

轨道电子地图基础数据建模与验证

黄旭

(中铁第一勘察设计院集团有限公司, 西安 710000)

摘要: 轨道电子地图数据是实现列车卫星定位的基础数据, 确保其正确性是实现列车精确定位的前提; 电子地图基础数据是生成电子地图文件的基础, 因此确保基础数据的正确性至关重要; 针对基础数据内容多、变更频繁的现状和基于文本语言描述规则存在二义性等问题, 提出一种电子地图基础数据建模与验证方法; 深度挖掘基础数据间的约束关系, 结合相关规范和领域专业知识, 提取基础数据验证规则, 利用 Prolog 对各类数据规则搭建其验证模型; 以实际线路数据为例, 证明该验证方法的高效性和准确性。

关键词: 轨道电子地图; 数据建模; 数据验证

Modeling and Verification of Basic Data of Track Electronic Map

HUANG Xu

(China Railway First Survey and Design Institute Group Co., Ltd., Xi'an 710000, China)

Abstract: Track electronic map data is the basic data for realizing train satellite positioning, and ensuring its correctness is the premise of realizing accurate train positioning. The electronic map basic data is the input basis for generating electronic map files, so it is very important to ensure the correctness of the basic data. Aiming at the problems of large basic data, frequent changes and ambiguity in text language description rules, a basic data modeling and verification method for the electronic maps is proposed. The constraint relationship of the basic data is analyzed deeply, combined with relevant specifications and domain expertise, the basic data validation rules is extracted, Prolog language is used to describe the data and validation rules, and build its validation model. the actual line data is taken as an example to prove the efficiency and accuracy of the verification method.

Keywords: track electronic map; data modeling; data verification

0 引言

随着全球导航卫星系统的发展, 卫星定位的技术在轨道交通领域的应用也日渐成熟。国外的 RENE^[1]、GAD-EROS^[2] 和 LOCOPROL^[3]; 国内的青藏铁路增强型列车控制系统 (ITCS) 以及正在试验的新型列控系统均是将卫星定位作为列车定位的重要手段。尤其随着我国自主研发的北斗卫星导航系统的不断发展, 将北斗卫星定位应用于铁路各个领域的研究也日益增长^[4-7]。

基于轨道线路的电子地图数据是实现列车卫星定位的前提, 不正确的电子地图数据将会导致列车定位失效或错误, 甚至会导致系统输出错误或停车等事故, 因此确保电子地图数据的正确性至关重要。电子地图数据的生成过程可概括为: 前期设计、现场测绘、数据生成、验证与确认。前期设计是生成电子地图数据的第一步, 将前期设计输出的内容暂且定义为电子地图基础数据 (简称“基础数据”)。基础数据是根据系统需求和相关规范进行设计, 可供后续测绘和制作电子地图数据文件使用, 基础数据是源头, 其正确与否直接关系到后续工作的正确性。为确保列车精确定位和系统安全运行, 需对电子地图基础数据进行正确性

验证。

当前电子地图基础数据的验证方式主要为人工校验, 存在以下问题: 一是易出错, 校核过程个人的能力水平直接决定着校核质量, 易带来错误判断或遗漏等问题; 二是效率低, 人工判断时间花费较长, 且因现场或试验结果, 基础数据的迭代频次较高, 这就导致人工核对效率无法满足实际需求。因此探索一种高效、准确的验证方法, 对确保电子地图的正确性有重要意义, 也有助于实现列车精确定位。

当前对轨道电子地图数据的研究主要集中在地图数据处理、约简和地图匹配算法等方面。文献 [8] 提出一套生成高精度轨道电子地图的数据处理方法, 分别采用卡尔曼滤波方法对采集数据进行滤波, 采用轨迹拟合方法对轨道曲线进行归一化处理, 采用垂直距离判据数据约简算法去除冗余数据, 最终可生成高精度电子地图数据。文献 [9] 提出用线段近似法描述曲线轨道, 可保证轨道精度又降低数据量。文献 [10] 提出基于方位角的曲率方法对线路的平面线性进行特征识别与分段拟合。文献 [11] 对电子地图数据进行分析, 开发了电子地图生成软件。文献 [12-14] 分别对电子地图的匹配算法进行了研究, 通过仿真可

收稿日期: 2022-10-25; 修回日期: 2022-11-30。

作者简介: 黄旭 (1994-), 女, 陕西西安人, 硕士, 工程师, 主要从事铁路信号的设计工作方向的研究。

引用格式: 黄旭. 轨道电子地图基础数据建模与验证[J]. 计算机测量与控制, 2023, 31(7): 284-289.

证明匹配算法的准确性。作为电子地图数据验证中重要的环节，电子地图基础数据的验证目前研究较少。

针对当前现状，本文首先对电子地图基础数据结构和内容进行分析；深入分析基础数据，结合相关规范和领域专家知识，提取出基础数据验证规则；针对基础数据内容多、变更频繁等特点，提出一种电子地图基础数据建模与验证方法，用 Prolog 语言描述数据和验证规则，搭建其验证模型。最后以实际线路数据为例，搭建其数据验证模型，证明该验证方法的高效性和准确性。这对提高电子地图数据的正确性有着积极作用。

1 轨道电子地图基础数据分析

当前不同列控系统对应的电子地图数据格式不尽相同，本文以其中一种列控系统对应的数据为例展开研究。电子地图数据以车站为最小数据组织单位，区间数据按照划分原则分别纳入相邻的车站数据组织。电子地图数据包含轨道地理信息文件和固定应用数据文件，轨道地理信息描述的是卫星数据与轨道位置数据间的对应关系，固定应用数据文件描述了区间和站内的关键设备及边界所在的轨道和坐标位置信息。地图数据均是依据基础数据加上测绘数据编制而成。

电子地图基础数据一般是由设计院输出的一套 Excel 表格，描述了每个车站管辖范围内测绘的轨道、关键点的名称、类型、所在轨道、里程等信息。基础数据包括车站信息表、轨道信息表、关键点信息表、应答器信息表和道岔信息表，如线路里程存在里程转换或断链则还需要补充里程转换信息表和断链信息表。车站信息表描述了线路中车站名称、对应的各级编号及车站所属的中心设备序号。轨道信息表描述了线路划分的轨道号、轨道的起始和结束边界以及边界对应的里程和相对轨道起点的位置。关键点包括 3 种类型：应答器、道岔和数据边界，关键点信息表分别描述了各种关键点的类型、名称、里程、所在轨道、距轨道起始边界的位置等信息。应答器信息表和道岔信息表则根据关键点信息表进行填写，应答器信息表中补充了应答器编号、用途等内容，道岔信息表中补充了开岔开向、定位开向等信息。基础数据中的位置数据均是基于线路运营里程计算所得，可供后续的测绘数据做一致性校验。对应的数据组织关系如图 1 所示。

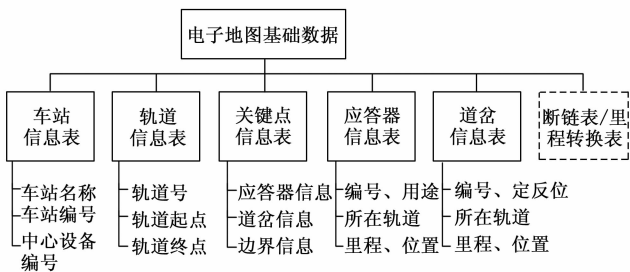


图 1 轨道电子地图基础数据结构组织

2 验证规则提取

电子地图基础数据验证规则来源可分为两类：一是铁路总颁布的各类行业规范文件；二是通过对基础数据的深入分析和挖掘结合领域专业知识提取所得。基础数据本身数据格式定义明确、数据间的关联性较强，因此根据其内容特点，可将规则概括为两类：值域规则和逻辑规则。

2.1 值域规则

基础数据包含几类数据对象，每一类数据对象所描述的信息均需满足其对应的值域规则，从数据类型、格式、精度和取值范围四方面对基础数据的值域规则进行描述。

2.1.1 数据类型

不同的数据对象描述的内容不同，每一项内容属性对应的数据类型不同。基础数据主要包含的数据类型有：整型、字符串、布尔类型、枚举类型。

2.1.2 数据格式

根据相关规范，基础数据格式有明确要求，各项数据均有其对应的命名或表示规则。如里程的格式应为 (KXXX+XXX)；应答器名称应以 B 开头等。

2.1.3 数据精度

数据值应与其对应的单位精度匹配，如里程值精确到米；位置值精确到厘米。

2.1.4 取值范围

每一项数据都有对应的实际意义，根据相关规范和专业常识，可总结得到某些数据对应的取值范围，如距轨道起点的位置值一定大于等于 0；轨道号取值应为正数；根据区间长度划分原则可确定距轨道起点位置值的上限。

表 1 为提取得到的基础数据值域规则具体内容的部分示例。

表 1 值域规则举例

属性名称	值域规则
序号	整型；自然连续且唯一
车站名称	字符串；在车站信息表内唯一
大区、分区、车站编号、TSRS 序号、RBC 序号	整型
轨道号	整型；车站内唯一
轨道起点属性	枚举型{起始数据边界,道岔}
轨道终点属性	枚举型{结束数据边界,道岔}
里程	整型；单位精确到米；格式为 KXXX+XXX
位置	非负数；单位精确到厘米
关键点类型	枚举型{数据边界,应答器,道岔}
应答器名称	根据规范命名,格式为 B 加数字；名称唯一
应答器编号	根据规范命名；编号唯一
应答器设备类型	枚举型{实体,虚拟}
应答器用途	枚举型{Q,XQ,JZ,CZ,FJZ...}
道岔编号	整型；车站内唯一
开岔方向	布尔型
.....

2.2 逻辑规则

值域规则是对数据本身的值域、格式等方面的约束，逻辑规则是描述数据之间的约束关系。以下定义并列举了部分电子地图基础数据的逻辑验证规则。

定义 1: 车站信息定义为 $\langle StationId, StationName, RegionNum, PartitionId, StationNum, TSRS_Num, RBC_Num \rangle$ ，分别表示序号、车站名称、大区编号、分区编号、车站编号、TSRS 序号、RBC 序号。

规则 1: 不可能存在编号相同的两个车站。

$$(Region_{num,i}, Partition_{num,i}, Station_{num,i}) \neq (Region_{num,j}, Partition_{num,j}, Station_{num,j})$$

定义 2: 若存在线路断链，断链数据定义为 $\langle D_{id}, D_{type}, D_{begin}, D_{end}, D_{length}, D_{dir} \rangle$ ，分别表示序号、断链类型、断链起点里程、终点里程、长度和线别。

定义 3: 轨道信息定义为 $\langle T_{id}, T_{station}, T_{num}, T_{begin_attr}, T_{begin_loa}, T_{begin_len}, T_{end_attr}, T_{end_loa}, T_{end_len} \rangle$ ，分别表示序号、轨道所属车站、轨道号、轨道起/终点属性、里程、位置。

规则 2: 断链长度等于断链起点和终点里程之差。

$$D_{length} = |D_{end} - D_{begin}|$$

规则 3: 轨道终点位置值与轨道终点里程、轨道起点里程以及是否包含断链信息有关。若轨道起终点之间不存在断链，则轨道终点位置等于终点里程与起点里程之差乘 100；若存在断链则需考虑断链长度。 D_i 为断链：

$$\text{当 } \forall D_i \in D, D_i \notin T, T_{end_len} = 100 * |T_{end_loa} - T_{begin_loa}|;$$
$$\text{当 } \forall D_i \in D, D_i \notin T, T_{end_len} = 100 * (|T_{end_loa} - T_{begin_loa}| \pm D_{length})。$$

式中， D 为断链合集， D_i 为断链合集中的一个元素； T 表示轨道合集。

规则 4: 同一车站内轨道号唯一。

$$(T_{station,i}, T_{num,i}) = \neq (T_{station,j}, T_{num,j})$$

规则 5: 一般情况一条轨道长度存在上限。

$$\max\{T_{end_len,1}, T_{end_len,2}, \dots, T_{end_len,n}\} < T_{end_len,max}$$

定义 4: 应答器信息定义为 $\langle B_{id}, BG_{name}, BG_{num}, G_{num}, B_{loa}, B_{type}, B_{use}, B_{station}, B_{track}, B_{len} \rangle$ ，分别表示序号、应答器组名称、应答器组编号、组内位置、里程、类型、用途、所属车站、所属轨道号、距所在轨道起点的位置。

规则 6: 当且仅当应答器类型是“虚拟”，对应的用途为“XQ”。

$$B_{type} = \text{“虚拟”} \iff B_{USE} = \text{“XQ”}$$

规则 7: 应答器组内间距为 5 m。

当 $BG_{name,i} = BG_{name,j}$ 且 $G_{num,j} = G_{num,i} + 1$ 时， $B_{loa,j} = B_{loa,i} \pm 5$ 。

规则 8: 应答器距轨道起点的位置与应答器里程、应答器所在轨道起点里程以及是否包含断链信息有关。

$$\text{当 } \forall D_i \in D, D_i \notin B, B_{len} = 100 * |B_{loa} - T_{begin_loa}|;$$

当 $\forall D_i \in D, D_i \notin B, B_{len} = 100 * (|B_{loa} - T_{begin_loa}| \pm D_{length})$ 。式中， D 为断链合集， D_i 为断链合集中的一个元素； B 表示应答器与轨道起点之间数据合集。

规则 9: 应答器所属车站名称一定是车站信息表中的车站。

$$B_{station} \in StationName$$

定义 5: 道岔信息定义为 $\langle S_{id}, S_{station}, S_{num}, S_{loa}, S_{dir}, ST_{track}, ST_{len}, SD_{track}, SD_{len}, SF_{track}, SF_{len} \rangle$ ，分别表示序号、道岔所属车站、道岔编号、里程、开向、岔尖所在轨道号、位置、定位轨道号、位置、反位轨道号、位置。

规则 10: 一个车站内，同一条轨道上不存在岔尖里程相同的两组道岔。

当 $(S_{station,i} = S_{station,j}) \cap (ST_{track,i} = ST_{track,j})$ 时， $S_{loa,i} \neq S_{loa,j}$

规则 11: 岔尖所属轨道要不与定位所属轨道一致，要不与反位所属轨道一致。

$$(ST_{track} = SD_{track} \cap ST_{track} \neq SD_{track} \cap ST_{track} = SF_{track})$$

定义 6: 关键点信息定义为 $\langle P_{id}, P_{station}, P_{type}, P_{name}, P_{type2}, P_{track}, P_{len}, P_{loa} \rangle$ ，分别表示序号、车站名称、关键点类型、名称、类型、轨道号、位置、里程。

规则 12: 关键点信息表中只描述组内第一个应答器信息。

当 $(G_{num} = 1 \cup G_{num} = null) \cap (P_{type} = \text{“应答器”})$ 时， $P = B$ 。

规则 13: 关键点信息表中应包含道岔信息表中所有道岔。

当 $(P_{type} = \text{“道岔”})$ 时， $P = S$ 。

规则 14: 关键点信息表中应包含所有轨道边界。

$(T_{begin_tr} = \text{“起始数据边界”}) \cup T_{begin_tr} = \text{“结束数据边界”}) \cap (P_{type} = \text{“数据边界”})$ 时， $P = T$ 。

其中： P 表示关键点合集， B 表示应答器合集， S 表示道岔合集， T 表示轨道合集。

3 建模与验证

为实现电子地图基础数据的自动验证，本文采用 Prolog 语言搭建基础数据的验证模型，目的一是对验证规则进行形式化描述，二是基于验证工具搜索出不满足规则的数据反例，最终可实现基础数据的自动验证。

3.1 Prolog 语言

Prolog 语言^[15]是以一阶谓词逻辑为基础的逻辑编程语言，采用陈述句式使用演绎推理进行问题求解，它利用已知事实和规则结合内部搜索、匹配和回溯的推理机智自动求解程序设定的目标，不需要在程序中列出详细的求解步骤。Prolog 语言仅具有事实、规则和目标 3 种基本类型^[16]。Prolog 语言在构建专家系统^[17-18]、智能检索系统^[19]、数据验证^[20-21]等方面已有不少研究。

事实是用来描述已知的对象和他们之间的关系，事实

由谓词及对象组成, 对象可以是一个或多个。谓词和对象可由用户自己定义。例如 like (bill, dog), 该事实描述的是 bill 喜欢 cindy。

规则用来描述事实之间的依赖关系, 由左边表示结论的谓词和右边表示条件的谓词组成, 中间通过“: -”连接, 以“.”结束。例如 like (cindy, X): -like (bill, X) 表示 cindy 喜欢 bill 喜欢的东西。

把事实和规则写进 Prolog 语言描述后, 可以其询问有关问题的答案, 问题就是程序运行的目标。目标可以是一个简单的谓词, 也可以是多个谓词组合。例如? -like (cindy, dog) 表示向程序询问“cindy 是否喜欢狗?”。

本文用 Prolog 语言的事实描述电子基础数据, 规则描述基础数据应满足的验证规则, 目标则是遍历数据是否满足所有规则, 并找出反例, 即不符合验证规则的数据。

3.2 数据事实

电子地图基础数据由车站信息表、轨道信息表、应答器位置表、道岔信息表、关键点信息表组成, 需要将 excel 表格中的每一行数据转换为 Prolog 可识别的事实。各数据表的事实表示如下:

车站信息: $station (A_{id}, A_{name}, Region_{num}, Partition_{num}, Station_{num}, TSRS_Num, RBC_Num)$ 。

轨道信息: $track (T_{id}, T_{station}, T_{num}, T_{begin_attr}, T_{begin_loa}, T_{begin_len}, T_{end_attr}, T_{end_loa}, T_{end_len})$ 。

应答器信息: $balise (B_{id}, BG_{name}, BG_{num}, G_{num}, B_{loa}, B_{type}, B_{use}, B_{station}, B_{track}, B_{len})$ 。

道岔信息: $switch (S_{id}, S_{station}, S_{num}, S_{loa}, S_{dir}, ST_{track}, ST_{len}, SD_{track}, SD_{len}, SF_{track}, SF_{len})$ 。

关键点信息: $poi (P_{id}, P_{station}, P_{type}, P_{name}, P_{type2}, P_{track}, P_{len}, P_{loa})$ 。

断链信息: $chain (D_{id}, d_{type}, D_{begin}, D_{end}, D_{length}, D_{dir})$ 。

数据事实由对应谓词和参数组成, 例如: track 是轨道数据事实的谓词, 参数 T_{id} 序号、 $T_{station}$ 表示轨道所属车站、 T_{num} 表示轨道号、 T_{begin_attr} 表示轨道起点属性、 T_{begin_loa} 表示起点里程、 T_{begin_len} 表示起点位置、 T_{end_attr} 表示轨道终点属性、 T_{end_loa} 表示终点里程、 T_{end_len} 表示终点位置。由此可将 excel 表中每一行数据转换为事实。

此外, 数据值域规则中枚举属性对应的数据事实可用列表来表示。

轨道起点属性 T_{begin_attr} 对应的数据事实为: $track_begin_attr (['起始数据边界', '道岔'])$ 。

轨道终点属性 T_{end_attr} 对应的数据事实为 $track_end_attr (['结束数据边界', '道岔'])$ 。

道岔开岔方向 S_{dir} 对应数据事实为: $switch_dir ([0, 1])$ 。

关键点类型 P_{type} 对应数据事实为: $poi_type (['应答器', '数据边界', '道岔'])$ 。

3.3 规则模型

将提取的基础数据验证规则转换为 Prolog 语言对应的规则表述, 构建数据验证的规则模型。

3.3.1 值域规则模型

值域规则对应的是数据类型和取值的约束。简单的数据类型约束可利用 Prolog 的内部谓词作为规则谓词, 如 integer (A), float (A), number (A) 等。其他特殊的格式约束需进行自定义规则。

轨道信息事实中轨道起点属性应满足固定几种类型, 验证轨道属性的规则可描述为:

$check_track_beginAttr (T_{id}): - track (T_{id}, _, _, _, _, T_{begin_attr}, _, _, _, _), track_begin_attr (Track_begin_attr), /+ member (T_begin_attr, Track_begin_attr)$ 。

其中: T_{id} 是轨道数据的序号, member 是 Prolog 的内部谓词, member (T_begin_attr, Track_begin_attr) 描述变量 T_begin_attr (轨道起点属性值) 是否是集合 Track_begin_attr (轨道起点属性值集合) 的一个元素, “/+” 是 Prolog 的内部符号, 表示取反值。执行 $check_track_attr (T_{id})$ 规则, 可搜索出轨道起点属性不满足要求的轨道序号。

应答器名称的第一个字符应是字母“B”。对应规则可描述为:

$check_blase_name (B_{id}): - balise (B_{id}, BG_{name}, _, _, _, _, _, _, _, _), name (BG_{name}, X), X = [X_1 | _], X_1 / = 66$ 。

其中: B_{id} 是应答器数据的序号, BG_{name} 是应答器组名称, name 是 Prolog 的内部谓词, name (BG_name, X) 是将 BG_{name} 转换为对应 ASCII 码列表, X_1 为 BG_{name} ASCII 码列表中的第一个元素。执行 $check_blase_name (B_{id})$ 规则可输出名称中第一个字符不是“B”的应答器数据对应的序号, 可供错误定位及修改。

3.3.2 逻辑规则模型

规则 1: 不可能编号相同的两个车站。

$check_stationNum (A_{id1}, A_{id2}): - station (A_{id1}, _, Region_{num1}, Partition_{num1}, Station_{num1}, _, _), station (A_{id2}, _, Region_{num2}, Partition_{num2}, Station_{num2}, _, _), A_{id1} < A_{id2}, Region_{num1} = Region_{num2}, Partition_{num1} =, Partition_{num2}, Station_{num1} =, Station_{num2}$ 。

执行 $check_stationNum (A_{id1}, A_{id2})$ 规则可以输出大区编号、分区编号和车站编号分别相同的车站数据序号。

规则 2: 断链的长度等于终点里程于起点里程之差, 对应规则可描述为:

$check_chainLen (D_{id}): - chain (D_{id}, _, D_{begin}, D_{end}, D_{length}, _), D_{length} / = abs (D_{end} - D_{begin})$ 。

其中: D_{id} 是断链序号, abs 是 Prolog 的内部谓词, 表示求绝对值。执行 $check_chainLen (D_{id})$ 规则可输出不满

足长度等于起终点里程差的断链序号。

规则 3: 轨道长度与起终点里程和是否存在断链有关, 该条规则需多条规则组合描述。

$check_ChainIn(T_{id}, D_{type}, D_{length}): -track(T_{id}, _, T_{num}, T_{begin_loa}, _, _, T_{end_loa}, _), chain(D_{id}, D_{type}, D_{begin}, D_{end}, D_{length}, D_{dir}), (T_{begin_loa} < D_{begin}, T_{end_loa} > D_{end}, is_odd(T_{num}), D_{dir} = '下行');$ ($T_{begin_loa} > D_{begin}, T_{end_loa} < D_{end}, is_even(T_{num}), D_{dir} = '上行'$)。

$check_T_endLen(T_{id}): -track(T_{id}, _, _, _, T_{begin_loa}, _, _, T_{end_loa}, T_{end_len}), check_Chain_in(T_{id}, D_{type}, D_{length}), (D_{type} = '长链', T_{end_len}/=100 * (|T_{begin_loa} - T_{end_loa}| + D_{length})); (D_{type} = '短链', T_{end_len}/=100 * (|T_{begin_loa} - T_{end_loa}| - D_{length}))。$

$check_T_endLen(T_{id}): -track(T_{id}, _, _, _, T_{begin_loa}, _, _, T_{end_loa}, T_{end_len}), /+ check_Chain_in(T_{id}, D_{type}, D_{length}), T_{end_len}/=100 * (|T_{begin_loa} - T_{end_loa}|)。$

谓词 $check_ChainIn$ 是判断序号为的轨道数据的起终点范围内是否包含断链, 并取得断链的类型和长度。其中“,”代表逻辑关系中的“且”, “;”代表逻辑关系中的“或”。谓词 $check_T_endLen$ 由两条规则组成, 用多条规则可以定义同一个谓词, 分为轨道数据包含断链和不包含断链两种情况, 执行 $check_T_endLen(T_{id})$ 可输出轨道终点位置与里程不符合逻辑规则的轨道数据序号。

规则 4: 同一车站轨道号唯一。

$is_uniq(X, Y_list): -member(X, Y_list), delete(Y_list, X, Last_l), length(Last_l, Len_last), length(Y_list, Len_Y), Len_Y is Len_last+1。$

首先自定义 is_uniq 规则, 其中 $member$, $delete$ 和 $length$ 均是内部谓词, 执行 $is_uniq(X, Y_list)$ 可判断变量 X 是否是列表 Y_list 中的非重复项, 如果是输出 FALSE, 如果是非重复项则输出 TRUE。

$Check_T_num(T_{id}): -track(T_{id}, T_{station}, _, _, _, _, _, _, _), findall(T_{num}, track(_, T_{station}, T_{num}, _, _, _, _, _, _), T_numList), findall(T_{id}, track(T_{id}, _, T_{station}, _, _, _, _, _, _), T_IdList), findall(T_{id}, (is_uniq(T_num, T_numList), track(T_{id}, _, T_{station}, T_num, _, _, _, _, _)), Uniq_id), member(T_{id}, T_IdList), /+member(T_{id}, Uniq_id)。$

利用 Prolog 内部谓词 $findall$ 首先找出轨道信息表中同一个车站所有轨道号和序号分别至变量 T_num_list , Id_list 中, 在找出轨道号 T_num 是 T_num_list 中的非重复项的轨道数据对应的序号至变量 $Uniq_id$ 中, 再判断如果序号是 Id_list 中的元素, 但不是 $Uniq_id$ 中的元素, 则说明该序号对应的轨道号是重复项。执行 $Check_T_num(T_{id})$ 可输出同一车站轨道号重复的数据对应的序号。

规则 5: 一般情况一条轨道长度存在上限

$check_T_max(T_{id}): -track(T_{id}, _, _, _, _, _, _, _, T_{end_len}), /+(T_{end_len} < 100000000)。$

执行 $check_T_max(T_{id})$ 可输出轨道长度不小于 100 km 的轨道数据序号。

规则 6: 当且仅当应答器类型是“虚拟”, 对应用途为“XQ”。

$check_baliseUse(B_{id}): -balise(B_{id}, _, _, _, _, B_{type}, B_{use}, _, _, _), /+(B_{type} = '虚拟', -> B_{use} = 'XQ', B_{use} = 'XQ' -> B_{type} = '虚拟')。$

3.3.3 错误输出模型

值域规则模型和逻辑规则模型可以验证数据事实是否满足对应规则, 并输出不满足规则的数据序号。但每一条规则需在 Prolog 编译器中逐条去询问后才会输出验证结果, 电子地图基础数据的值域规则和逻辑规则多达几十至上百条, 每次验证逐条询问规则效率太低。因此本文设计了错误输出模型, 可将自动输出违反规则的数据信息至一个文档, 可供错误修改。

第一步: 将每条规则对应的错误数据信息添加至 $error$ 事实;

在每一条规则后添加 “ $* -> assertz(error(message)).$ ”, 可将错误数据信息添加至 $error$ 事实中。” “ $* ->$ ” 是 Prolog 的内部操作符, “ $A -> B$ ” 等价于 “if A then B”, $assertz$ 是 Prolog 内部谓词, 是将插入当前动态数据库中的同名谓词的事实之后。例如:

$check_track_beginAttr(T_{id}): -track(T_{id}, _, _, _, T_{begin_attr}, _, _, _, _), track_begin_attr(Track_begin_attr), /+member(T_begin_attr, Track_begin_attr), * -> assertz(error(T_{id}, T_begin_attr, 'not in', Track_begin_attr))。$

执行 $check_track_beginAttr$ 规则后, $error(X)$ 事实中会增加轨道起点属性不满足值域规则的数据信息。

第二步: 设置一个总询问规则 $all_rule(_): -rule1(T_{id}); rule2(B_{id}); \dots rule_n(B_{id})。$ 每次验证数据只需在编译器询问一次 $all_rule(_)$, 即可将所有规则对应的错误信息添加至 $error$ 事实中。

第三步: 将所有 $error$ 事实输出至文档中。

$save(FileName): -telling(Old), tell(FileName), listing(error), told, tell(Old)。$

其中 $save$, $telling$, $tell$, $listing$, $told$, $tell$ 均是 Prolog 的内部谓词, $save$ 谓词可将当前内存中的事实存入文件 “Filename” 中, $listing(error)$ 列出谓词 $error$ 中的所有子句。

3.4 验证结果

利用规则模型对实际线路电子地图基础数据进行验证, 本文采用 GUN prolog 解释器作为验证工具, 以某单线铁路为例, 线路全长 800 多公里, 共设有 21 个车站, 电子地图

基础数据共 2 499 条, 提取的值域规则 30 条, 逻辑规则 54 条。表 2 是数据验证结果。

表 2 规则模型验证结果

事实 / 条	规则 / 条	验证时间 / ms	错误 / 处	检测 / 处	准确率
2 499	84	2 586	100	94	94%

验证结果可知, 验证 2 499 条数据, 84 条规则总耗时约 2.6 s, 相较于人工验证数据, 本文建立的验证模型及方法具有高效性。

错误验证率是模型监测出的错误数量占数据真实的错误数量的百分比, 其中真实的错误人工添加所得, 是由非模型设计人员对已确认正确的数据添加错误。2 599 条数据的错误验证率达 94%, 可见该验证模型和方法具有较高的准确性。

分析未被模型识别的错误数据, 发现主要原因有两方面: (1) 数据逻辑关系独立, 例如车站信息表中 TSRS 设备序号与其他数据之间无规则约束, 因此当序号由“4”被错误修改为“3”, 规则模型无法检测错误; (2) 多个错误叠加, 构成数据巧合。例如同时修改轨道终点里程和位置值, 使得修改后位置和里程之间的关系仍满足规则。

4 结束语

针对电子地图基础数据校核当前存在的问题, 提出基于 Prolog 的数据验证方法, 深入分析基础数据格式、内容, 通过相关规范以及领域专业知识, 提取出基础数据的验证规则, 采用 Prolog 语言搭建数据事实和验证模型, 以实际线路为例进行试验, 证明了模型的可行性和有效性。

参考文献:

- [1] ALBANESE A, MARRADI L, CAMPA L, et al. The Rune project: Navigation performance of GNSS-based railway user navigation equipment [C] // Proc Navitec, Noordwijk, The Netherlands, 2004.
- [2] URECH A, DIESTRO J P, GONZÁLEZ O, et al. Galileo demonstrator for railway operation system [C] // Proc Data Syst Aerosp, Dublin, 2002: 46.
- [3] PIERRE, MERTENS, JEA MERTENS P, FRANCKART J P. Low-cost signalling on low-density lines [J]. Railway Gazette Int, 2003, 159 (8): 498.
- [4] 蔡 焯, 陶汉卿, 侯宇婷, 等. 北斗卫星导航系统在列车定位中的应用研究与发展 [J]. 铁道科学与工程学报, 2022, 19 (8): 2417-2427.
- [5] 杨 杰, 王一军, 余明杨. 北斗定位系统在轨道参数测量上的应用 [J]. 计算机测量与控制, 2018, 26 (1): 14-17.
- [6] 上官伟, 袁重阳, 蔡伯根, 等. 北斗二代在西部低密度铁路中的应用 [J]. 交通运输工程学报, 2016, 16 (5): 132-141.
- [7] 开祥宝, 张 森, 高 媛, 等. 基于北斗导航的列车运行控制系统架构设计 [J]. 铁道运输与经济, 2018, 40 (10): 66-70.
- [8] 左自辉, 王开锋, 许 聪, 等. 生成高精度轨道电子地图的数据处理方法 [J]. 中国铁道科学, 2016, 37 (4): 134-138.
- [9] 陈志强, 鲍鹏宇. 电子地图轨道地理坐标描述方法研究 [J]. 铁路通信信号工程技术, 2021, 18 (2): 22-26.
- [10] 陶维杰, 蔡伯根, 王 剑, 等. 数字轨道地图平面线形特征提取方法研究 [J]. 铁道学报, 2019, 41 (9): 81-87.
- [11] 王子琪. 面向列控应用的电子轨道地图生成方法研究 [D]. 北京: 北京交通大学, 2012.
- [12] 吴波前, 蔡伯根, 陆德彪, 等. 基于卫星导航的列车轨道占用加权识别方法研究 [J]. 交通运输系统工程与信息, 2019, 19 (3): 62-67.
- [13] 王 霞. 基于北斗卫星的列车定位地图匹配算法研究 [D]. 大连: 大连交通大学, 2019.
- [14] 徐嘉驰. 基于 GIS 的列控数字轨道地图构建与匹配算法研究 [D]. 成都: 西南交通大学, 2020.
- [15] DIAZ D, CODOGNET P. The GNU prolog system and its implementation [C] // Proceedings of the 2000 ACM symposium on Applied computing-Volume 2, 2000: 728-732.
- [16] 王湘云. 基于谓词逻辑的知识表示和知识推理及在 Prolog 中的实现 [D]. 天津: 南开大学, 2008.
- [17] 张畅东. 基于 Prolog 的丝绸文物辅助设计专家系统核心技术开发 [D]. 郑州: 浙江大学, 2016.
- [18] 陈 瑜, 林 涛, 陈永雷. Visual Prolog 语言在测试设备故障诊断专家系统中的应用 [J]. 计算机测量与控制, 2007 (9): 1244-1246.
- [19] 杨 明. 基于 prolog 语言的军用直升机故障智能检索系统设计与实现 [D]. 哈尔滨: 哈尔滨工业大学, 2015.
- [20] 陈鹤文, 周 勇, 燕雪峰. 一种基于 Prolog 的时间约束业务流程验证方法 [J]. 计算机科学, 2016, 43 (6): 173-178.
- [21] 谭冠华, 徐田华, 王海峰, 等. 基于 Prolog 的列控工程数据验证方法 [J]. 铁道学报, 2019, 41 (6): 91-99.
- [22] 王强宇. 基于深度神经网络的动态手势识别技术研究 [D]. 北京: 中国矿业大学, 2019.
- [23] 刘德强. 基于机器视觉的手势识别系统设计与实现 [J]. 时代农机, 2015, 42 (11): 49-51.
- [24] 高 翔. 基于手势识别的虚实交互技术研究 [J]. 电子制作, 2022, 30 (9): 88-90.
- [25] 丛玉华, 何 啸, 邢长达, 等. 基于计算机视觉手势识别的人机交互技术研究 [J]. 兵器装备工程学报, 2022, 43 (1): 152-160.
- [26] 凌 利, 陶 俊, 吴 瑰. 基于 YOLOv3 的手势识别技术 [J]. 江汉大学学报 (自然科学版), 2021, 49 (5): 79-87.
- [27] 黄昌正, 陈 曦, 周言明, 等. 视觉手势识别技术在新冠肺炎疫情防控的应用研究 [J]. 中国科技信息, 2021 (20): 87-88.
- [28] 何 坚, 刘 炎, 祖天奇. 时空特征与通道注意力融合的视觉手势识别技术 [J]. 北京工业大学学报, 2021, 47 (8): 824-832.