         // 规则链
:
path_rule_chain
|
metric_rule_chain
;

metric_rule_chain  // 度量规则链
:
mid_segment * segment_for_metric
;

```



```

path_rule_chain    // 路径规则链
:
mid_segment * segment_for_path
;

rule_segment      // 规则段
:
attribute_segment | compound_attribute | metric_segment | dual_metric | compound_metric | string_sugar
;

mid_segment       // 可位于中间的规则段
:
attribute_segment | compound_attribute | dual_metric | compound_metric | string_sugar
;

non_mid_segment   // 不可位于中间的规则段
:
metric_segment
;

segment_for_metric // 度量规则链的结尾规则段
:
metric_segment | dual_metric | compound_metric | string_sugar
;

segment_for_path  // 路径规则链的结尾规则段
:
attribute_segment | compound_attribute
;

attribute_segment // 属性段
:
ITS identifier ( ':' identifier )?
|
ITS SELF
;

metric_segment    // 度量段
:
metric? operator values
;

dual_metric       // 对偶度量段
:

```

```

LPAREN
    path_rule_chain metric?
    operator
    path_rule_chain metric?
RPAREN
;

compound_metric    // 复合度量段
:
LPAREN inner_rule_expression RPAREN
;

inner_rule_expression    // 复合度量段的内部规则表达式
:
inner_rule_term (logical_operator inner_rule_term) *
;

inner_rule_term    // 复合度量段的内部规则语句
:
metric_rule_chain
|
LPAREN inner_rule_expression RPAREN
|
NOT inner_rule_term
;

compound_attribute    // 复合属性段
:
LPAREN path_rule_chain
    (OR path_rule_chain) *
RPAREN
;

string_sugar    // 名称语法糖
:
LPAREN string_literal RPAREN
;

metric    // 度量符号
:
M_VALUE | M_SIZE | M_TYPE | M_UNIQUE | M_EXISTS
;

operator    // 比较符号

```

```

:
EQUAL
| NOT_EQUAL
| GREATER_THAN
| GREATER_THAN_OR_EQUAL
| LESS_THAN
| LESS_THAN_OR_EQUAL
;

logical_operator    // 逻辑连接符号
:
AND | OR | XOR
;

values              // 值
:
single_value (OR single_value) *
|
LPAREN values RPAREN
;

single_value        // 单值
:
NULL
| string_literal
| logical_literal
| real_literal
| range_literal
;

logical_literal     // 逻辑值
:
TRUE | FALSE | UNKNOWN
;

real_literal        // 浮点数与整数
:
FLOAT | INT
;

range_literal       // 值域
:
real_literal '~' real_literal
|

```

```
real_literal '+' real_literal
;

string_literal // 字符串
:
STRING
| REGEX_STRING
| CONTAINS_STRING
| STARTS_WITH_STRING
| ENDS_WITH_STRING
;

identifier // 标识符
:
NAME | NAME_STRING
;
```



附录 B

(资料性)

适用于 IFC 的常用 MVDLite 规则模板

B.1 定义类规则模板

B.1.1 分类编码方式概念定义

MVDLite 语言表示的分类编码方式定义概念宜包含如下内容。

```
//头部 (Header):
类型 as (IfcObject)->IsTypedBy:IfcRelDefinesByType->RelatingType:IfcTypeObject
分类编码 as
  (IfcObject)->HasAssociations:IfcRelAssociatesClassification->RelatingClassification:IfcClassificationReference
(->ReferencedSource:IfcClassification->Name='{特定分类标准}')->Identification
分类项名称 as
  (IfcObject)->HasAssociations:IfcRelAssociatesClassification->RelatingClassification:IfcClassificationReference
(->ReferencedSource:IfcClassification->Name='{特定分类标准}')->Name

//正文 (Body):
定义
{概念名称} 继承 {父类概念名称} //父类概念可直接继承 IFC 类型
{概念名称}[类型] >= '{概念对应的 IFC 类型}' //(选填)
{概念名称}->分类编码='{概念分类编码}'
{概念名称}->分类项名称='{概念分类项名称}'
```

注 1: 其中“{}”表示用户在使用中需要补充的内容。

注 2: {特定分类标准}通常是已发布的分类编码标准、数据字典标准等,或模型中自定义的分类标准。

注 3: 本模板根据概念的分类编码或分类项定义模型构件,使用中两者选其一,用户需保证需要验证的信息模型构件包含准确的“分类编码”或“分类项名称”。

注 4: {概念分类编码}和{概念分类项名称}是模型中使用的对象标识,有多个时使用“|”隔开。

示例:

```
定义
防火墙 继承 IfcWall
防火墙->分类项名称='防火外墙' | '防火隔墙'
```

B.1.2 特定属性值方式概念定义

MVDLite 语言表示的特定属性值方式定义概念模板宜包含如下内容。

```
//头部 (Header):
类型 as (IfcObject)->IsTypedBy:IfcRelDefinesByType->RelatingType:IfcTypeObject
实例属性集 as (IfcObject)->IsDefinedBy->RelatingPropertyDefinition:IfcPropertySet
类型属性集 as (IfcObject)->类型->HasPropertySets:IfcPropertySet
任意属性集 as (IfcObject)(->实例属性集 OR ->类型属性集)
单值属性 as (IfcPropertySet)->HasProperties:IfcPropertySingleValue
属性值 as (IfcPropertySingleValue)->NominalValue:IfcValue
```

//正文(Body):

定义

{概念名称} 继承 {父类概念名称}

{概念名称}->{属性集类型}->单值属性({特定属性名称})->属性值{比较符号}{特定属性值}

注 1: 其中“{}”表示用户在使用中需要补充的内容。

注 2: {属性集类型}选项为“实例属性集”“类型属性集”或“任意属性集”,其中任意属性集表示实例属性集和类型属性集均可。

注 3: {特定属性值}填写时,数值形如“1”,字符串形如“甲级”,有多个时使用“|”隔开。

示例:

定义

防火墙 继承 IfcWall

防火墙->类型属性->单值属性(是否防火墙)->属性值=TRUE

B.1.3 比较两个属性值方式概念定义

MVDLite 语言表示的比较属性值方式定义概念模板宜包含如下内容。

//头部(Header):

类型 as (IfcObject)->IsTypedBy:IfcRelDefinesByType->RelatingType:IfcTypeObject

实例属性集 as (IfcObject)->IsDefinedBy->RelatingPropertyDefinition:IfcPropertySet

类型属性集 as (IfcObject)->类型->HasPropertySets:IfcPropertySet

任意属性集 as (IfcObject)(->实例属性集 OR ->类型属性集)

单值属性 as (IfcPropertySet)->HasProperties:IfcPropertySingleValue

属性值 as (IfcPropertySingleValue)->NominalValue:IfcValue

//正文(Body):

定义

{概念名称} 继承 {父类概念名称}

{概念名称}(->{属性集类型}->单值属性({属性名称 1})->属性值 {比较符号} ->{属性集类型}->单值属性({属性名称 2})->属性值)

注 1: 其中“{}”表示用户在使用中需要补充的内容。

注 2: {属性集类型}选项为“实例属性集”“类型属性集”或“任意属性集”,其中任意属性集表示实例属性集和类型属性集均可。

注 3: 用于比较属性值的两个{属性名称},其计量单位保持一致。

示例:

定义

防火墙 继承 IfcWall

防火墙(->实例属性->单值属性(耐火极限)->属性值 >= ->类型属性->单值属性(耐火极限阈值)->属性值)



B.1.4 命名规则方式概念定义

MVDLite 语言表示的命名规则方式定义概念模板宜包含如下内容。

```
//头部 (Header):
类型 as (IfcObject)->IsTypedBy:IfcRelDefinesByType->RelatingType:IfcTypeObject
默认名称 as ()(->Name OR ->LongName)
类型名称 as (IfcObject)->类型->默认名称

//正文 (Body):
定义
{概念名称} 继承 {父类概念名称}
{概念名称}->{名称位置}=r{概念名称正则表达式}
{概念名称}->{名称位置}=c{概念名称必含字符}
{概念名称}->{名称位置}=s{概念名称起始字符}
{概念名称}->{名称位置}=e{概念名称结束字符}
{概念名称}[类型] >= '{概念对应的 IFC 类型}' //(选填)
```

注 1: 其中“{}”表示用户在使用中需要补充的内容。

注 2: 本模板根据概念命名规则进行定义,包括正则表达式、包含字符、起始字符和结束字符四类语句,根据实际情况选择一个或多个规则进行概念定义。

注 3: r{概念名称正则表达式}、c{概念名称必含字符}、s{概念名称起始字符}、e{概念名称结束字符}中,r、c、s、e 后不接空格。

注 4: {名称位置}选项为“默认名称”“类型名称”。

示例:

```
定义
防火墙 继承 IfcWall
防火墙->类型名称=c"防火"
```

B.2 约束类规则模板

B.2.1 模型结构

模型结构约束包括基点坐标约束、计量单位约束和空间包含约束。

a) 基点坐标约束

MVDLite 语言表示的模型基点坐标约束的模板宜包含如下内容。

```
//正文 (Body):
定义
项目坐标变换 继承 IfcMapConversion

约束
//项目基点相关坐标信息需存在
项目坐标变换[存在]=TRUE
//项目基点东西坐标应满足要求
项目坐标变换->Eastings={坐标值下限}~{坐标值上限}
```

```
//项目基点北南坐标应满足要求
项目坐标变换->Northings={坐标值下限}~{坐标值上限}
//项目基点高程基准应满足要求要求
项目坐标变换->OrthogonalHeight={坐标值下限}~{坐标值上限}
//项目基点正北方向偏角需存在
(项目坐标变换->XAxisAbscissa !=) 且 (项目坐标变换->XAxisOrdinate !=)
```

注 1: {坐标值下限}、{坐标值上限}由用户在使用中补充数值,其取值范围与单位按照项目规定采用特定的坐标与高程标准(如 2000 国家大地坐标系、1985 国家高程基准),并符合当地实际情况。

注 2: “!=”与“[存在]=TRUE”的区别在于,“[存在]=TRUE”的含义是“要求值存在,但是可以为空字符串”;而“!=”的含义是“要求值存在且不为空字符串”;“!=”也用于数值与逻辑值的存在规则,因为数值与逻辑值一定不为空字符串。

示例:

```
约束
项目坐标变换->OrthogonalHeight=0.0~300.0
```

b) 计量单位约束

MVDLite 语言表示的模型计量单位规则模板宜包含如下内容。

```
//正文(Body):
约束
//单位为标准单位,无单位前缀
IfcProject->UnitsInContext->Units(->UnitType='{计量类型}' 且 ->Prefix[存在]=FALSE 且 ->Name='{标准单位}')[存在]=TRUE
//单位为非标准单位,有单位前缀
IfcProject->UnitsInContext->Units(->UnitType='{计量类型}' 且 ->Prefix='{单位前缀}' 且 ->Name='{标准单位}')[存在]=TRUE
```

注 1: 其中“{}”表示用户在使用中需要补充的内容。

注 2: 模型计量单位分为标准单位和非标准单位,标准单位如米、克等,非标准单位为毫米、千克等。

注 3: 计量类型如长度计量表示为LENGTHUNIT,面积计量表示为AREAUNIT;基础单位如METRE、SQUARE_METRE等;导出单位使用前缀表示,如MILLI、KILO。适用于 GB/T 51447 的计量类型与基础单位见表 B.1,导出单位前缀见表 B.2。

注 4: 对于长度单位,在项目级别的数据交付中使用毫米作为单位,而涉及大地坐标、高程、里程等时使用米作为单位。

示例:

```
约束
IfcProject->UnitsInContext->Units(->UnitType='LENGTHUNIT' 且 ->Prefix='MILLI' 且 ->Name='METRE')[存在]=TRUE
```

表 B.1 适用于 GB/T 51447 的计量类型与基础单位

计量类型	单位类型	基础单位	中文名称
LENGTHUNIT	长度	METRE	米
AREAUNIT	面积	SQUARE_METRE	平方米
VOLUMEUNIT	体积	CUBIC_METRE	立方米
MASSUNIT	质量	GRAM	克
TIMEUNIT	时间	SECOND	秒
FREQUENCYUNIT	频率	HERTZ	赫兹
THERMODYNAMICTEMPERATUREUNIT	温度	DEGREE_CELSIUS	摄氏度
ELECTRICCURRENTUNIT	电流	AMPERE	安
ELECTRICVOLTAGEUNIT	电压	VOLT	伏
POWERUNIT	功率	WATT	瓦
FORCEUNIT	力	NEWTON	牛
ILLUMINANCEUNIT	光照强度	LUX	勒克斯
LUMINOUSFLUXUNIT	光通量	LUMEN	流明
LUMINOUSINTENSITYUNIT	发光强度	CANDELA	坎
PRESSUREUNIT	压强	PASCAL	帕

表 B.2 适用于 GB/T 51447 的导出单位前缀

导出单位前缀	中文名称	相比基础单位的倍率
EXA	艾	10^{18}
PETA	拍	10^{15}
TERA	太	10^{12}
GIGA	吉	10^9
MEGA	兆	10^6
KILO	千	10^3
HECTO	百	10^2
DECA	十	10
DECI	分	10^{-1}
CENTI	厘	10^{-2}
MILLI	毫	10^{-3}
MICRO	微	10^{-6}

表 B.2 适用于 GB/T 51447 的导出单位前缀 (续)

导出单位前缀	中文名称	相比基础单位的倍率
NANO	纳	10 ⁻⁹
PICO	皮	10 ⁻¹²
FEMTO	飞	10 ⁻¹⁵
ATTO	阿	10 ⁻¹⁸

c) 空间包含约束

MVDLite 语言表示的空间包含约束的模板宜包含如下内容。

```

//正文(Body):
定义
地形 继承 IfcSite
建筑 继承 IfcBuilding
空间 继承 IfcSpace
构件 继承 IfcElement

约束
//地形需要关联建筑
地形->IsDecomposedBy->RelatedObjects[存在]=TRUE
//建筑需要关联楼层
建筑->IsDecomposedBy->RelatedObjects[存在]=TRUE
//空间需要关联楼层
空间->Decomposes->RelatingObject[存在]=TRUE
//构件(非开洞)需要包含在空间或楼层内,或者关联到父构件
构件->ContainedInStructure->RelatingStructure[存在]=TRUE
或
构件->Decomposes->RelatingObject[存在]=TRUE
    
```

注: 其中楼层关联规则针对建筑领域。

B.2.2 对象

对象约束包含对象标识约束、对象存在性约束和对象命名约束。

a) 对象标识约束

MVDLite 语言表示的对象标识约束规则是对模型中的对象标识概念定义是否符合要求进行检查,其模板宜包含如下内容。

```

//头部(Header):
分类编码 as (IfcObject)->HasAssociations:IfcRelAssociatesClassification->RelatingClassification:IfcClassificationReference(->ReferencedSource:IfcClassification->Name='{特定分类标准}')->Identification
分类项名称 as (IfcObject)->HasAssociations:IfcRelAssociatesClassification->RelatingClassification:IfcClassificationReference(->ReferencedSource:IfcClassification->Name='{特定分类标准}')->Name 类型 as (IfcObject)->IsTypedBy:IfcRelDefinesByType->RelatingType:IfcTypeObject
    
```

```

实例属性集 as (IfcObject)->IsDefinedBy->RelatingPropertyDefinition;IfcPropertySet
类型属性集 as (IfcObject)->类型->HasPropertySets;IfcPropertySet
任意属性集 as (IfcObject)(->实例属性集 OR ->类型属性集)
默认名称 as ()(->Name OR ->LongName)
单值属性 as (IfcPropertySet)->HasProperties;IfcPropertySingleValue
属性值 as (IfcPropertySingleValue)->NominalValue;IfcValue 类型标签一 as (IfcObject)->分类编码
类型标签二 as (IfcObject)->分类项名称
类型标签三 as (IfcObject) (->任意属性集->单值属性(‘特定属性名称’) ->属性值)
类型标签四 as (IfcObject) ->默认名称
类型标签 as (IfcObject)(->类型标签一 OR ->类型标签二 OR ->类型标签三 OR ->类型标签四)

//正文(Body):
定义
建筑 继承 IfcBuilding
空间 继承 IfcSpace
构件 继承 IfcElement

约束
//建筑标识存在且满足要求
建筑->{类型标签选项}={对象标识枚举}
//空间标识存在且满足要求
空间->{类型标签选项}={对象标识枚举}
//构件标识存在且满足要求
构件->{类型标签选项}={对象标识枚举}

```

注 1: 其中“{}”表示用户在使用中需要补充的内容。

注 2: {类型标签选项}内容是 header 里已经写入的缩写标识符,如类型标签一、类型标签二等。如对象标识通过分类编码或分类项定义,则选类型标签一或标签二;如对象标识通过特定属性定义,则选类型标签三,如对象标识通过名称定义,则选类型标签四。当用户有其他描述类型标签的方式时,在 header 中追加更多的缩写表达式。

注 3: {对象标识枚举}是模型中使用的全部对象标识,有多个时使用“|”隔开。

注 4: 当仅检查标识是否存在时,将“={对象标识枚举}”替换为“! =”。

注 5: 其他相关要求参考 B.1。

示例:

```

约束
建筑->类型标签='住宅建筑'|'商业建筑'
防火墙(->类型标签一 或 ->类型标签三)='防火外墙'|'防火隔墙'

```

b) 对象存在性约束

MVDLite 语言表示的对象存在性规则模板宜包含如下内容。



```
//正文(Body):
约束
//对象应存在
{概念名称}[存在] = TRUE
//对象不应存在
{概念名称}[存在] = FALSE
```

注：其中“{}”表示用户在使用中需要补充的内容。

示例：

```
约束
防火墙[存在] = TRUE
```

c) 对象命名约束

MVDLite 语言表示的概念命名约束规则模板宜包含如下内容。

```
//头部 (Header):
类型 as (IfcObject)->IsTypedBy:IfcRelDefinesByType->RelatingType:IfcTypeObject
默认名称 as ()(->Name OR ->LongName)
类型名称 as (IfcObject)->类型->默认名称

//正文(Body):
约束
//对象的命名应满足要求
{概念名称}->{名称位置} = r{概念名称正则表达式}
{概念名称}->{名称位置} = c{概念名称必含字符}
{概念名称}->{名称位置} = s{概念名称起始字符}
{概念名称}->{名称位置} = e{概念名称结束字符}
```

注 1：其中“{}”表示用户在使用中需要补充的内容。

注 2：{概念名称}为 IFC 已定义的概念，如楼层(IfcBuildingStorey)、轴网(IfcGrid)等，或为自定义概念名称。

注 3：r{概念名称正则表达式}、c{概念名称必含字符}、s{概念名称起始字符}、e{概念名称结束字符}中，r、c、s、e 后不接空格。

注 4：{名称位置}选项为“默认名称”“类型名称”。

示例：

```
约束
防火墙->类型名称 = c防火
楼层->默认名称 = r^(F|B)\d+ $
```

B.2.3 可视化

可视化约束包含包含材质贴图约束、材料组成约束和颜色设置约束。

a) 材质贴图约束

MVDLite 语言表示的构件材质贴图约束规则模板宜包含如下内容。

```

//头部 (Header):
类型 as (IfcObject)->IsTypedBy:IfcRelDefinesByType->RelatingType:IfcTypeObject
简单几何表达 as
(IFcProduct)->Representation:IfcProductDefinitionShape->Representations:IfcShapeRepresentation
映射几何表达 as (IfcProduct)->简单几何表达
->Items:IfcMappedItem->MappingSource:IfcRepresentationMap->MappedRepresentation:IfcShapeRepresenta-
tion
类型几何表达 as (IfcProduct)->类型->RepresentationMaps->MappedRepresentation:IfcShapeRepresentation
组装几何表达 as (IfcProduct)->IsDecomposedBy->RelatedObjects(->简单几何表达 OR ->映射几何表达)
几何表达 as (IfcProduct)(->简单几何表达 OR ->映射几何表达 OR ->类型几何表达 OR ->组装几何表达)
表面几何 as (IfcElement)->HasCoverings:IfcCovering->几何表达
几何 as (IfcElement)(->几何表达 OR ->表面几何 OR ->组装几何表达)
网格几何 as (IfcElement)->几何->Items:IfcTessellatedFaceSet
非网格几何 as (IfcElement)->几何(->Items:IfcRepresentationItem[Type]! =IfcTessellatedFaceSet)->SELF
几何样式 as (IfcRepresentationItem)->StyledByItem:IfcStyledItem
直接表面样式 as (IfcStyledItem)->Styles:IfcSurfaceStyle
间接表面样式 as (IfcStyledItem)->Styles:IfcPresentationStyleAssignment->Styles:IfcSurfaceStyle
表面材质贴图 as (IfcSurfaceStyle)->Styles:IfcSurfaceStyleWithTextures->Textures:IfcSurfaceTexture
几何材质贴图 as (IfcRepresentationItem)->几何样式(->直接表面样式 OR ->间接表面样式)->表面材质
贴图

//正文 (Body):
约束
//构件应有材质贴图和明确的 UV 坐标
{概念名称}->网格几何
(
(->几何材质贴图[存在]=TRUE)
且
(->HasTextures:IfcIndexedTextureMap->Maps:IfcSurfaceTexture[存在]=TRUE)
)
// 构件应有材质贴图
{概念名称}->非网格几何->几何材质贴图[存在]=TRUE
或
{概念名称}->网格几何
(
(->几何材质贴图[存在]=TRUE)
且
(->HasTextures:IfcIndexedTextureMap->Maps:IfcSurfaceTexture[存在]=TRUE)
)

```

注：其中“{}”表示用户在使用中需要补充的内容。

示例：

```

约束
幕墙嵌板->网格几何
(
    (->几何材质贴图[存在]=TRUE)
    且
    (->HasTextures:IfcIndexedTextureMap->Maps:IfcSurfaceTexture[存在]=TRUE)
)
    
```



b) 材料组成约束

MVDLite 语言表示的构件材料组成约束规则模板宜包含如下内容。

```

//头部 (Header):
材料关联 as (IfcObjectDefinition)->HasAssociations:IfcRelAssociatesMaterial
普通材料 as (IfcObjectDefinition)->材料关联->RelatingMaterial:IfcMaterial
材料成分 as (IfcObjectDefinition)->材料关联
->RelatingMaterial:IfcMaterialConstituent->Material:IfcMaterial
材料成分集 as (IfcObjectDefinition)->材料关联
->RelatingMaterial:IfcMaterialConstituentSet->MaterialConstituents:IfcMaterialConstituent->Material:IfcMa-
terial
材料层 as (IfcObjectDefinition)->材料关联->RelatingMaterial:IfcMaterialLayer->Material:IfcMaterial
材料层集 as (IfcObjectDefinition)->材料关联
->RelatingMaterial:IfcMaterialLayerSet->MaterialLayers:IfcMaterialLayer->Material:IfcMaterial
材料截面 as (IfcObjectDefinition)->材料关联
->RelatingMaterial:IfcMaterialProfile->Material:IfcMaterial
材料截面集 as (IfcObjectDefinition)->材料关联
->RelatingMaterial:IfcMaterialProfileSet->MaterialProfiles:IfcMaterialProfile->Material:IfcMaterial
材料层用途 as (IfcObjectDefinition)->材料关联
->RelatingMaterial:IfcMaterialLayerSetUsage->ForLayerSet:IfcMaterialLayerSet->MaterialLayers:IfcMateri-
alLayer->Material:IfcMaterial
材料截面用途 as (IfcObjectDefinition)->材料关联
->RelatingMaterial:IfcMaterialProfileSetUsage->ForProfileSet:IfcMaterialProfileSet->MaterialProfiles:IfcMa-
terialProfile->Material:IfcMaterial
实例属性集 as (IfcObject)->IsDefinedBy->RelatingPropertyDefinition:IfcPropertySet
类型属性集 as (IfcObject)->IsTypedBy->RelatingType->HasPropertySets:IfcPropertySet
任意属性集 as (IfcObject)(->实例属性集 OR ->类型属性集)
默认名称 as ()(->Name OR ->LongName)
单值属性 as (IfcPropertySet)->HasProperties:IfcPropertySingleValue
属性值 as (IfcPropertySingleValue)->NominalValue:IfcValue
材料属性 as (IfcObject)->任意属性集->任意属性(‘{材料属性名称}’)->属性值

//材料组成以材料关系定义时,使用此模板
材料的关系表示 as (IfcObjectDefinition)(->普通材料 OR ->材料成分 OR ->材料成分集 OR ->材料层 OR
->材料层集 OR ->材料截面 OR ->材料截面集 OR ->材料层用途 OR ->材料截面用途)

//材料组成以属性方式定义时,使用此模板
任意的属性表示 as (IfcObjectDefinition) ->材料属性

任意材料表示 as (IfcObjectDefinition)(->材料的关系表示 OR ->材料的属性表示)
    
```

```
//正文(Body):
约束
//构件应有材料组成
{概念名称}->任意材料表示->默认名称[Value]! ='
```

注 1: 其中“{}”表示用户在使用中需要补充的内容。

注 2: 约束中的“! =”根据需要换为材料组成的要求枚举, 有多个时使用“|”隔开。

示例:

```
约束
结构柱->任意材料表示->默认名称[Value]! ='
```

c) 颜色设置约束

MVDLite 语言表示的颜色设置规则模板宜包含如下内容。

```
//头部 (Header):
类型 as (IfcObject)->IsTypedBy:IfcRelDefinesByType->RelatingType:IfcTypeObject
简单几何表达 as
(IfcProduct)->Representation;IfcProductDefinitionShape->Representations;IfcShapeRepresentation
映射几何表达 as (IfcProduct)->简单几何表达
->Items;IfcMappedItem->MappingSource;IfcRepresentationMap->MappedRepresentation;IfcShapeRepresentation
类型几何表达 as (IfcProduct)->类型->RepresentationMaps->MappedRepresentation;IfcShapeRepresentation
组装几何表达 as (IfcProduct)->IsDecomposedBy->RelatedObjects(->简单几何表达 OR ->映射几何表达)
几何表达 as (IfcProduct)(->简单几何表达 OR ->映射几何表达 OR ->类型几何表达 OR ->组装几何表达)
表面几何 as (IfcElement)->HasCoverings;IfcCovering->几何表达
几何 as (IfcElement)(->几何表达 OR ->表面几何 OR ->组装几何表达)
网格几何 as (IfcElement)->几何->Items;IfcTessellatedFaceSet
非网格几何 as (IfcElement)->几何(->Items;IfcRepresentationItem[Type]! =IfcTessellatedFaceSet)->SELF
几何样式 as (IfcRepresentationItem)->StyledByItem;IfcStyledItem
直接表面样式 as (IfcStyledItem)->Styles;IfcSurfaceStyle
间接表面样式 as (IfcStyledItem)->Styles;IfcPresentationStyleAssignment->Styles;IfcSurfaceStyle
表面颜色 as (IfcSurfaceStyle)->Styles;IfcSurfaceStyleRendering->SurfaceColour;IfcColourRgb
几何颜色 as (IfcRepresentationItem)->几何样式(->直接表面样式 OR ->间接表面样式)->表面颜色
一般几何颜色 as (IfcElement)->几何->Items;IfcRepresentationItem->几何颜色
网格几何颜色 as (IfcElement)->网格几何
->HasColours;IfcIndexedColourMap->Colours;IfcColourRgbList->ColourList

//正文(Body):
约束
//配色 RGB 值应满足要求
{概念名称}(->一般几何颜色(->Red={颜色 R 值/255}+- 1E-3 且 ->Green={颜色 G 值/255}+- 1E-3 且
->Blue={颜色 B 值/255}+- 1E-3))
或
{概念名称}(->网格几何颜色(->ITEM_1={颜色 R 值/255}+- 1E-3 且 ->ITEM_2={颜色 G 值/255}+-
1E-3 且 ->ITEM_3={颜色 B 值/255}+- 1E-3))
```

注 1：其中“{}”表示用户在使用中需要补充的内容。

注 2：IFC 中使用[0.0,1.0]范围内的浮点数记录颜色，因此需将常用的[0,255]范围的整数 RGB 值映射到浮点数，并考虑浮点误差范围，故使用 RGB 值/255+- 1E-3。

示例：

```

约束
// 使用(120, 100, 255)设置颜色
排水管线(->一般几何颜色(->Red= 0.471 +- 1E-3 且 ->Green= 0.392 +- 1E-3 且 ->Blue= 1.0 +- 1E-3))
或
排水管线(->网格几何颜色(->ITEM_1= 0.471 +- 1E-3 且 ->ITEM_2= 0.392 +- 1E-3 且 ->ITEM_3= 1.0 +- 1E-3))
    
```

B.2.4 属性

属性约束包含包含属性存在性约束和属性取值约束。

a) 属性存在性约束

MVDLite 语言表示的属性存在性规则模板宜包含如下内容。

```

//头部 (Header):
类型 as (IfcObject)->IsTypedBy:IfcRelDefinesByType->RelatingType:IfcTypeObject
实例属性集 as (IfcObject)->IsDefinedBy->RelatingPropertyDefinition:IfcPropertySet
类型属性集 as (IfcObject)->类型->HasPropertySets:IfcPropertySet
任意属性集 as (IfcObject)(->实例属性集 OR ->类型属性集)
默认名称 as ()(->Name OR ->LongName)
单值属性 as (IfcPropertySet)->HasProperties:IfcPropertySingleValue
属性值 as (IfcPropertySingleValue)->NominalValue:IfcValue

//正文(Body):
约束
//对象应具有属性
{概念名称}->{属性集类型}->单值属性(<属性名称>)->属性值! =
    
```

注 1：其中“{}”表示用户在使用中需要补充的内容。

注 2：{属性集类型}选项为“实例属性集”“类型属性集”或“任意属性集”，其中任意属性集表示实例属性集和类型属性集均可。

示例：

```

约束
防火墙->类型属性集(<尺寸标注>)->单值属性(<厚度>)->属性值! =
    
```



b) 属性取值约束

MVDLite 语言表示的概念属性约束规则模板宜包含如下内容。

```

//头部 (Header):
类型 as (IfcObject)->IsTypedBy:IfcRelDefinesByType->RelatingType:IfcTypeObject
实例属性集 as (IfcObject)->IsDefinedBy->RelatingPropertyDefinition:IfcPropertySet
类型属性集 as (IfcObject)->类型->HasPropertySets:IfcPropertySet
任意属性集 as (IfcObject)(->实例属性集 OR ->类型属性集)
默认名称 as ()(->Name OR ->LongName)
单值属性 as (IfcPropertySet)->HasProperties:IfcPropertySingleValue
属性值 as (IfcPropertySingleValue)->NominalValue:IfcValue

//正文 (Body):
约束
//对象的属性值域应满足要求
{概念名称}->{属性集类型}->单值属性({属性名称})->属性值{取值范围}
//对象的属性值类型应满足要求
{概念名称}->{属性集类型}->单值属性({属性名称})->属性值[Type]>={值类型要求}

```

注 1: 其中“{}”表示用户在使用中需要补充的内容。

注 2: {属性集类型}选项为“实例属性集”“类型属性集”或“任意属性集”,其中任意属性集表示实例属性集和类型属性集均可。

注 3: 属性约束规则的前提是属性存在。

注 4: {取值范围}分为数值约束和枚举值约束:

数值约束时使用“>”(大于某值)、“<”(小于某值)、“>=”(大于或等于某值)、“<=”(小于或等于某值)、“!=”(不等于某值)、“=”(等于某值),当多个数值约束时,根据情况使用“且”“或”连接,并用半角括号括起,例如属性取值大于 0 且小于或等于 5 表示为(>0 且<=5);

枚举值约束时表示方式为={可选值 1}|{可选值 2}|{可选值 n},例如属性取值范围为“甲级”“乙级”“丙级”时,表示为=甲级|乙级|丙级。

注 5: {值类型要求}为 IFC 支持的具体值类型,其中“>=”表示左侧类型是右侧类型的一个子类或是相同类型;详见 9.2.11。填写符合 GB/T 51447 规定的值类型,或者上述值类型在 GB/T 16656.11 中对应的简单数据类型。部分常用的值类型要求示例见表 B.3,在 GB/T 16656.11 中,INTEGER 表示值类型为整数,REAL 表示值类型为浮点数,NUMBER 表示值类型为整数或浮点数。

示例:

```

约束
防火墙->类型属性集->单值属性(防火等级)->属性值[Type]>=IfcText
防火墙->类型属性集->单值属性(防火等级)->属性值=A 级|B 级|C 级|D 级

```

表 B.3 部分常用的值类型要求

值类型	GB/T 51447 中的值类型	GB/T 16656.11 中的简单数据类型
文字	IfcLabel、IfcText、IfcDateTime、……	STRING
整数值	IfcInteger、IfcDimensionCount、……	INTEGER
	IfcCountMeasure	NUMBER

表 B.3 部分常用的值类型要求 (续)

值类型	GB/T 51447 中的值类型	GB/T 16656.11 中的简单数据类型
无单位实数值	IfcReal	REAL
	IfcNumericMeasure	NUMBER
有单位度量值	IfcLengthMeasure、IfcEnergyMeasure、……	REAL
逻辑值	IfcBoolean	BOOLEAN
	IfcLogical	LOGICAL
二进制值	IfcBinary	BINARY

B.2.5 关系

关系约束包含包含系统设置约束和连接关系约束。

a) 系统设置约束

MVDLite 语言表示的系统设置约束规则模板宜包含如下内容。

```

//头部 (Header):
类型 as (IfcObject)->IsTypedBy:IfcRelDefinesByType->RelatingType:IfcTypeObject
实例属性集 as (IfcObject)->IsDefinedBy->RelatingPropertyDefinition:IfcPropertySet
类型属性集 as (IfcObject)->类型->HasPropertySets:IfcPropertySet
任意属性集 as (IfcObject)(->实例属性集 OR ->类型属性集)
默认名称 as ()(->Name OR ->LongName)
单值属性 as (IfcPropertySet)->HasProperties:IfcPropertySingleValue
属性值 as (IfcPropertySingleValue)->NominalValue:IfcValue
系统类型 as
(IfcDistributionElement)->HasAssignments:IfcRelAssignsToGroup->RelatingGroup:IfcSystem->ObjectType
//系统设置模板兼容了以系统关系和属性两种方式定义的规则,当有具体要求以某种方式表达系统设置时,可
两者选其一
系统设置 as (IfcDistributionElement)(->系统类型 OR ->任意属性集->单值属性(系统类型)->属性值)

//正文(Body):
约束
//构件应具有系统类型
{概念名称}->系统设置! =
    
```

注 1: 其中“{}”表示用户在使用中需要补充的内容。

注 2: 约束中的“! =”根据需要换为材质组成的要求枚举,有多个时使用“|”隔开。

b) 连接关系约束

MVDLite 语言表示的连接关系约束规则模板宜包含如下内容。

```

//头部 (Header):
填充洞口 as (IfcElement)->FillsVoids:IfcRelFillsElement->RelatingOpeningElement
开口于 as (IfcOpeningElement)->VoidsElements:IfcRelVoidsElement->RelatingBuildingElement
    
```

```

有洞口 as (IfcElement)->HasOpenings;IfcRelVoidsElement->RelatedOpeningElement
有填充 as (IfcOpeningElement)->HasFillings;IfcRelFillsElement->RelatedBuildingElement
连接到 1 as (IfcElement)->ConnectedTo;IfcRelConnectsElements->RelatedElement
连接到 2 as (IfcElement)->IsNestedBy;IfcRelNests
->RelatedObjects;IfcDistributionPort
->ConnectedTo;IfcRelConnectsPorts
->RelatedPort;IfcDistributionPort
->Nests;IfcRelNests
->RelatingObject
连接自 1 as (IfcElement)->ConnectedFrom;IfcRelConnectsElements->RelatingElement
连接自 2 as (IfcElement)->IsNestedBy;IfcRelNests
->RelatedObjects;IfcDistributionPort
->ConnectedFrom;IfcRelConnectsPorts
->RelatingPort;IfcDistributionPort
->Nests;IfcRelNests
->RelatingObject
连接到 as (IfcElement)(->连接到 1 | ->连接到 2)
连接自 as (IfcElement)(->连接自 1 | ->连接自 2)
连接 as (IfcElement)(->连接自 OR ->连接到)
被交叠 as (IfcElement)->IsInterferedByElements;IfcRelInterferesElements->RelatingElement
交叠并剪切 as (IfcElement)->InterferesElements;IfcRelInterferesElements->RelatedElement

//正文(Body):
约束
//{概念名称}应与其他构件具有连接关系
{概念名称}->连接关系[存在]=TRUE
//{概念名称 1}应具有指定的关系至{概念名称 2}
{概念名称 1}->{关系名称}[Type]>='{概念名称 2}'

```

注 1: 其中“{}”表示用户在使用中需要补充的内容。

注 2: 连接关系检查主要针对 MEP 设备,所以其中的{概念名称}的类型一般为 IfcDistributionElement。

注 3: {关系名称}选项为 header 里已经写入的缩写标识符,如填充洞口、开口于、有洞口等。

示例:

```

约束
IfcValve->连接关系[存在]=TRUE
IfcValve->连接到[Type]>='IfcPipeSegment'

```



B.2.6 几何

MVDLite 语言表示的几何类型约束规则模板宜包含如下内容。

```

//头部 (Header):
类型 as (IfcObject)->IsTypedBy:IfcRelDefinesByType->RelatingType:IfcTypeObject
简单几何表达 as
(IfcProduct)->Representation:IfcProductDefinitionShape->Representations:IfcShapeRepresentation
映射几何表达 as (IfcProduct)->简单几何表达
->Items:IfcMappedItem->MappingSource:IfcRepresentationMap->MappedRepresentation:IfcShapeRepresenta-
tion
类型几何表达 as (IfcProduct)->类型->RepresentationMaps->MappedRepresentation:IfcShapeRepresentation
组装几何表达 as (IfcProduct)->IsDecomposedBy->RelatedObjects(->简单几何表达 OR ->映射几何表达)
几何表达 as (IfcProduct)(->简单几何表达 OR ->映射几何表达 OR ->类型几何表达 OR ->组装几何表达)
表面几何 as (IfcElement)->HasCoverings:IfcCovering->几何表达
几何 ID as (IfcElement) (->几何表达 OR ->表面几何)->RepresentationIdentifier:IfcLabel
几何 Type as (IfcElement) (->几何表达 OR ->表面几何)->RepresentationType:IfcLabel
几何 Item as (IfcElement) (->几何表达 OR ->表面几何)->Items:IfcRepresentationItem

//正文(Body):
定义
具有三维几何的构件 继承 IfcElement
(具有三维几何的构件->几何 ID[Value]='Body') OR (具有三维几何的构件->几何 ID[存在]=FALSE)
(具有三维几何的构件->几何 Type
[Value]='SurfaceOrSolidModel'|'SurfaceModel'|'Tessellation'|'SweptSolid'|'AdvancedSweptSolid'|'Brep'|'Advanced-
Brep'|'CSG'|'Clipping'|'SolidModel'|'MappedRepresentation') OR (具有三维几何的构件->几何 Type[存在]=FALSE)
具有三维几何的构件->几何 Item[Type] >=
'IfcTessellatedItem'|'IfcShellBasedSurfaceModel'|'IfcFaceBasedSurfaceModel'|'IfcSolidModel'|'IfcTessellatedFaceSet'
'|'IfcExtrudedAreaSolid'|'IfcRevolvedAreaSolid'|'IfcSweptAreaSolid'|'IfcSweptDiskSolid'|'IfcFacetedBrep'|'
IfcAdvancedBrep'|'IfcCSGSolid'|'IfcBooleanClippingResult'|'IfcMappedItem'
{概念名称}[类型]>='具有三维几何的构件'

约束
//对象的几何类型应满足要求
{概念名称}(->几何 Type='{几何 Type 类型}'且 ->几何 Item[类型]>='{几何 Item 类型}')

```

注 1：其中“{}”表示用户在使用中需要补充的内容。

注 2：几何 ID 类型、几何 Type 类型以及几何 Item 类型，分别对应 GB/T 51447 中的“表达标识符表”(Representation identifiers)、“表达类型表”(Representation types)以及“几何表达 Item 类型”(IfcGeometricRepresentationItem)。

注 3：约束块中的几何 Type 类型以及几何 Item 类型选择参考表 B.4；当选项为多个值时，使用“|”将多个值分隔。

注 4：本规则主要针对三维几何构件，IFC4.0.2.1IfcRepresentation 的属性 RepresentationIdentifier 和 RepresentationType 选填。因此，本模板在筛选三维构件时，将前两个属性为空的情况也考虑进来；此外，也基于这个原因，当要求构件 RepresentationIdentifier 为三维时，将几何 Item 中表达三维的选项全部列出来表示。

示例：

```

约束
IfcColumn(->几何 Type='SweptSolid'且 ->几何 Item[类型]>='IfcExtrudedAreaSolid')

```

表 B.4 适用于 GB/T 51447 的三维几何 Type 类型和几何 Item 类型

几何 Type	中文描述	几何 Item
SurfaceOrSolidModel	体表面或实体几何体	IfcTessellatedItem IfcShellBasedSurfaceModel IfcFaceBasedSurfaceModel IfcSolidModel
SurfaceModel	体表面模型几何体	IfcTessellatedItem IfcShellBasedSurfaceModel IfcFaceBasedSurfaceModel
Tessellation	体镶嵌几何体	IfcTessellatedFaceSet
SweptSolid	体扫描几何体	IfcExtrudedAreaSolid IfcRevolvedAreaSolid
AdvancedSweptSolid	体高级扫描体	IfcSweptAreaSolid IfcSweptDiskSolid
Brep	体边界表达几何体	IfcFacetedBrep
AdvancedBrep	体高级边界表达几何体	IfcAdvancedBrep IfcFacetedBrep
CSG	体 CSG 几何体	IfcCSGSolid
Clipping	体裁剪几何体	IfcBooleanClippingResult
SolidModel	实体几何体	IfcExtrudedAreaSolid IfcRevolvedAreaSolid IfcSweptAreaSolid IfcSweptDiskSolid IfcFacetedBrep IfcAdvancedBrep IfcCSGSolid IfcBooleanClippingResult IfcMappedItem
MappedRepresentation	映射几何体	IfcMappedItem

B.3 模型视图抽取规则模板

B.3.1 根节点保留规则、排除规则

根节点保留规则与排除规则是约束规则，其中可以引用来自定义规则的概念。根节点保留规则与排除规则的模板同约束类规则的对象存在性约束模板，见 B.2.2。

示例：

```
约束
//保留规则
IfcProject[存在] = TRUE
IfcBuildingStorey[存在] = TRUE
外墙[存在] = TRUE

//排除规则
内墙[存在] = FALSE
机电构件[存在] = FALSE
```

B.3.2 路径保留规则、排除规则

路径保留规则与排除规则是约束规则，其中可以引用来自定义规则的概念，以及来自头部的缩写。路径保留规则与排除规则可采用各种约束类规则模板，见 B.2。

示例：

```
约束
//保留模型中的空间结构关联
地形->IsDecomposedBy->RelatedObjects[存在]=TRUE
建筑->IsDecomposedBy->RelatedObjects[存在]=TRUE

//保留子构件与父构件的关联
构件->Decomposes->RelatingObject[存在]=TRUE

//保留构件关联的属性
构件->任意属性集->单值属性[存在]=TRUE

//排除特定构件的特定属性
机电构件->任意属性集->单值属性("生产厂家")[存在]=FALSE

//排除特定值域的特定属性
机电构件->任意属性集->单值属性(->属性值=c"@my_company.net")[存在]=FALSE
```

B.4 标记规则模板

B.4.1 头部标记

头部标记位于整个规则集的开头，记录全局的信息。

示例：

```
# 语言版本 = "GBT 45393_3_2025"
# 标题 = "示例规则集"
# 信息模式 = "IFC4"
# 版本 = "1.0.1"
```

```
# 最后编辑时间 = "2023-02-22T17:17:54+08:00"

//头部的主要内容
实例属性集 as (IfcObject)->IsDefinedBy->RelatingPropertyDefinition;IfcPropertySet
```

B.4.2 表达式标记

表达式标记紧接于一个规则表达式的上方,作为这条规则表达式的标记信息。

示例:

```
定义
防火门 继承 IfcDoor

# "防火门的分类编码定义"
防火门->分类编码=c"10.10.30"

约束

# 属性名称 = "耐火极限"
# 等级 = "重要"
防火门->类型属性集->单值属性("耐火极限")->属性值!=""
```

B.4.3 分块标记

分块标记紧接于一个分块开始标记的下方,作为这个块的标记信息。

示例:

```
# 块

# 名称 = "构件定义规则块"
# 描述 = "根据分类编码标准制定的构件概念定义规则"
# 状态 = "草案"

定义
防火门 继承 IfcDoor
防火门->分类编码=c"10.10.30"

防火墙 继承 IfcWall
防火墙->分类编码=c"10.20.30"

# 块结束
```



参 考 文 献

- [1] GB/T 16656.1—2008 工业自动化系统与集成 产品数据表达与交换 第1部分:概述与基本原理
- [2] GB/T 16656.11 工业自动化系统与集成 产品数据表达与交换 第11部分:描述方法:EXPRESS 语言参考手册
- [3] GB/T 36456.1—2018 面向工程领域的共享信息模型 第1部分:领域信息模型框架
- [4] GB/T 36456.3—2018 面向工程领域的共享信息模型 第3部分:测试方法
- [5] ISO 29481-1 Building information models—Information delivery manual—Part 1: Methodology and format
- [6] buildingSMART. mvdXML Specification 1.1: P9-002 [S/OL]. [2016-02-15]. https://standards.buildingsmart.org/MVD/RELEASE/mvdXML/v1-1/mvdXML_V1-1-Final.pdf.
-

