

文章编号: 0529-6579 (1999) 02-0001-05

一种基于事例推理的检索模型^{*}

印 鉴, 李师贤

(中山大学计算机科学系, 广东 广州 510275)

摘 要: 以计算逻辑为理论基础设计了 1 种对象检索模型以解决定理机器证明中的“同理可证”问题, 该模型采用基于事例的推理方法来进行推理, 并将领域知识、控制性知识与事例知识分别统一到对象-类层次结构上, 因此结构清晰, 可重用性高.

关键词: 定理机器证明; 计算逻辑; 同理可证; 基于事例的推理

中图分类号: TP 18 **文献标识码:** A

1979 年, Robert 等^[1]提出了定理机器证明中的数学归纳法, 并且经过 10 多年的努力建立了 Boyer-Moore 定理证明器. 这种方法来源于 Goodstain 建立的递归数论, 它把递归函数论、数学归纳法、与定理机器证明巧妙地结合起来, 开辟了定理机器证明的新途径. 本文以他们创立的计算逻辑为理论基础, 设计了 1 种模型, 用以解决定理机器证明中的“同理可证”问题, 该模型采用基于事例的推理方法来进行推理.

1 问题的提出

首先看 2 个例子.

例 1 证明定理 COMMUTATIVITY.OF.PLUS(重写)

P: (EQUAL (PLUS X Y) (PLUS Y X))

限于篇幅, 证明中没加说明的一些引理及一些函数的定义, 可参见文[1, 2].

证明 引用归纳法原理, 按下述模式作归纳

(AND (IMPLIES (ZEROP X) (p X Y))

(IMPLIES (AND (NOT (ZEROP X)) (p (SUB1 X) Y))

(p X Y)))

不等式 SUB1.LESSP 建立, 在该模式的归纳步骤中, 测度(COUNT X)按良基关系 LESSP 减小, 上述归纳模式生成 2 个新公式.

情况 1: (IMPLIES (ZEROP X) (EQUAL (PLUS X Y) (PLUS Y X)))

展开 ZEROP 的定义后, 上述又可分为 2 个公式.

^{*} 基金项目: 国家自然科学基金重点项目 (69733030-1)

收稿日期: 1998-05-18

作者简介: 印鉴, 男, 1969 年生, 博士, 副教授.

$$(1) \quad (\text{IMPLIES } (\text{EQUAL } X \ 0) (\text{EQUAL } (\text{PLUS } X \ Y) (\text{PLUS } Y \ X)))$$

引用引理 PLUS. RIGHT. ID, 并展开 PLUS 的定义后, 上式简化为 T.

$$(2) \quad (\text{IMPLIES } (\text{NOT } (\text{NUMBERP } X)) (\text{EQUAL } (\text{PLUS } X \ Y) (\text{PLUS } Y \ X)))$$

展开函数 PLUS 后, 上式可简化为

$$(\text{IMPLIES } (\text{AND } (\text{NOT } (\text{NUMBERP } X)) (\text{NUMBERP } Y)) \\ (\text{EQUAL } Y (\text{PLUS } Y \ X)))$$

记上式为 p_1 .

$$\text{情况 2: } (\text{IMPLIES } (\text{AND } (\text{NOT } (\text{ZEROP } X))$$

$$(\text{EQUAL } (\text{PLUS } (\text{SUB1 } X) \ Y) (\text{PLUS } Y (\text{SUB1 } X)))$$

$$(\text{EQUAL } (\text{PLUS } X \ Y) (\text{PLUS } Y \ X)))$$

展开函数 ZEROP, PLUS, 用 $(\text{ADD1 } Z)$ 替代 X , 并增加引入 SUB1 时提到的类型限制引理, 然后再使用等式假设, 把 $(\text{PLUS } Z \ Y)$ 交融成 $(\text{PLUS } Y \ Z)$ 并抛弃该等式可得

$$(\text{IMPLIES } (\text{NUMBERP } Z) \\ (\text{EQUAL } (\text{ADD1 } (\text{PLUS } Y \ Z)) (\text{PLUS } Y (\text{ADD1 } Z))))$$

再用归纳法原理证明上式, 按下述模式进行归纳

$$(\text{AND } (\text{IMPLIES } (\text{ZEROP } Y) (p \ Y \ Z)) \\ (\text{IMPLIES } (\text{AND } (\text{NOT } (\text{ZEROP } Y)) (p \ (\text{SUB1 } Y) \ Z)) \\ (p \ Y \ Z)))$$

不等式 SUB1. LESSP 建立: 在该模式的归纳步骤中, 测度 $(\text{COUNT } Y)$ 按良基关系 LESSP 减小, 上述归纳模式产生 2 个新推测.

$$(1) \quad (\text{IMPLIES } (\text{AND } (\text{ZEROP } Y) (\text{NUMBERP } Z)) \\ (\text{EQUAL } (\text{ADD1 } (\text{PLUS } Y \ Z)) (\text{PLUS } Y (\text{ADD1 } Z))))$$

展开 PLUS 的定义后, 上式可简化为 T.

$$(2) \quad (\text{IMPLIES } (\text{AND } (\text{ZEROP } Y) \\ (\text{EQUAL } (\text{ADD1 } (\text{PLUS } (\text{SUB1 } Y) \ Z)) \\ (\text{PLUS } (\text{SUB1 } Y) (\text{ADD1 } Z))) \\ (\text{NUMBERP } Z)) \\ (\text{EQUAL } (\text{ADD1 } (\text{PLUS } Y \ Z)) (\text{PLUS } Y (\text{ADD1 } Z))))$$

展开 ZEROP 及 PLUS 的定义后, 使用等式假设, 把 $(\text{ADD1 } (\text{PLUS } (\text{SUB1 } Y) \ Z))$ 交融成 $(\text{PLUS } (\text{SUB1 } Y) (\text{ADD1 } Z))$ 后, 上式可简化为 T.

再用归纳法原理, 易证 p_1 .

例 2 证明定理 COMMUTATIVITY. OF. TIMES

$$(\text{EQUAL } (\text{TIMES } X \ Y) (\text{TIMES } Y \ X))$$

照例 1 的思路, 不难给出证明.

两点说明: ① 从上面这 2 个例子可以看出, 除了极少数细节外, 其证明过程几乎完全一样, 可以借用数学上的术语“同理可证”来描述它. 事实上, “同理可证”是数学定理证明中不可缺少的技术, 具有易读、简洁的特点. 从数学理论上讲, “证明 A”与“证明 B”同理是指“证明 A”与“证明 B”或者用了同一定理, 或者用了同一方法, 即 2 个证明同构, 而让机器证明这种同构目前还没有有效的算法, 但机器学习可提供一种指导证明的方法, 这正如不太懂计算

逻辑的人在看了例 1 之后, 也会较快地证明例 2; ② 在数学定理中, 某些规律具有普遍性和典型性, 如交换律, 对加法、乘法、集合的交运算、并运算等等都成立. 现在就产生这样的问题, 给定函数 F (也可称之为算子), 当研究其性质时, 是否可把那些具有普遍性的规律“移植”到其上呢? 如对 PLUS (加) 和 TIMES (乘), 它们都具有 ASSOCIATIVITY (结合性), 当引入函数 APPEND 时, 发现它也具有这一性质, 并可依样给出其证明过程. 当然, 也有运用不成功的例子 (如测试 APPEND 的 COMMUTATIVITY (交换) 性质时, 就不成功). 这里所设计的模型, 正是针对上面的 2 个问题, 采用了基于事例的推理方法 (case-based reasoning, CBR).

2 模型

2.1 事例の表示

结合计算逻辑与 LISP 语言的特点, 采用框架知识表示方法来描述某个事例. 框架的形式定义为

$\langle \text{框架} \rangle ::= (\langle \text{框架名} \rangle \langle \text{结构槽集} \rangle)$
 $\langle \text{结构槽集} \rangle ::= \langle \text{结构槽} \rangle | \langle \text{结构槽} \rangle \langle \text{结构槽集} \rangle$
 $\langle \text{结构槽} \rangle ::= (\langle \text{结构槽名} \rangle \langle \text{侧面集} \rangle)$
 $\langle \text{结构槽名} \rangle ::= \text{FORM} | \text{MODULE} | \text{STRATEGY}$
 $\langle \text{侧面集} \rangle ::= \langle \text{侧面} \rangle | \langle \text{侧面} \rangle \langle \text{侧面集} \rangle$
 $\langle \text{侧面} \rangle ::= (\langle \text{侧面名} \rangle \langle \text{侧面集} \rangle)$
 $\langle \text{侧面名} \rangle ::= \text{HYPOTHESIS} | \text{CONCLUSION} | \text{TYPE} | \text{VALUE} | \text{FUNCTION} |$
 $\text{LEMMAS} | \text{自然数}$

例如，对例 1，可用框架将其描述为

```
(COMMUTATIVITY.OF.PLUS
  (FORM (CONCLUSION' (EQUAL (PLUS X Y)
                               (PLUS Y X))))
  (TYPE' REWRITE))
(MODULE
  (VALUE' (AND (IMPLIES (ZEROP X) (p X Y))
                (IMPLIES (AND (NOT (ZEROP X))
                              (p (SUB1 X) Y)) (p X Y)))))
(STRATEGY (1 STRATEGY1)
           (2 STRATEGY2)))
```

其中, STRATEGY1, STRATEGY2 可类似定义.

2.2 检索模型

这里采用了 1 种受控的 CBR 检索模型, 如图 1 所示.

在这个模型中, 事例知识与其它知识成同构的类层次结构. 事例表示成 1 个对象, 反映在类层次结构中是事例类的实例对象, 即叶结点. 约束类用于相应事例类的约束关系的描述与检索行为的控制. 其中, 约束父类是约束子类的抽象, 与事例类不同的是, 任一约束类都是虚类, 都没有实例. 具体而言, 1 个约束类包括如下内容

领域知识, 包括各种模型知识, 如定理形式、良基关系等, 它们为选择满足模型约束的事

例提供导向与结果检测, 并可为求解过程与结果提供解释.

相似度. 因为要达到事例的完全匹配几乎是不可能的, 所以必须为检索提供评判依据和启发式信息 (如属性权值). 在类层次结构下, 不同的事例类可具备不同的相似度, 并且它们可由用户根据需要动态设置.

语义关联知识. 这是一类特殊的领域知识, 因为多个事例之间很可能存在某种关系. 例如, 1 个复杂的证明过程由多项子证明聚合而成, 这些子证明之间存在着相互制约关系, 从而影响到事例类之间的语义关系, 该类型知识在进行事例适配时的作用尤其重要.

首指针. 用来确定满足此约束条件的事例集的第 1 个事例所处的位置, 余下的事例按链表方式存贮. 新事例的加入由约束类的根类完成, 领域知识亦驻留在约束类的根类中, 类层次结构上的继承关系使得父类的信息能被子类共享. 例如, 可使限制类只继承与自身 (即所控制的事例类) 有关的那一部分领域知识. 除非约束子类重新定义相似度, 否则它将从其父类中继承相似度.

例如前面所述例 1 在检索模型中的示意图如图 2 所示.

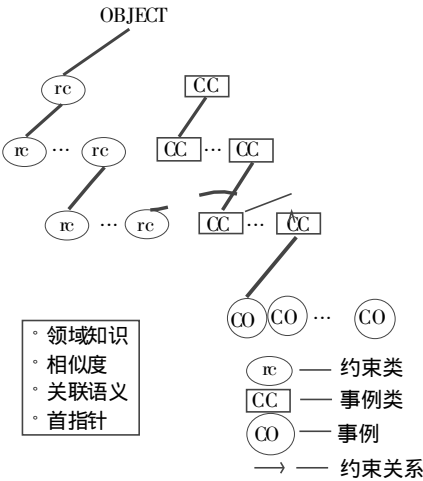


图 1 CBR 检索模型

Fig.1 The CBR retrieval model

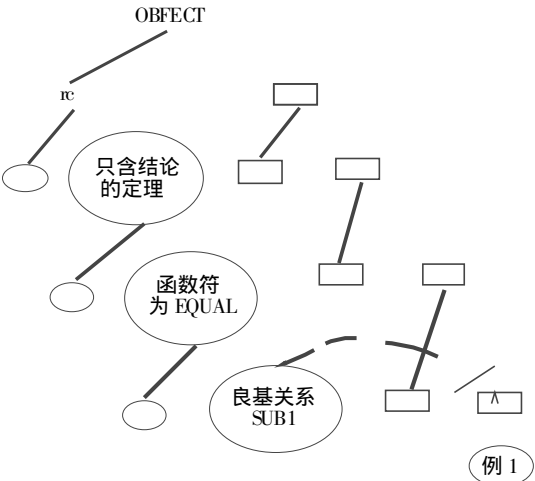


图 2 检索模型的 1 个实例

Fig.2 An example of retrieval model

在此模型下完成 1 次事例检索的过程为

- (1) 首先根据约束类根类的领域模型知识确定要检索的事例类及相应的约束类.
- (2) 在约束类的控制下 (例如, 通过相似度), 完成 1 个事例类及其子类的检索.
- (3) 根据需要, 对已检索出来的事例进行测试, 筛选, 从中挑选出满足要求的事例并按有关原则, 完成新事例策略的生成与调整.
- (4) 最后, 完成事例知识库的事例接收与有关信息的更新、索引方式与相似度以及限制类的调整.

2.3 事例的改写与学习

事例的改写操作在事例推理中相当重要. 它的主要作用 是: 适当调整在检索操作中获得的最佳事例的求解方案, 使之适合于求解当前问题. 因为事例的改写操作往往跟某个特定的领域相关, 因而很难定义通用的方式来执行事例的改写操作. 由于上面已经定义了事例的

表示方法及事例检索的模型, 因此, 事例的改写已在相当程度上由它们二者决定了. 随着新的事例不断增加, 如果不采取适当的措施, 将会使事例库变得十分庞大, 其推理效率也必受到影响. 为了将事例库控制在一定规模内, 必须对加入到事例库中的事例进行学习. 这里的学习可看成是“合并”或者是“泛化”过程, 具体来说, 当事例被加到事例库时, 有以下几种情况:

- (1) 若事例库中已有相同的事例, 则抛弃之.
- (2) 替换事例库中的旧事例.
- (3) 与事例库中的旧事例合并, 形成 1 个新的事例.
- (4) 直接作为新的事例独立地加到事例库中.

本文提出了 1 种对象检索模型, 该模型将领域知识、控制性知识与事例知识分别统一到对象-类层次结构上, 结构清晰, 可重用性高, 避免了概念与结构上的冗余. 对事例的修改与维护不会干扰到领域知识, 反之亦然. 同时, 约束类知识的更改除了影响其子类外, 不会影响到其它约束类. 因而, 其管理与维护十分方便. 通过对计算逻辑中一些定理的实验, 本模型取得了很好的效果.

参考文献:

- [1] BOYER R S, MOORE J S. A computational logic [M]. New York: Academic Press, 1979. 1 ~ 200.
- [2] 丽奇 E 著. 人工智能引论 [M]. 李卫华, 汤怡群, 文中坚编译. 广州: 广东科技出版社, 1986. 198 ~ 228.
- [3] 程显毅. 机器证明的困难所在 [J]. 计算机科学, 1992, 19(2): 81 ~ 82.
- [4] 徐明, 胡守仁. 基于事例推理的检索模型研究 [J]. 计算机科学, 1993, 20(4): 32 ~ 35.
- [5] 耿卫东, 潘云鹤, 何志均. 基于事例的设计推理研究 [J]. 计算机科学, 1993, 20(4): 36 ~ 39.

A Case-based Reasoning Retrieval Model

YN Jian^{*}, LI Shi-xian

Abstract: Based on a computational logic, an object retrieval model is presented, and is used to solve the problem of “proving by the same methods”. This model adopts inferences by using the case-based reasoning method, and unifies the field knowledge, control knowledge, and case knowledge to the object-class level respectively, so it is clear in structure, and easy to reuse.

Keywords: mechanical theorem proving; computational logic; proving by the same methods; case-based reasoning

^{*} Department of Computer Sciences, Zhongshan University, Guangzhou 510275, China