IntervalTree+结构的函数式建模、机械化验证及其应用

左正康[1,2,3],张晗庆[1],王昌晶[1,2,3],游 珍[2,3*]
(1. 江西师范大学计算机信息工程学院,江西南昌 330022;2. 江西师范大学网络化支撑软件国家科技合作基地,江西南昌 330022;
3. 江西师范大学高性能计算江西省重点实验室,江西南昌 330022)

摘 要: 区间树(IntervalTree)是一种对动态集合进行维护的搜索树,可用于高效地存储和搜索区间集合. 当前实现了IntervalTree在Isabelle/HOL的建模与验证,其区间信息是在二叉搜索树上进行扩充的,IntervalTree结构支持的基本操作时间复杂度较高. 为此,本文对IntervalTree结构的节点附加额外颜色信息且保证树的平衡,提出了IntervalTree+结构,相较于IntervalTree结构的实现,插入和删除等操作最坏时间复杂度$O(n)$改进到$O(\log n)$. 然后使用Isabelle定理证明器对IntervalTree+结构及其操作函数进行了函数式建模,对其不变量进行了机械化验证,保证了IntervalTree+结构操作函数的正确性和可靠性. 进一步,首次提出一种区域匹配算法的通用验证规约,旨在解决一系列的区域匹配算法正确性验证问题. 提出的IntervalTree+结构通过严格的机械化验证,且操作最坏时间复杂度相较于IntervalTree结构由$O(n)$优化到$O(\log n)$,可应用于区域匹配、视觉日志和评估模型等相关算法优化.

关键词: 区间树;IntervalTree+;函数式建模;机械化验证;区域匹配算法;Isabelle定理证明器
基金项目: 国家自然科学基金(No.62462036, No.62462037);江西省自然科学基金(No.20242BAB26017, No.20232BAB202010);江西省主要学科学术与技术带头人培养项目(No.20232BCJ22013)
中图分类号: TP311 **文献标识码:** A **文章编号:** 0372-2112(2025)02-0474-09
电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20240427

Functional Modeling, Mechanized Verification and Application of IntervalTree+

ZUO Zheng-kang[1,2,3], ZHANG Han-qing[1], WANG Chang-jing[1,2,3], YOU Zhen[2,3*]
(1. *School of ComputerInformation Enjineering*, *Jiangxi Normal University*, *Nanchang*, *Jiangxi* 330022, *China*;
2. *State International S&T Cooperation Base of Networked Supporting Software*, *Jiangxi Normal University*, *Nanchang*, *Jiangxi* 330022, *China*; 3. *Jiangxi Provincial Key Laboratory for High Performance Computing*, *Jiangxi Normal University*, *Nanchang*, *Jiangxi* 330022, *China*)

Abstract: IntervalTree is a search tree used for maintaining dynamic sets, specifically designed for efficient storage and retrieval of interval collections. The current implementation of IntervalTree involves modeling and verification in Isabelle/HOL, where interval information is expanded upon a binary search tree. However, the time complexity of the basic operations supported by the IntervalTree structure is relatively high. To address this issue, this paper proposes the IntervalTree+ structure, augmenting nodes of the IntervalTree with additional color information to ensure tree balance. As compared to the original IntervalTree structure, the worst-case time complexity for operations such as insertion and deletion is improved from $O(n)$ to $O(\log n)$ in the IntervalTree+ implementation. Subsequently, functional modeling of the IntervalTree+ structure and its operations is performed using the Isabelle theorem prover. Mechanical verification of invariants is conducted to ensure the correctness and reliability of IntervalTree+ structure operations. Additionally, for the first time, a generic verification specification for region matching algorithms is proposed to address correctness verification issues across a series of such algorithms. The proposed IntervalTree+ structure has been rigorously verified through formal mechanization. Compared to IntervalTree structure, its worst-case time complexity is optimized from $O(n)$ to $O(\log n)$. This optimization makes it applicable to algorithmic enhancements in areas such as region matching, visual logging, and model evaluation.

收稿日期:2024-05-09;修回日期:2025-01-13;责任编辑:李勇锋
*通讯作者:游珍

Key words: IntervalTree; IntervalTree+; functional modeling; mechanized verification; region matching algorithm; Isabelle theorem prover

Foundation Item(s): National Natural Science Foundation of China (No.62462036, No.62462037); Natural Science Foundation of Jiangxi Province (No.20242BAB26017, No.20232BAB202010); Academic and Technical Leaders Training Program of Jiangxi Province (No.20232BCJ22013)

1 引言

IntervalTree是一种对动态集合进行维护的搜索树,可用于高效地存储和搜索区间集合[1]. 它在区间重叠查询[2]、空间和时间数据库[3~7]、隐私保护视觉日志服务[8]、新文档评估模型[9]等方面有着广泛的应用. 对区间树满足性质的验证主要分为手工验证和机械化验证. 文献[3~7]在算法层面设计了区间树的结构及其相关操作,上述文献主要从理论角度对区间树的性质进行了手工验证. 然而,手工验证涉及大量的分析、归纳和证明过程,这些过程复杂、容易出错,同时也会增加验证的时间成本. 文献[2]在Isabelle/HOL定理证明器[10]中对IntervalTree结构及其基本操作进行了函数式建模和正确性验证,其区间树是基于二叉搜索树,插入、删除等基本操作的时间复杂度较高. 本文基于文献[2]形式化定义的IntervalTree结构,对其节点附加额外颜色信息且保证树的平衡,扩展成为一种新的IntervalTree+结构;通过添加颜色平衡调整函数对IntervalTree+结构的插入、删除和搜索等操作进行了函数式建模,有效降低了树的高度,提高了区间搜索效率;然后,对IntervalTree+结构的元素不变量和结构不变量进行了机械化验证,保证了其操作的正确性和可靠性;最后,基于Isabelle/HOL中的locale区域首次提出满足其逻辑规约的区域匹配算法通用验证规约,该规约基于IntervalTree+结构解释(interpretation命令)[11]可实例化为数据分发管理系统的函数式区域匹配算法(Functional Region Matching Algorithm,FRMA),并通过严格的机械化证明有效保证了FRMA的正确性,基于所提的验证规约,有望解决一系列的区域匹配算法正确性验证问题.

本文工作的创新点总结如下:

(1)基于Nipkow等人[2]提出的IntervalTree结构,在对其节点附加额外的颜色信息且保证树的平衡基础上,提出了IntervalTree+结构,并对其操作进行了函数式建模和机械化验证. 其插入、删除和搜索操作的效率相较于IntervalTree结构有一定的改进,最坏时间复杂度由$O(n)$改进到$O(\log n)$.

(2)基于所提的IntervalTree+结构及其基本操作,实现了FRMA的建模,在保证精确区域匹配的前提下,提高了区域匹配的时间效率.

(3)首次提出了区域匹配算法通用验证规约,实例化可保证FRMA的正确性. 并且该验证规约可根据不同应用场景通过locale区域解释实例化到各类区域匹配算法中,有利于解决一系列的区域匹配算法正确性验证问题.

本文所提的IntervalTree+结构及基本操作、区域匹配算法的通用验证规约和FRMA的建模及验证的脚本可以参阅:https://github.com/ZZIsabelle/Interval-Tree-Plus.

2 相关研究

本节主要介绍区间树建模及其验证的国内外相关工作.

(1)区间树建模方面. 文献[1,12]在算法层面使用区间信息对二叉搜索树进行扩充,设计了自平衡区间树结构及其插入、删除和搜索操作,从理论角度对区间树的性质进行了手工验证. 文献[13]实现了m叉区间树,并将所有集合的计数统计值映射为一棵满m叉区间树,对该树的各个节点值添加噪声,提出了一种基于最优线性无偏的差分隐私直方图发布方法. 文献[14]提出了基于任意结构的区间树构造方法,该方法将直方图转换为伪完全k叉区间树,提高了查询和数据发布精度. 文献[15]实现了最大边缘区间树,并通过最小化基于边缘的判别目标函数来学习树,提供了一个在对数线性时间内计算最优解的动态规划算法.

上述区间树建模的相关文献对其结构进行手工验证或未验证. 本文提出了一种新的IntervalTree+结构,该结构对IntervalTree结构的节点附加颜色信息,并在其基本操作中添加了颜色及平衡调整函数,保证了树的平衡,最坏时间复杂度由$O(n)$改进到$O(\log n)$. 进一步,对IntervalTree+结构的操作进行了机械化验证,有效地保证了其正确性.

(2)区间树验证方面. 文献[12]通过手工归纳并验证了其操作函数的正确性性质定理. 文献[2]对二叉搜索树结构进行扩充并在Isabelle中实现了IntervalTree结构及插入、删除、搜索等操作的函数式建模及其验证. 文献[16]将Isabelle中auto2证明器应用于命令程序的验证,具体证明过程分为验证程序的功能版本和使用分离逻辑将其细化为命令版本,其中验证实例包括基于二叉搜索树的区间树.

文献[12]仅通过手工推导对其基本操作进行正确性分析,涉及大量的分析、归纳和证明过程,过程复杂、易出

错. 文献[2,16]提出的IntervalTree在Isabelle中进行机械化验证,其搜索IntervalTree中某区间的最坏时间复杂度为$O(n)$. 本文提出了IntervalTree+结构,并对其操作进行了机械化验证,且最坏时间复杂度优化为$O(\log n)$.

3 IntervalTree+结构的函数式建模

IntervalTree在插入顺序有序区间集合的情况下会退化为链表结构,导致其高度变为N(N为区间的数量级),进而导致插入、删除和搜索操作最坏时间复杂度达到$O(n)$. 本文提出了一种IntervalTree+结构,该结构基于IntervalTree,在区间扩充的二叉搜索树的每个节点中附加了颜色属性,具有如下优势:(1)在插入或删除节点时,尽可能地通过调整节点颜色信息来减少树的旋转次数,使其在频繁进行插入或删除操作时操作效率更高;(2)同时给出了颜色及平衡调整函数,使IntervalTree+结构具有自平衡性,避免了其退化为链表结构的情况,其插入、删除和搜索操作最坏时间复杂度改进到$O(\log n)$.

3.1 IntervalTree+结构的定义

二叉搜索树的'a tree类型使用Isabelle提供的归纳类型datatype定义,IntervalTree在二叉搜索树结构的基础上,对其节点附加区间信息. 基于IntervalTree结构,本文对其节点进行扩充,附加额外了color属性,提出了IntervalTree+结构并使用类型别名关键字type_synonym定义了'a tree类型的同义词RBs_ivl_tree,IntervalTree+结构中的节点信息由一个二元组构成,其中'a ivl类型表示为闭区间类型,'a是一个泛化类型,可以实例化为自然数或实数等类型,IntervalTree+结构的定义如下:

type_synonym 'a RBs_ivl_tree
="(('a ivl × 'a) × color) tree"

IntervalTree+内部节点包含一个区间信息,每个节点都具有颜色属性(红色或者黑色),其含义如下:通过设置红色和黑色属性来标识节点间的关系,并定义颜色约束规则(规则3~规则5)确保IntervalTree+平衡. 此外,将黑色节点高度(黑色节点最大数量的路径)定义为IntervalTree+的高度. IntervalTree+满足以下规则,其中规则1和规则2保留了文献[2]中IntervalTree所满足结构性质,规则3~规则5是本文IntervalTree+结构需要满足的新性质,即保持平衡的特性.

规则1:IntervalTree+节点的中序遍历产生一个线性升序的列表.

规则2:IntervalTree+的每个节点记录着当前节点及其子树中区间上界的最大值.

规则3:对于任意的内部节点所形成的路径中,不能存在2个连续的红色节点.

规则4:对于任意的内部节点到其各个叶子节点所形成的路径中,黑色节点的数量需要相同.

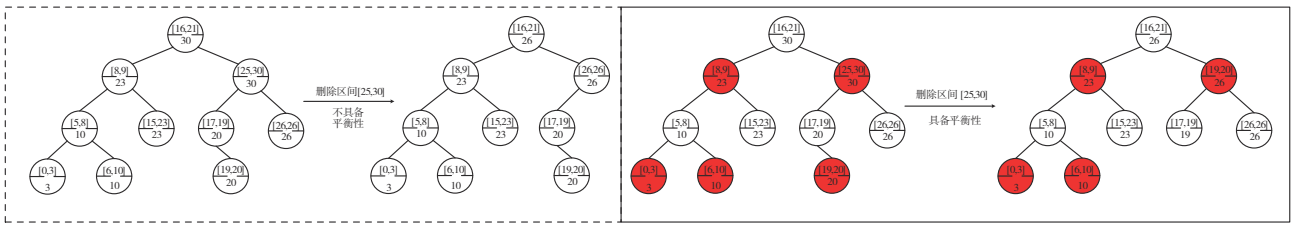
规则5:根节点的颜色为黑色.

3.2 IntervalTree+基本操作的函数式建模

IntervalTree+结构支持插入、删除、搜索等操作. 插入和删除操作在IntervalTree+中插入或删除某一节点后生成的树仍然保持IntervalTree+结构;搜索操作用于搜索IntervalTree+树是否与给定某区间有重叠部分. 限于篇幅,本节选择删除操作进行描述.

本文删除操作建模的基本思想为:若对空树进行删除,则不进行任何操作;若对非空树进行删除,则对该树进行遍历,找到需要删除的节点,并用该树右子树中区间下界最小的元素替换所要删除的节点,然后对树进行平衡及颜色调整. 删除函数被形式化为del函数,其在完成删除操作时,根结点颜色可能为红色,破坏了IntervaTree+结构在3.1节中定义的规则5,需要将其颜色设置为黑色,并且通过node函数求出树中每一个节点当前记录的区间上界最大值. 最后将删除操作定义为delete_IntervalT函数.

图1(a)为文献[2]中的IntervalTree进行删除操作后,树中每条路径的高度不同,不具备平衡性. 图1(b)为本文的IntervalTree+进行删除操作后,对整棵树进行平衡调整,使每条路径高度相同,具有平衡性.



(a) IntervalTree (b) IntervalTree+

图1 删除操作对比

3.3 时间复杂度对比分析

IntervalTree+结构在每个节点附加了颜色属性,并且在插入、删除操作时进行了平衡及颜色调整,与Nipkow等人[2]的IntervalTree结构相比,本文结构具有自平衡特性. IntervalTree+结构基本操作的最坏时间复杂度如表1所示,其中n为树中结点的个数.

表1　时间复杂度对比

	IntervalTree 最坏时间复杂度	IntervalTree+ 最坏时间复杂度
插入操作	$O(n)$	$O(\log n)$
删除操作	$O(n)$	$O(\log n)$
搜索操作	$O(n)$	$O(\log n)$

由3.1节中的规则3和规则4可知,在IntervalTree+树中,从根节点到每一个叶子节点路径中的黑色节点的数量相同并且不能存在连续的2个红色节点,每个叶子节点的深度最多是其他叶子节点的2倍. 由此可以得到IntervalTree+的高度与满足当前树高满二叉树之间的关系为:$h\ \text{tree} \leqslant 2\lg\left(|\text{tree}|_1+1\right)$. 其中,$h$ tree为树的高度;$|\text{tree}|_1$为树tree满二叉树时节点的个数. 详细的证明过程如下.

证明　由规则3和规则4,同一路径2个相邻黑色节点之间有且只有一个红色节点. 因此,树的高度最多为该路径中(黑色节点数量×2+1),树tree的高度h tree与黑色节点高度bh tree之间的关系如下:

$$\begin{aligned}&\text{invc tree} \wedge \text{invh tree} \rightarrow h\ \text{tree} \leqslant 2\text{bh tree}\\&\quad +(\text{if inv}_{\text{color}}\ \text{tree} = \text{Black then 1 else 0})\end{aligned}.$$

可以得到

$$\frac{h\ \text{tree}}{2} \leqslant \text{bh tree} \tag{1}$$

根据规则3和规则4可得bh tree$\leqslant h$ tree. 因此,$2^{\text{bh tree}}-1 \leqslant \left(2^{h\text{tree}}-1=|\text{tree}|_1\right)$,得到

$$\text{invc tree} \wedge \text{invh tree} \rightarrow 2^{\text{bh tree}}-1 \leqslant |\text{tree}|_1 \tag{2}$$

于是

$$2^{\frac{h\ \text{tree}}{2}} \leqslant 2^{\text{bh tree}} \leqslant |\text{tree}|_1+1 \tag{3}$$

注意到IntervalTree+的高度h tree受限于$|\text{tree}|_1$,则IntervalTree+的高度与黑色节点高度之间的关系为

$$h\ \text{tree} \leqslant 2\lg\left(|\text{tree}|_1+1\right) \tag{4}$$

综上,IntervalTree+基本操作最坏时间复杂度均为$O(\log n)$.

4　IntervalTree+结构的机械化验证

IntervalTree+操作函数的验证分为终止性证明和功能正确性验证[17]. 对于函数的终止性证明,本文实现的函数式建模脚本均是通过fun和definition定义,终止性在Isabelle中已被自动检查并验证. 因此,仅需对操作函数功能正确性进行验证.

4.1　IntervalTree+的不变量

不变量是指程序在执行过程中必须遵守的逻辑规则[18],对程序进行正确性验证时,可以通过验证其不变量在执行操作前后均为真来完成[19]. 对IntervalTree+的不变量分为两类.

(1)结构不变量:数据结构所遵守的结构性质的逻辑规则,本文形式化定义了5个结构不变量,对应3.1节中描述的5条规则.

(2)元素不变量:数据结构进行操作前后,其他元素值及个数保持不变的逻辑规则. 以下给出删除操作的元素不变量定义:

$$\begin{aligned}&\text{invar\_BItree tree} \Rightarrow \text{set\_tree(delete } x \text{ tree)}\\&\qquad = \text{set\_tree tree} - \{x\}\end{aligned}.$$

4.2　结构不变量的正确性验证

相较于IntervalTree+其他操作函数正确性验证,删除函数的正确性验证考虑的情况更多,验证难度更大. 因此,本节仅给出IntervalTree+删除操作结构不变量的正确性验证过程,可构造为定理1.

定理1　theorem BIT_delete: "invar_BItree tree⇒invar_BItree (delete_IntervalT x tree)".

定理1表示IntervalTree+树删除操作前满足结构不变量能推出删除操作后也满足其对应结构不变量. 本节通过不变量和删除操作函数定义将定理1展开得到中间目标引理. 接着,对中间目标引理中涉及的函数构造一系列辅助引理,使用归纳法和slegehammer对辅助引理进行证明. 最终,通过对一系列辅助引理的证明即可完成对定理1的证明. 删除操作结构不变量的证明过程如下.

(1)通过不变量定义展开中间目标引理

定理1首先通过IntervalTree+结构不变量定义展开得到5个中间目标引理,5个中间目标引理分别为删除操作前后需要满足的5个结构不变量情况. 然后,使用auto证明方法,将删除操作delete_IntervalT函数定义和结构不变量invar_BItree函数定义对定理1化简,可以得到下面5个中间目标引理.

引理1　每个节点在删除前后记录着当前节点及其子树中区间上界的最大值.

引理2　每个节点在删除前后的中序遍历产生一个线性升序的列表.

引理3　IntervalTree+删除前后任意的内部节点所形成的路径中,不能存在两个连续的红色节点.

引理4　IntervalTree+删除前后任意的内部节点到其各个叶子节点所形成的路径中,黑色节点的数量相同.

引理5　删除前后根节点的颜色为黑色. 引理1~引理5分别对应3.1节中描述的5条规则.

(2)构造辅助引理

为了验证中间目标引理,还需构造辅助引理. 以引理2证明举例,即IntervalTree+在删除操作前后每个节点的中序遍历都是一个线性升序的列表. 本文将IntervalTree+删除操作转换为对IntervalTree+中序遍历得到

的列表进行删除元素的操作，并验证IntervalTree+中删除一个节点后，对其中序遍历的列表的元素等价于对其中序遍历得到的列表中删除该节点元素．因此，构造辅助引理2A完成对引理2的证明，引理2A证明如下：

引理2A lemma inorder_del：＂sorted（inorder tree）⇒ inorder（delete_IntervalT x tree）=del_list x（inorder tree）＂.

（3）使用归纳法和slegehammer完成证明

引理2A的证明涉及删除函数，而删除函数涉及多种自平衡情况，只需构造删除节点后使用自平衡函数得到的IntervalTree+中序遍历列表是升序的即可．由于列表在Isabelle中是归纳类型定义，通过归纳法和slegehammer即可自动完成对引理2A的证明．

完成引理2A的证明后，通过metis化简方法即可完成引理2证明．引理1和引理3~引理5与引理2的证明思路类似．在Isabelle中对引理1~引理5进行机械化证明后，定理1即可得到证明．

4.3 元素不变量的正确性验证

IntervalTree+元素不变量定义可见4.1节．本节以删除操作的元素不变量正确性验证为例．

定理2 theorem set_delete：＂invar_BItree tree⇒set_tree（delete_IntervalTxtree）= set_tree tree−{x}＂.

定理2表示删除IntervalTree+中的元素后的集合等价于在该IntervalTree+元素集合中减去该元素．本文证明定理2的思路：首先使用结构不变量invar_BItree函数定义化简展开可以得到中间目标引理；然后，将IntervalTree+的操作转换为对IntervalTree+中序遍历列表的操作，构造辅助引理6，IntervalTree+中序遍历得到的列表中删除一个元素转换的集合等价于在IntervalTree+节点转换的集合直接删除该元素．

引理6 lemma set_del_list：＂sorted xs⇒set（del_list x xs）= set xs −{x}＂.

引理6在Isabelle中只需使用induct归纳方法对其进行归纳证明，使用induct归纳方法得到的子目标均可被auto方法自动完成证明．通过4.2节已证明的引理2A和上述引理6，定理2即可证明成功．

5 IntervalTree+结构的应用

IntervalTree+是一种对动态区间集合进行维护的搜索树，可用于高效地存储和查询区间，能够快速地搜索树中与某个特定区间所重叠的区间，在处理与区间相关的算法或系统中被广泛应用，例如隐私保护视觉日志服务、新文档评估模型、区域匹配算法等方面．

传统的区域匹配算法[20~22]实现方法效率较低且算法未得到正确性验证，使用IntervalTree+对区域匹配算法建模，能够高效地对大规模、高维度的区间集合进行处理，并且优化区域匹配算法的效率．基于IntervalTree+结构与区域匹配算法通用验证规约，本文建模和验证了数据分发管理系统二维区域匹配算法（FRMA）和威胁辐射源多维区域匹配算法（Functional Multidimensional Region Matching Algorithm，FMRMA），其实现路线如图2所示．限于篇幅，本节仅展示FRMA的函数式建模和机械化验证过程．

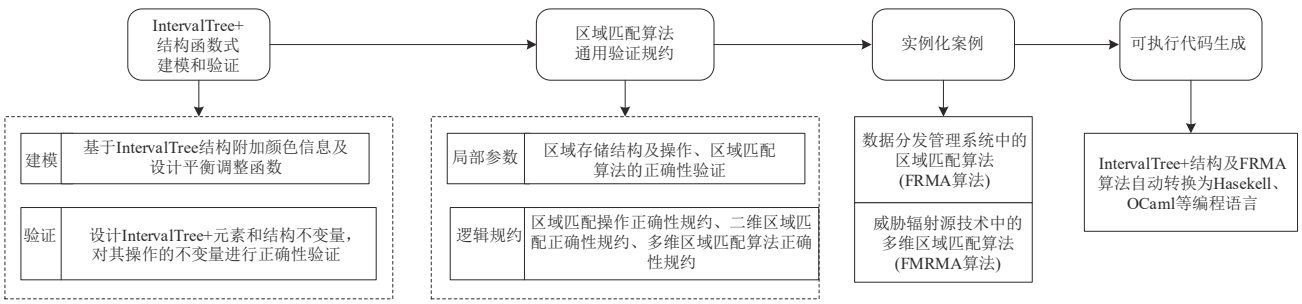


图2 本文区域匹配算法实现路线

5.1 基于IntervalTree+的DDM区域匹配算法函数式建模

数据分发管理（Data Distributed Management，DDM）[23]可以有效地降低网络冗余数据，其功能是实现数据的过滤，其中区域匹配算法直接影响数据分发管理的过滤效率，其核心问题是判断公布域和订购域是否重叠[24]，即矩阵相交问题．目前诸多的区域匹配算法在效率上不够理想，然而基于区间树的区域匹配算法[23]可以大大减少匹配计算的时间，能有效确定集合中的任意两个矩阵是否重叠[2]．本文FRMA的基本思想是：首先要建立一个多维坐标系统；然后，将公布区域和订购区域的相同维的坐标值映射到一个同维坐标系统中，从该同维坐标系统中选择公布区域的区间集构建公布区域区间树；接着，依次使用该坐标系统订购区域的区间集在上述构建的区间树中进行搜索，寻找公布区域和订购区域是否存在重叠区间．以二维坐标系统举例，如图3所示．在二维坐标系统的路径空间中，公布区域（Publication）分别为P1、P2，订购区域（Order）分别为S1、S2，其数据类型定义如下：

type_synonym 'a Publication = "'a Region list",

type_synonym 'a Order = "'a Region list".

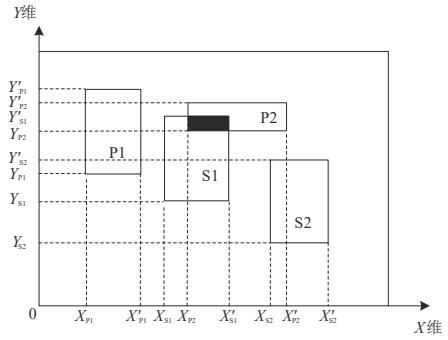


图3　二维坐标系统中区域分布图

每一个公布区域和订购区域由区域列表组成,其中区域定义如下:

type_synonym 'a Region = "'a ivl × 'a ivl".

图3中所示的二维坐标系统Isablle的定义如下:

type_synonym 'a Coordinate_System
= "'a Publication × 'a Order"

接下来,将二维坐标系统中的公布区域和订购区域分别映射到同维区间集,构建二维IntervalTree+. 图4给出了图3中的公布区域和订购区域映射到X维区间集的示例.

公布区域和订购区域映射的某一维区间集的操作可构建为IntervalTree+的插入、删除和搜索操作. 最后,将订购区域映射的区间集在公布区域构建的IntervalTree+中进行匹配,寻找公布区域和订购区域是否存在重叠区间.

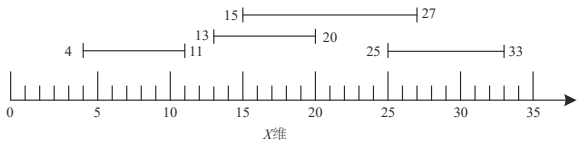


图4　公布区域和订购区域的区间集

5.2　二维区域匹配算法的通用验证规约

基于Nipkow[25]提出的二叉搜索树验证框架,本文提出了区域匹配算法的通用验证规约,该通用验证规约可实例化到各类二叉搜索树中,从而可验证不同场景下的区域匹配算法.

区域(locale)是一种程序模块化和参数化的复用机制,能充分表达函数式程序结构之间复杂的依赖关系[11]. 通过区域声明可定义通用的验证规约,以规范定理和证明的作用范围并可对其实例化. 因此,基于locale给出了区域匹配算法的通用验证规约Region_Match,如图5所示.

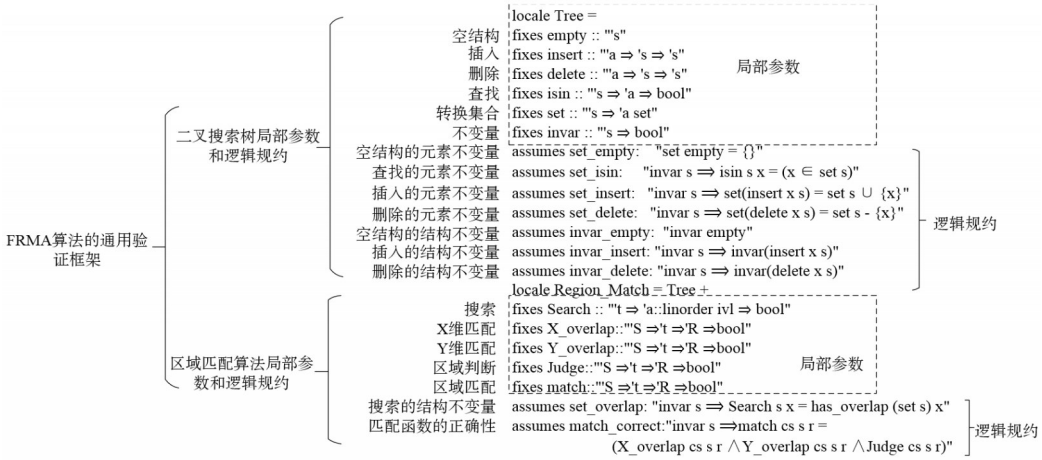


图5　区域匹配算法通用验证规约

在区域匹配算法通用验证规约中,本文使用fixes关键字定义了区域匹配算法的高阶泛化局部参数. 它们之间的逻辑规约使用assumes关键字定义,以上高阶泛化的局部参数可根据不同应用场景将区域匹配算法实例化到各类二叉搜索树中,使得区域匹配算法更加灵活和具有可扩展性.

5.3　基于IntervalTree+的DDM区域匹配算法机械化验证

区域定义后,可以在程序的上下文中进行动态地解释,这样区域中的所有信息(包括内部函数等)都被传递到当前上下文中,从而可以在当前上下文中进行复用[11]. 通过interpretation命令可将区域及其内部信息传递到当前上下文将其实例化. 基于5.2节所提的区域匹配算法通用验证规约Region_Mtach,通过将其区域解释可得到FRMA,如图6所示.

区域解释实例化为FRMA后的逻辑规约验证过程如图7所示. 对于区域匹配算法正确性验证,本文通过构造正确性定理对其逻辑规约进行证明. 在FRMA的逻辑规约证明过程中,区域匹配算法的9条逻辑规约需要构造7条定理进行验证. 其中,空结构的2条逻辑规约

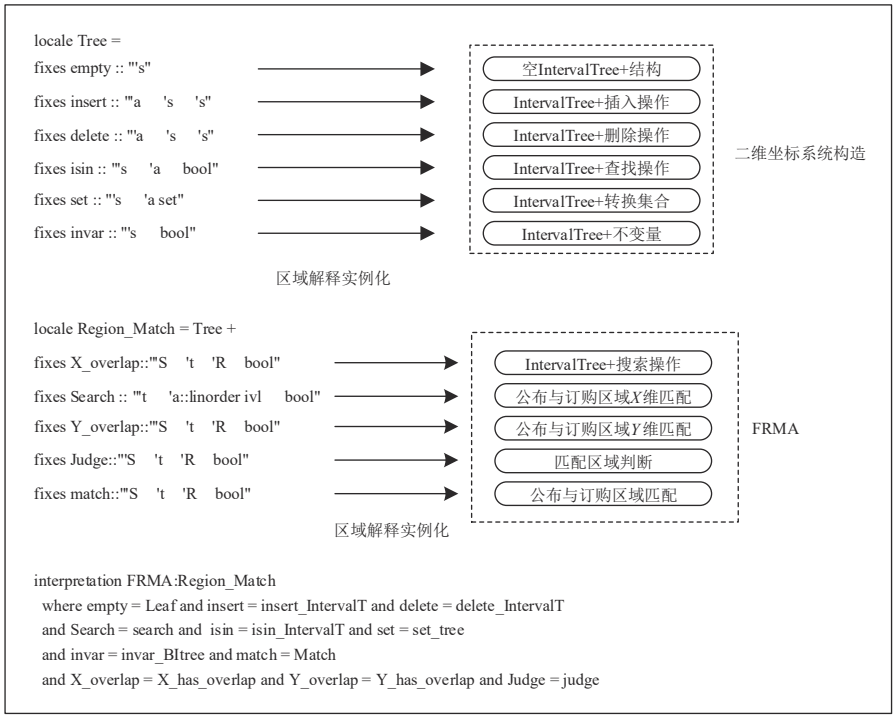


图6 区域解释验证区域匹配算法

图7 实例化FRMA算法后逻辑规约的证明过程

set_empty 和 invar_empty 可通过函数定义完成证明，无需构造定理；剩余7条逻辑规约，本文通过构造定理1~定理7完成验证. 定理1和定理2分别是IntervalTree+删除操作逻辑规约构造的正确性定理；定理6和定理7分别是IntervalTree+插入操作逻辑规约构造的正确性定理；定理5是IntervalTree+查找操作逻辑规约构造的正确性定理；定理3是IntervalTree+搜索操作逻辑规约构造的正确性定理；定理4是区域匹配操作逻辑规约构造的正确性定理. 该7条定理均在Isabelle中被机械化证明.

使用Isabelle/HOL内置的代码生成命令export_code[26]，可以将验证后的DDM区域匹配算法自动转换成相应的Haskell语言的可执行程序. 此外，可以将本文提出的二维区域匹配算法通用验证规约扩展到多维区域匹配，并应用于威胁辐射源区域匹配算法中.

6 总结与展望

本文基于Nipkow等人提出的IntervalTree结构，对其节点附加额外的颜色信息，同时保证了树的平衡，首次提出了IntervalTree+结构并在Isabelle/HOL实现了插入、删除、搜索等操作函数式建模，通过严格的机械化验证保证了其正确性，其操作最坏时间复杂度相较于IntervalTree结构由$O(n)$改进到$O(\log n)$.

本文还首次提出了区域匹配算法通用验证规约，

该验证规约可以根据不同应用场景通过区域解释实例化到各类区域匹配算法中. 基于IntervalTree+结构与区域匹配算法通用验证规约,本文建模和验证了数据分发管理系统二维区域匹配算法. 未来的研究工作可以将本文提出的验证规约扩展到多维区域匹配算法中,建模和验证更多涉及区域匹配算法的应用.

参考文献

[1] MEHTA D P, SAHNI S. Handbook of Data Structures and Applications[M]. New York: Chapman and Hall/CRC, 2004.

[2] NIPKOW T, BLANCHETTE J, EBERL M. Functional algorithms, verified![EB/OL]. [2024-05-09]. https://functional-algorithms-verified.org.

[3] GRAEFE G. B-tree indexes for high update rates[J]. ACM Sigmod Record, 2006, 35(1): 39-44.

[4] GYTING R H, BEHR T. SECONDO: A platform for moving objects database research and for publishing and integrating research implementations[EB/OL]. [2024-05-09]. https://secondodatabase.github.io/files/papers/PaperPlugin-s.pdf.

[5] KAUSHIK R, NAUGHTON J F, BOHANNON P, et al. Updates for structure indexes [C]//VLDB'02: Proceedings of the 28th International Conference on Very Large Databases. Amsterdam: Elsevier, 2002: 239-250.

[6] LEE M L, HSU W, JENSEN C S, et al. Supporting frequent updates in R-Trees: A bottom-up approach[C]//Proceedings 2003 VLDB Conference. Amsterdam: Elsevier, 2003: 608-619.

[7] XU J Q, WEI J H. Efficiently update disk-resident interval tree[C]//Spatial Data and Intelligence. Cham: Springer International Publishing, 2021: 198-207.

[8] PHAM V A, HOANG D H, CHUNG-NGUYEN H H, et al. Privacy preserving visual log service with temporal interval query using interval tree-based searchable symmetric encryption[C]//Proceedings of the 10th International Symposium on Information and Communication Technology. New York: ACM, 2019: 425-432.

[9] XIONG Z Y, WANG Y J. New document scoring model based on interval tree[J]. Journal of Visual Languages & Computing, 2018, 45: 39-43.

[10] PAULSON L C, NIPKOW T, WENZEL M. From LCF to isabelle/HOL[J]. Formal Aspects of Computing, 2019, 31(6): 675-698.

[11] 赵永望. 函数式程序设计与证明[EB/OL]. [2024-05-09]. https://www.yuque.com/zhaoyongwang/fpp/.

[12] CORMEN T H. Introduction to Algorithms[M]. 4th edition. Cambridge: MIT Press, 2022.

[13] BERTINO E, ATZENI P, TAN K L, et al. Boosting the accuracy of differentially-private histograms through consistency[J]. Proceedings of the VLDB Endowment, 2010, 3(1-2): 1021-1032.

[14] 李丽, 张琳, 王汝传. 基于动态区间树的差分隐私数据发布算法[J]. 南京邮电大学学报(自然科学版), 2017, 37(4): 103-112.
LI L, ZHANG L, WANG R C. Differential privacy data publishing algorithm based on dynamic interval tree[J]. Journal of Nanjing University of Posts and Telecommunications (Natural Science Edition), 2017, 37(4): 103-112. (in Chinese)

[15] DROUIN A, HOCKING TD, LAVIOLETTE F. Maximum margin interval trees[C]//31st Conference on Neural Information Processing Systems. New York: Curran Associates, 2017: 4947-4956.

[16] ZHAN B H. Verifying imperative programs using Auto2[EB/OL]. [2024-05-09]. https://www.isa-afp.org/brow-ser_ info/current/AFP/Auto2_Imperative_HOL/document.pdf.

[17] 左正康, 黄志鹏, 黄箐, 等. LLRB算法的函数式建模及其机械化验证[J]. 软件学报, 2024, 35(11): 5016-5039.
ZUO Z K, HUANG Z P, HUANG Q, et al. Functional modeling and mechanized verification of LLRB algorithm[J]. Journal of Software, 2024, 35(11): 5016-5039. (in Chinese)

[18] 左正康, 柯雨含, 黄箐, 等. Trie+结构函数式建模、机械化验证及其应用[J]. 软件学报, 2024, 35(9): 4242-4264.
ZUO Z K, KE Y H, HUANG Q, et al. Trie+ structural functional modeling, mechanized verification and application[J]. Journal of Software, 2024, 35(9): 4242-4264. (in Chinese)

[19] BACK R J. Invariant based programming: Basic approach and teaching experiences[J]. Formal Aspects of Computing, 2009, 21(3): 227-244.

[20] BOUKERCHE A, LU K Y. A novel approach to real-time RTI based distributed simulation system[C]//38th Annual Simulation Symposium. Piscataway: IEEE, 2005: 267-274.

[21] AYANI R, MORADI F, TAN G. Optimizing cell-size in grid-based DDM[C]//Proceedings Fourteenth Workshop on Parallel and Distributed Simulation. Piscataway: IEEE, 2000: 93-100.

[22] PAN K, TURNER S J, CAI W T, et al. An efficient sort-based DDM matching algorithm for HLA applications with a large spatial environment[C]//21st International Workshop on Principles of Advanced and Distributed Simulation. Piscataway: IEEE, 2007: 70-82.

[23] 尚福华, 张海波, 解红涛. 区间树在DDM区域匹配中的应用[J]. 计算机工程与应用, 2013, 49(11): 110-113, 165.
SHANG F H, ZHANG H B, XIE H T. Application of interval-tree in region matching for DDM[J]. Computer Engineering and Applications, 2013, 49(11): 110-113, 165. (in Chinese)

[24] 王磊, 张慧慧, 李开生, 等. 基于动态R-树结构的DDM区域匹配算法[J]. 计算机工程, 2008, 34(3): 56-58.
WANG L, ZHANG H H, LI K S, et al. Region matching algorithm for DDM based on dynamic R-tree[J]. Computer Engineering, 2008, 34(3): 56-58. (in Chinese)

[25] NIPKOW T. Automatic functional correctness proofs for functional search trees[C]//International Conference on Interactive Theorem Proving. Cham: Springer International Publishing, 2016: 307-322.

[26] HAFTMANN F, BULWAHN L, NIPKOW T. Code generation from Isabelle/HOL theories[EB/OL]. [2024-05-09]. https://isabelle.in.tum.de/dist/Isabelle2023/doc/codegen.pdf.

作者简介

左正康 男,1980年3月生于江西抚州. 现为江西师范大学计算机信息工程学院教授、硕士生导师. 主要研究方向为高可信与智能化软件.

王昌晶 男,1977年10月生于江西丰城. 现为江西师范大学计算机信息工程学院教授、博士生导师. 主要研究方向为高可信软件,智能化软件.

张晗庆 男,2000年12月生于江西抚州. 现为江西师范大学计算机信息工程学院硕士研究生. 主要研究方向为形式化方法.

游 珍 女,1982年6月生于江西高安. 现为江西师范大学计算机信息工程学院副教授、硕士生导师. 主要研究方向为形式化方法、分布式虚拟现实、并发分布式计算.
E-mail: youzhen@jxnu.edu.cn