大学水平的"数学分析知识"的获取和分析研究*)

刘汉武[1,2] 曹存根[2] 曾庆田[2,3]
(首都师范大学计算机科学联合研究院 北京 100037)[1] (中国科学院计算技术研究所 北京 100080)[2]
(中国科学院研究生院 北京 100039)[3]

摘 要 在过去的几十年里,数学软件系统变得越来越强大和复杂。各种数学系统的集成变得非常有意义。基于Web的分布式数学平台使得与数学相关的活动(称之为数学服务)都可以在Web上得以实现。所有这些都需要一个底层的、可共享的、面向内容的知识库作为支撑。本文介绍一个大学水平的、可共享的数学分析知识库的获取和分析方法。NKIMath采用基于本体、框架、逻辑和类型的形式表示,采用面向概念的知识获取方法,从一套大学本科数学分析课本(2本)中获取了所有主要概念和定理。本文还总结和分析了知识获取过程中出现的错误。

关键词 数学知识,本体,框架,逻辑,类型,知识表示,知识获取

Acquisition and Analysis of University-Level Knowledge of Mathematical Analysis

LIU Han-Wu[1,2] CAO Cun-Gen[2] ZENG Qing-Tian[2,3]
(Joint Faculty of Computer Scientific Research, Capital Normal University, Beijing 100037)[1]
(Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080)[2]
(Graduate School of the Chinese Academy of Sciences, Beijing 100039)[3]

Abstract Over the past decades mathematical software systems have become increasingly powerful and complex. The integration of mathematical formal systems is highly desirable. Web-oriented platforms of distributed mathematical services enable mathematical activities to be implemented on the Web. All those require an underlying, shareable, and content-oriented mathematical repository. In this paper, based on ontology, frame representation, predicate logic and type, we present a knowledge representation for mathematical analysis. A concept-oriented knowledge acquisition has been developed and with the method we have extracted all the main concepts and theorems from a university-level textbook on mathematical analysis. The paper will also discuss possible errors that a knowledge engineer may commonly make in acquiring mathematical knowledge.

Keywords Mathematical knowledge, Ontology, Frame, Predicate calculus, Type, Knowledge representation, Knowledge acquisition

1 引言

机器自动定理证明(ATP)就是把人类证明定理的一般知识和规则以适当形式存储到计算机中,通过机器的运转,自动证明定理,既是早期人工智能研究的重要课题,又属于数学和计算机的一门交叉学科。简单地说,利用计算机证明非数值性的结果,就是自动定理证明[1~4]。

随着ATP研究的深入,基于知识的ATP得到广泛的研究。在文[5]中Freek W根据知识库的大小、逻辑表达的力度(strength)和自动化的程度这三个主要尺度比较了15个著名的数学定理证明系统,在这15个系统中有8个均需要庞大的数学知识库支持,这说明数学定理自动证明的发展对数学知识的获取存在巨大的需求。

在过去几十年里,数学软件系统变得越来越强大和复杂[6]。数学定理自动证明(ATP)和计算机代数系统(CAS,又称符号计算系统),它们有着各自的处理对象和方法。在现实的问题解决过程中常常既需要用到数学定理自动证明(推理)又要用到数学符号计算,是它们的混合使用。比如,当涉及数学对象的计算时,数学定理证明系统就无能为力,同样,虽然计算机代数系统能够进行复杂的数学计算,但是却不能够产生任何形式的证明或者推理,因此,数学自动定理证明和计算机代数系统的集成是非常有意义的。而这些系统的集成需要研究:(1)一种底层的适合多种系统的知识表示格式;(2)系统之间进行通讯的协议[7,8]。

随着计算机的出现和Web技术的兴起与发展,互联网在人们的生活中扮演着越来越重要的角色。与数学相关的活动将在Web上进行,这需要建立一个通用的分布式的数学知识Web平台,它能够保证数学知识的模块化、分配、内部操作性、协作性等。平台上的所有数学活动将通过数学服务的形式进行。这首先需要建立一个所有服务都可以访问和通讯的数学知识库。基于现有的计算技术和Web技术,我们期待所有的数学服务通过相互协作能够完成大部分的数学任务:数学知识查询,数学知识问答,数学定理自动证明,符号计算,数学智能教学,数学知识的导航、重用、共享、分布、更新、管理,

*)本文中的工作受到国家自然科学基金(课题号#60273019)和科技部重大基础研究专项(课题号#2001CCA03000)联合资助;还受到首都师范大学实验室开放资金资助。刘汉武 硕士,研究兴趣:知识获取,人工智能,分布式数学;曹存根 博士导师,研究员,研究兴趣:人工智能,知识工程,大规模知识处理,情感计算;曾庆田 博士生,研究兴趣:知识获取,知识V&V技术,Petri网理论与应用。

数学知识(新的概念、定理等)的发现,数学文献的出版等,并期望推动数学家之间的合作与交流,推动数学学科的发展[9,10]。

知识获取是知识工程的一个公认的瓶颈问题,因此知识获取受到了广泛的重视和研究[11~13]。领域知识获取的途径主要有两种,一是从领域专家处获得专业知识,二是从文本或数据库中直接获取。对数学知识而言,由于专家的研究领域和研究精力的局限,很难给出完整的学科体系,而且据统计90%以上的知识可以从文本中获取。因此,从领域文本中直接获取知识无疑是一种更可取的方法。在知识获取的过程中,我们采用人机交互的半自动的知识获取方法。具体地说,领域知识的获取工作主要分为四步:

(1)建立数学领域本体:由知识工程师在领域专家的指导下完成;

(2)领域文本的半结构化:将自然语言描述的数学知识文本形式由知识工程师转变为数学知识描述语言表示的半结构化的领域知识文本;

(3)基于数学领域本体的知识编译和检查:在领域本体的驱动下,由计算机系统自动地实现半结构化知识文本到知识库存储形式的转换;

(4)知识分析及知识链接:对获得的领域知识进行分析,检查其可能存在的异常现象(如:知识的不一致性、冗余性、非完备性等),以确保所获得的知识的可用性。

领域知识获取的前两步由人工来完成,第3步是由计算机系统自动完成,因此整个知识获取过程是一个半自动的过程。第4步是知识获取的一个重要步骤。

本文第2节介绍国内外相关研究工作。第3节介绍数学分析知识的表示和获取方法,我们以复旦大学数学系陈传璋、朱学炎、金福临和欧阳光中编写的由高等教育出版社出版的《数学分析(上下卷)》[32]教科书为基础展开研究。第4节具体讨论数学分析知识的获取过程。第5节分析数学知识获取过程中常见的错误。最后总结全文。

2 国内外相关研究工作

2.1 数学知识标记语言

MathML(Mathematics Markup Language)是数学领域的标记语言[14]。它是一个XML[15,16]应用程序,不仅用来表示数学符号,而且用来获取数学结构和内容。MathML的目的是使得数学知识能够在WWW上被服务(Served)、接受和操作处理,就像HTML对文本文件实现这一功能一样。一些重要的数学编辑软件已经提供对MathML的支持,可能会在几年内大量取代TeX[17]。

OpenMath是由XML实现的数学对象表示标准,允许程序之间的数学对象进行交互交流、允许数学对象在知识库中的存储以及在Web上的发布[18]。虽然MathML也能够获取数学对象的结构和内容,但是它的主要目的是实现数学对象的表示,而OpenMath设计的主要目的是表示数学对象的结构,也就是数学对象的内容。

OMDoc(An Open Markup Format for Mathematical Documents)是对OpenMath标准的扩展,OMDoc可以标注不同数学文档(如articles、textbooks、interactive books、course等)的结构和语义,允许不同数学系统之间进行通讯[19~22]。

2.2 数学知识工程

MBase是美国卡麦隆大学联同其他多所大学正在建设的基于Web的分布式数学知识库,它由一组分布式的MBase服务器组成,每一个MBase服务器都包括一个通过标准的数据库接口(比如JDBC)连接到一个MOZART程序(用来产生MathWeb服务)的关系数据库管理系统,MBase的作用就是存储和维护知识库中结构化的知识,其知识表示是基于OMDoc格式的[23]。

HELM工程是意大利博洛尼亚大学正在研究的一个课题,其目标是研究一套为建立、维护虚拟的分布式超文本的(hypertext)和形式化的数学知识库而需要的技术。HELM工程有一个普遍的、应用无关的元语言,相似的软件工具可以用于不同的逻辑运算而不管它们具体的细节。这个普遍的表示层虽然不能够解决不同应用程序之间的所有的内部互操作问题,但是朝着这个方向迈出了关键的第一步[24]。

2.3 中科院数学机械化研究中心

吴文俊院士在数学的机械化证明方面做出了突出贡献。国际上公认的吴方法,将几何证明问题转换为方程的求解问题。数学机械化研究中心先后主持了国家"八五"攀登计划项目"机器证明及其应用",国家"九五"攀登计划预选项目:"数学机械化及其应用"和国家重点基础研究发展规划项目:"数学机械化与自动推理平台"。通过这三个项目的实施,该中心在几何定理自动证明与发现、方程求解、微分动力系统的稳定性研究、构造性代数几何、符号计算等理论研究领域与机器人、智能CAD、几何造型、计算机视觉、软件开发等高科技领域做出了很好的工作[25]。

2.4 中科院计算所知识获取与共享课题组

国家知识基础设施(National Knowledge Infrastructure,NKI)是一个庞大的、可共享的知识群体,它不仅集成了各个学科的公共知识,而且还融入了各学科专家的个人知识[26~28]。上述学科包括医学、军事、物理、化学、数学、化学、信息科学、宗教、民俗等等,为科研、教学、科普和知识服务提供有效的基础,使知识共享成为可能。中科院计算所的曹存根研究员带领的知识获取与共享课题组正在进行关于"国家知识基础设施的前瞻性研究"。

"国家知识基础设施"(NKI)课题,已建立了包括中医、西医、数学、天文、地理、生物、教育、军事、考古、民族、宗教、音乐等16个学科的580多个专业本体,各学科本体按照继承和实现等关系形成了相对独立的体系结构。依据专业本体形式化描述了大量专业知识,目前获取的专业知识大约有几百万条。定义了过程性知识、本体的公理等的形式化表示方法,初步形成了本体建立与分析的方法和理论,并建立了本体的开发、管理和集成环境OKEE[29]。

NKIMath是NKI的数学知识库。NKIMath对整个NKI的建设是十分有意义的,一方面是由于数学知识自身的重要性,另一方面数学知识也是NKI其它学科(如:物理学、化学、机械动力学等)知识建设的基础[30,31]。

2001年,我们开始了数学知识的表示和获取工作的研究。我们已经在数论、集合论、代数学、分析学、图论等分支进行了知识获取和分析的试验,已经获得了上千个数学知识框架(包括概念、断言和例子),有效地检验了本文方法的可行性和正确性。

3 数学分析知识的表示方法

3.1 数学本体

近年来,本体论的应用越来越受到重视,很多知名的知识

系统都采用了本体论的思想。本体原本是一个哲学名词，是关于事物存在的研究。我们在这里给出的本体含义如下：本体是由概念组成的有机系统。概念是本体最基本的元素。本体中最重要的元素是概念和概念之间的关系。我们引入本体论的思想是从数学学科的概念集和概念集中概念之间的关系集出发，建立面向内容的数学知识库，从而为上面提到的数学知识 Web 平台提供数学知识库支持。

数学分析知识内容繁多。虽然难以给出一套严格的形式语言去刻画数学分析知识的表示方式，但是仔细分析所有的数学分析知识文献，其基本的知识对象主要包括：

(1)各种符号：Σ，$\cap$，$\in$，$\geqslant$等；

(2)概念的定义知识：这是数学知识最主要的描述对象，如：集合、函数、微分、积分等；

(3)断言知识：描述概念和概念之间的关系，包括公理、引理、定理、猜想等，如：微分第一中值定理；

(4)数学实例知识：如符号函数是一个具体的函数。

对于这些基本的数学知识对象之间存在着明显的依赖层次关系。首先，概念的定义知识是数学知识描述的最基本单位，每个概念有相应的断言。数学知识的一般表达方式为：一个新概念的引入通常是以先前定义过的概念为基础；一个定理的阐述依赖于定义过的概念；一个数学实例是与一个概念或断言相关的；定理的证明知识是给出相应的定理的证明过程、方法等。

根据对数学分析知识描述对象的分类，借鉴 OMDoc 的思想，我们建立了数学分析本体体系，包括数学概念、数学断言、数学例子、数学分支等：

(1)数学概念：数学知识描述的主要对象，包括概念的基本知识(概念名称、简称等)，各种定义方式(形式定义、非形式定义等)，概念之间的关系表示等；

(2)数学断言：以数学概念本体定义的数学概念为基础，给出概念的性质描述，概念与概念之间的关系等；

(3)数学实例：数学概念或者数学断言的实例，以适当的方式有效地继承概念或者断言的知识；

另外，还有证明过程、数学方法、数学家、数学著作本体等。

```
defcategory 数学概念
{
      关系: 简称
            : 类型 字符串
            : 注释 "如: 函数在一点的极限简称极限, 左闭右开区间简称左闭区间"
      关系: 又称
            : 类型 字符串
            : 注释 "如: 单调上升数列又称单调增加数列"
      属性: 类型
            : 类型 字符串
      属性: 提出人
            : 类型 字符串
            : 注释 "不是任意字符串, 应该是对应的数学家的名字"
      属性: 提出时间
            : 类型 时间
      属性: 适应范围
            : 类型 字符串
            : 注释 "不是任意字符串, 应该是数学分支"
      关系: 是一个
            : 类型 字符串
      属性: 参数
            : 类型 字符串
            : 注释 "数学概念中的一个重要的属性; 如: 规则 和 集合是概念函数的两个参数."
      属性: 返回参数
            : 类型 字符串
            : 侧面 类型
            : 注释 "用于区分是否为函数型谓词的标志, 若是一个函数型谓词, 需要通过该参数返回函数值, 若是一个简
              单的谓词, 不需要该属性. 如: 函数在一点的值( )是一个函数型谓词; 奇偶性( )是一个非函数型谓词."
      属性: 非形式定义
            : 类型 字符串
            : 注释 "给出概念的自然语言描述"
      属性: 等价定义
            : 类型 字符串
            : 注释 "与非形式定义等价的概念定义"
      属性: 形式定义
            : 类型 字符串
            : 注释 "概念的基于一阶逻辑的谓词形式, 用于底层的知识推理"
}
```

图 1　数学概念本体的定义

图 1 给出了概念本体的内容，包括了最基本和最重要的部分属性和关系。下面对部分属性和关系加以说明：

◆名称：给出具体概念的名字，这是区别于其他概念的唯一 ID。

◆又称：概念的别名，通过引入"又称"关系可以解决同物异名带来的不一致性。

◆提出人：第一次提出概念的数学家，它的类型不是任意的字符串，必须是某一数学家的名字，为了保证知识的完备性和一致性，应该确保这个数学家知识框架在获取到的知识库中。

◆参数：一组带有类型说明的变量。概念框架中的参数是概念知识框架中的重要属性。

◆返回参数：对于函数的知识框架，除了具有输入参数外，还有相应的"返回参数"属性以及相应的类型说明。

◆使用范围：指明概念所属的具体学科分支，比如，"函数"的使用范围是"数学分析"。可以利用框架的属性"使用范围"来组织知识框架，管理知识。

◆非形式定义：以自然语言的形式给出概念的描述，对应于 OMDoc 中的 CMP 属性，这部分知识是面向用户的，主要用于知识查询、知识问答和知识教学。

◆形式定义：基于一阶逻辑给出概念的严格的形式化定义，对应于 OMDoc 中的 FMP 属性，这部分知识面向机器，用于底层的知识推理和定理证明。形式定义是概念知识表示中的核心部分，通过形式定义可以建立概念之间的关系（IS-A 关系、知识继承关系等）。

数学分析本体的构建原则和 NKI 其他学科本体的构建是一致的，但是由于数学本身的特点（如数学知识之间有着很强的依赖层次关系），我们引入了“参数”属性作为概念关系的一种特殊形式并提供谓词变量的传入形式。

概念“函数”的参数有两个：规则(f)和集合(X)。规则(f)和集合(X)是组成“函数”的两个概念，同时又是概念框架“函数”的两个谓词变量参数，用“函数(f,X)”来判断规则 f 对于集合 X 是否是一个函数。

“返回参数”主要用于数学中的函数型谓词的表示，其作用是用来获得函数值的返回值。

3.2 数学分析知识的表示方法

鉴于数学知识的多用途性和应用广泛性，我们提出了分层的数学分析知识表示方法，采用本体、框架、逻辑和参数类型相结合的方法：

(1)本体：如前面所述，我们定义了包含数学概念、断言、例子等在内的数学领域本体。

(2)框架：在数学领域本体的指导下，对每个数学对象给出框架形式的知识表示。

(3)逻辑：在框架内部基于一阶逻辑给出数学对象的严格的形式化定义，建立以”符号”为中心的 NKIMath 谓词表示系统，每个符号都有自己明确的定义和自己满足的公理。这部分知识主要用于将来的知识推理和定理证明。

(4)参数类型：在知识框架的内部，参数的类型可以是复合类型(“类型”型概念)。我们将数学概念分为“类型”型概念和“非类型”型概念。“类型”型概念就是可以作为变量类型的概念，比如，“函数”，“集合”，“积分”等，这些概念可以作为其他知识框架的属性“参数”的类型。而“非类型”型概念是不可以作为变量类型的概念，比如，“函数的复合”，“集合包含”等，这些概念不能作为参数的类型出现在其他的知识框架中。

我们设想能够通过扩充四层的数学知识表示方法来满足将来数学分析知识的更新和发展。

4 数学分析知识的获取

基于面向多用途的数学分析知识表示方法，在数学分析知识获取中我们采用基于概念的知识获取方法 COKAM。所有的概念并不是相互独立的，一个新的概念往往通过一些已经定义或者已经获取了的概念来定义。

设 C 表示一个数学概念，所有的数学概念集合用 CS 表示。下面给出一些定义和原理：

定义 1 如果概念 C 的定义依赖于概念 C'，则说 C 是 *CDDO*(Concept-Definition-Dependent-On)，即概念定义依赖于 C'，记做 $R(C,C')$。

定义 2 任意 C 属于 CS，集合 $\Omega(C)=\{C'\mid C'$ 属于 $CS \wedge R(C,C')\}$ 叫做概念 C 的 *CDDO* 集。

定义 3 任意 C 属于 CS，如果 $\Omega(C)=\Phi$，则 C 叫做原子概念。所有的原子集合用 *TCS* 表示，显然有 $TCS \subseteq CS$。

为了数学概念知识的表达，原子概念集合作为形式化其他概念的基础并且它们不需要被其他概念形式化。

定义 4 一个 *CDDO* 关系表是一个二元组 $G=\langle CS;R\rangle$，这里 CS 是所有数学概念的集合，R 是 CS 上的 *CDDO* 关系集合。在 *CDDO* 关系表中有两类概念：一类是长方形节点表示的非原子概念；另一类是椭圆节点表示的原子概念。如果 $R(C,C')$ 属于 R，则有一条从 C 到 C' 的直接弧。

图 2 表示了交换群的 *CDDO* 关系表。椭圆结点表示原子概念，长方形节点表示非原子概念，直接弧表示概念之间的 *CDDO* 关系。

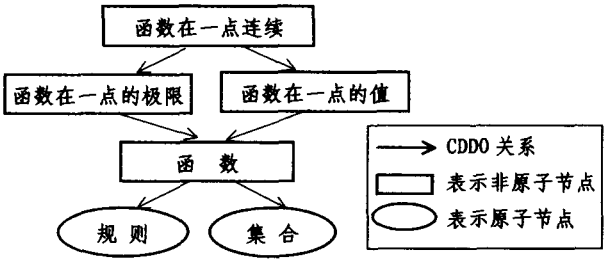

图 2 概念“函数在一点的值”的 *CDDO* 关系图

定义 5 任意 C 属于 CS，如果 C 属于 *TCS*，那么称 C 定义完整。

定义 6 一个概念集合 $CS_1 \subseteq CS$ 称为定义完整，如果任意 C 属于 CS_1，C 定义完整。

基于上面的定义，我们为数学知识获取提供 COKAM (concept-oriented knowledge acquisition method)方法，下面是具体的算法步骤：

步骤 1 如果 $\Omega(C)=\Phi$，则称一个给定概念 C 的获取完成了，否则，转到步骤 2；

步骤 2 任意 C' 属于 $\Omega(C)$，若 C' 是一个原子，则 $\Omega(C) \leftarrow \Omega(C)-\{C'\}$ 并转到步骤 1，否则继续获取并形式化概念 C'，然后 $(\Omega(C) \leftarrow \Omega(C)-\{C'\}) \cup \Omega(C')$ 并转到步骤 1；

如上图所示，为了获取“函数在一点连续”这个概念，必须首先获得“函数在一点的极限”和“函数在一点的值”这两个概念，否则概念“函数在一点连续”的获取工程就还没有完成。由于“函数在一点连续”的概念不是原子概念，获取过程应当继续下去直到所有的 *CDDO* 概念都是原子概念为止。

从上面的获取方法中可以得知，有一些原子概念是不能够形式化的。我们指出两类原子类型：数和集合，其他所有的数学分析知识最后都可以形式化到这两个原子概念为止，整个数学分析知识的形式定义都建立在这两类概念的基础上，这样，我们就给出了所有概念的严格的形式化定义。

结合上面的数学分析知识表示方法，采用面向概念的知识获取方法，图 3 给出一些数学分析领域中获取到的例子。

“闭区间”参数 G 的类型是二元组（二元组是前面提到的“类型”型概念），返回参数“R”是所有属于“闭区间”的元素组成的集合，“非形式定义”以自然语言的形式给出“闭区间”的定义描述，“形式定义”以一阶逻辑的形式给出概念“闭区间”的严格定义。“超函数”这个概念是对数学分析中一般函数的抽象，它的参数 F 的类型是二元组，“实变量的实值函数”指的就是数学分析中一般的函数概念，其参数类型是“超函数”，以“超函数”类型为基础在形式定义部分增加约束条件就得到“实变量的实值函数”严格的逻辑表示，概念“实变量的实值函数”的属性“适用范围”指明概念所属的具体学科分支。

显然，根据数学知识获取方法 COKAM 的思想，要获取“实变量的实值函数”概念，首先应该获取“超函数”概念，由于“超函数”还不是原子概念，获取过程应当继续下去直到所有

的CDDO概念都是原子概念为止。

```
defframe 闭区间：数学概念本体
{
    适用范围：数学分析
    参数：G
        ：类型 二元组(a，b)
    返回参数：R
        ：类型 集合
    非形式定义：设a，b是有限实数，a<b，满足不等式a≤x≤b的x的全体组成一个闭区间，记为[a，b]。
    形式定义：有限实数(a)∧有限实数(b)∧小于(a，b)∧任意(x：实数)(小于等于(a，x)∧小于等于(x，b)∧属于(x，R))
}
defframe 超函数：数学概念本体
{
    适用范围：数学分析
    又称：映射
    参数：F
        ：类型        二元组(f，X)
    非形式定义：给定一个规则f={(x，y)}，给定两个集合X和Y，由序对(x，y)(x∈X，y∈Y)所成的集合f={(x，y)}称为函数或
        者映射，如果满足条件：如果对(x'，y')，(x"，y")∈f，则当y'≠y"时有x'≠x"。
    形式定义：规则(f)∧集合(X)∧任意(x：元素)(属于(x，X)∧唯一存在(y：元素)属于(序对(x，y)，f))
}
defframe 实变量的实值函数：数学概念本体
{
    适用范围：数学分析
    简称：函数
    参数：F
        ：类型        超函数((f，X))
    非形式定义：如果对于某个范围X内的每一个实数x，可以按照确定的规则f，得到实数集合内唯一的一个实数y和这个x对
        应，我们就称f是X上的函数，它在x的数值(称为函数值)是y，记为f(x)，即y=f(x)。
    形式定义：子集(X，实数集)∧子集(超函数的值域(F)，实数集)
}
defframe 复变量的复值函数：数学概念本体
{
    适用范围：复变函数
    参数：F
        ：类型 超函数((f，X))
    非形式定义：如果对于某个范围X内的每一个复数x，可以按照确定的规则f，得到复数集合内唯一的一个复数y和这个x对
        应，我们就称f是X上的函数，它在x的数值(称为函数值)是y，记为f(x)，即y=f(x)。
    形式定义：子集(X，复数集)∧子集(超函数的值域(F)，复数集)
}
```

图3 数学分析领域知识获取实例

5 数学分析知识获取过程中的错误分析

知识工程师在获取知识的过程中，由于自身的或者知识源等各种原因，会出现各种错误，因此，总结出这些错误对以后的知识获取有着极大的意义。

5.1 有关知识正确性的错误分析

5.1.1 *知识源本身的错误* 知识工程师所使用的知识资料源本身可能有错误，这就不可避免导致获取的知识是错误的，所以，选择正确的知识源是进行知识获取的第一步。

5.1.2 *简单的拷贝错误* 这种错误常常发生在给出已知概念或者断言等的非形式定义而需要获取未知的相似知识的非形式定义时，比如获取了“单调增加的实变量的实值函数”这个概念，现在要获取“严格单调增加的实变量的实值函数”这个概念，显然，两个概念的非形式定义部分非常相似，如果在获取的过程中不认真而简单地把这部分知识拷贝过来不加修改就会发生错误，这种在非形式定义中出现的人为的错误是很难发现的。因此，知识工程师在获取这部分知识时应该非常认真、谨慎。

5.1.3 *概念分类错误* 有些概念和例子很难判断是概念本体的实例还是例子本体的实例，比如概念“符号函数”，如果不慎导致分类错误，或者，直接把应该属于概念本体的实例判断成断言本体的实例(还有其他类似情况)，就会影响到知识库的正确性、完备性和一致性。

5.1.4 *混淆数学分析分类体系本体之间的属性和关系* 在概念本体中不会出现“关于”关系，在断言本体中没有“是一个”关系，如果在概念本体实例中出现了属于断言本体的属性和关系，或者在断言本体实例中出现了属于概念本体的属性和关系，都是错误的。这类错误可以通过程序得到有效的解决。

5.1.5 *形式定义中出现的错误* 由于我们在本体、框架、逻辑和参数类型相结合的思想指导下，采用分层的知识表示方法，知识的形式定义用一阶逻辑的形式给出，这是知识推理、定理证明和建立概念之间各种关系的基础，所以“形式定义”在整个数学知识库中占有重要的地位。由于知识工程师自身的知识结构不完善等各种原因，很容易在“形式定义”中出现错误，比如将断言“微分第一中值定理”的形式化定义部分写成：“…∧闭区间(a,b)∧…”，这样，我们就曲解了“微分第一中值定理”的意思，这种人为的错误在知识获取的过程中是很难发现的，但是对整个知识库的影响却非常之大，这也是比较容易犯的错误之一。

5.2 有关知识一致性的错误分析

5.2.1 *引用已有知识库中的概念时发生错误* 由于我们采用面向概念的数学分析知识获取方法，一个新知识的获取依赖于知识库中已获取到的知识，所以，这种引用知识库中的知识来获取新知识的事情是最常发生的。然而，如果缺乏对知识库的充分了解，很容易就会犯错。

(1)引用名称错误：比如知识库中关于加法运算用“加()”来表示，知识工程师在运用它来获取新的概念时很错用成“加法()”。

(2)参数个数错误：比如在运用“函数在一点的值”来定义“函数在一点连续”时，很容易将“函数在一点的值$((f,X),x)$”表示成“函数在一点的值(f,X,x)”，参数个数不匹配。

(3)参数类型错误：是指使用了错误的参数类型，比如：在用到函数知识时，“函数(C)”要求C是一个函数，如果传入的

是参数类型不是一个函数，就会发生错误。

(4)缺少返回参数错误：比如函数“加()”知识框架需要一个返回参数，表示两个被加数的和，如果缺少这个参数，引用函数“加()”知识时就会发生错误。

5.2.2 *形式定义中发生的不一致性错误* 形式定义在数学知识库中占据核心地位，所以，我们特别强调形式定义的正确性和不一致性。

(1)未识别的符号：出现在形式定义中的各种符号(包括保留字、标识符、运算符等等)都应该获取到，和谓词符号系统中的符号保持一致。

(2)文法归约错误：括号不匹配、不符合文法定义、遗漏量词等等。

(3)谓词类型错误：在数学知识库中每个谓词都是经过严格定义的，在形式定义中使用的谓词必须符合它自己的定义，否则，会引起知识的不一致性，包括：3.1)未定义的谓词：在知识库中没有与谓词相对应的知识的严格定义；3.2)谓词类型不一致；3.3)谓词的参数个数不一致。

(4)函数类型错误：在数学知识库中每个函数都是经过严格定义的，在形式定义中使用的函数必须符合它自己的定义，否则，会引起知识的不一致性，包括：3.1)未定义的函数：在知识库中没有与函数相对应的知识的严格定义；3.2)函数类型不一致；3.3)函数的参数个数不一致；3.4)返回值的类型和被引用函数的参数类型不一致。

(5)非形式定义和形式定义的不一致性：概念“严格单调增加函数”的非形式定义为：“如果对于某区间 X 内的任何两点 $x1<x2$，总成立着 $f(x1)<f(x2)$，则称函数 $y=f(x)$ 在区间 X 内为严格单调增加，有时也称严格单调上升。”而其形式定义如果写成如下形式：“任意(x1,x2:实数)(属于(x1,X)∧属于(x2,X)∧小于(实变量的实值函数在一点的值(G,x1)，实变量的实值函数在一点的值(G,x2)))”，很明显，概念的非形式定义和形式定义是不一致的，非形式定义描述了概念“严格单调增加函数”，而形式定义却定义了概念“单调增加函数”。

(6)等等

5.2.3 *属性和关系的不一致性* 上面的正确性分析中提到了容易混淆数学分析分类体系本体之间的属性和关系，它们的一不致性也是常犯的错误：

(1)数学概念实例中的提出人应该是一个科学家，而不是任意的字符串，为保证知识的一致性，必须有数学家本体的某个实例，在属性“提出”中包括前面的数学概念实例。

(2)数学概念实例中的提出时间必须是时间的表示格式，而不应该是任意的字符串。

(3)断言实例中的关于属性中出现的符号应该是经过严格定义过的概念。

(4)知识继承时发生的不一致错误：设概念 A 继承概念 B，则 A 可以继承 B 中的所有关系和属性，在 A 中重复定义从 B 中继承过来的属性和关系就导致了继承的内容冗余。

(5)等等

5.2.4 *相同知识框架的重复获取* 在知识获取的过程中，由于没有充分利用现有的已经获取的知识库，使得知识工程师单独重新获取了原本在知识库中的已有的知识框架。这样不仅仅造成了重复劳动，给知识库的正确性、一致性等检查增加了难度，容易带来使用上的不一致性，并且也不便于知识的维护、管理和使用。这种错误可以通过程序检查出来。

5.2.5 *两个知识工程师获取的知识库的一致性* 如果两个知识工程师独立获取的知识库有交叉的部分，如何保证两个知识库的一致性是至关重要的。

(1)命名空间的不一致性：对于同一个概念、断言取了不同的名称或者对不同的概念、断言取了相同的名称。比如，右闭区间和左开区间表示相同的概念知识。通过在数学概念本体中增加“又称”属性来解决这种不一致性。

(2)两个相同的知识框架，形式定义不一致：对于同一个概念，不同的知识工程师均可以得出不同的正确的形式化定义。比如，开区间的形式定义可以表示为“有限实数(a)∧有限实数(b)∧小于(a,b)∧任意(x:实数)(小于(a,x)∧小于(x,b)∧属于(x,R))”，也可以表示成“实数(a)∧小于(a,正无穷大)∧实数(b)∧小于(b,正无穷大)∧小于(a,b)∧任意(x:实数)(小于(a,x)∧小于(x,b)∧属于(x,R))”，这涉及到数学知识表示的粒度问题，如何保证它们的形式定义之间的等价性是非常重要的。

(3)参数个数和类型的不一致性：情况与 5.2.1 引用已有知识库中的概念时发生错误大致相似。

5.2.6 *基于归结原理的知识库一致性检查* 知识库的一致性检查是数学知识获取中的一个重要问题。在我们的工作中，采用的归结推理的方法。其基本算法如下：将待检查的知识库转化为子句集，反复对子句应用归结推理规则，直到没有更多的归结项可以被添加，或者产生一个空子句。如果出现空子句，则表示“一致”。归结反驳具有完备性，表现为如果知识库一致，则归结反驳过程将推导出空子句。

5.3 命名问题

我们所获取的知识库最终是面向用户的，用户可以用数学知识描述语言(MKDL)通过用户界面提问，所以要求知识库中的知识的名称是易于被人们接受的。比如，用户要查找“群的第一同态基本定理”知识，而在我们的知识库里这条知识是用“群的同态基本定理”来表示的，那么，用户就得不到他想要的知识。

知识工程师利用已有的知识来获取未知的知识时也会碰到同样的问题。比如，我们在知识库中用“周期函数的周期”定义来定义数学分析中周期函数的周期这个概念，而我们在自然语言中通常就用“周期”这个词来说明这个概念，所以引用这个概念时就会发生错误。

总结和进一步的工作 数学知识的获取和分析是一个非常庞大和复杂的任务，但是数学知识获取的重要性是不言而喻的。本文探讨了面向多用途的数学分析知识表示方法，采用基于面向概念的获取方法，以数学百科全书和复旦大学的数学分析教材为依据，已经获取了近 1000 个知识框架，涵盖了大学水平的数学分析教学的内容，有效地验证了本文方法的可行性和正确性。总结和分析了出现在知识获取过程中的错误，这对以后的知识获取有着极大的意义和指导作用。

本文主要讨论的是数学分析知识的获取和分析，我们认为这些方法也可以应用于其他数学分支。我们将在其他具体数学分支上展开类似的研究。目前，我们也基本完成了大学水平的代数知识获取。

参 考 文 献

1 张景中. 机器证明的回顾与展望. 大自然探索，1997，16(1)：6～9

2 吴文俊. 数学的机械化. 百科知识，1980(3)

3 赵子都. 定理机器证明. 自然辩证法研究，1994，10(5)：46～50

4 陆汝钤. 人工智能(下册). 北京：科学出版社，2000

5 Wiedijk F. Comparing Mathematical Provers. In: MKM2003, LNCS 2594, 2003. 188～202

(下转第 138 页)

策表个数。

结论 本文分析了 Rough 集理论中正区域和边界域的一些性质，认为在不一致决策表的约简过程中，既应该考虑正区域对约简过程影响，又应该考虑边界域对约简过程的影响。这类似于机器学习中的概念学习，概念不仅要覆盖正例，而且不能覆盖反例。在此基础上，给出了一种新的属性重要性定义，这种定义既含有正区域信息又含有边界域信息。

最后，以新的属性重要性为启发信息，给出了一个新的算法，该算法的时间复杂度与基于正区域的约简算法时间复杂度相同。通过例子和仿真实验说明，新算法在搜索最优或次优约简上优于传统的算法。

参 考 文 献

1 Pawlak Z. Rough sets approach to multi-attribute decision analysis. European Journal of Operational Research, 1994, 72:443～459

2 刘少辉，盛秋戬，吴斌，史忠植，胡斐. Rough 集高效算法研究. 计算机学报，2003，26(5)：524～529

3 苗夺谦，胡桂荣. 知识约简的一种启发式算法. 计算机研究与发展，1999，36(6)：681～684

4 王国胤，于洪，杨大春. 基于条件信息熵的决策表约简. 计算机学报，2002，25(7)：759～766

5 Wang Jue, Wang Ju. Reduction Algorithms Based on Discernibility Matrix: The ordered Attributes Method. Journal Computer Science & Technology, 2001, 16(6): 489～504

6 张文修，等. 粗糙集理论与方法. 北京：科学出版社，2001

(上接第 123 页)

6 Smirnova E S, So C M, Watt S M. An Architecture for Distributed Mathematical Web Services. In: MKM2004, LNCS 3119, 2004. 363～377

7 Franke A, Hess S M, Jung C G, et al. Agent-Oriented Integration of Distributed Mathematical Services. Journal of Universal Computer Science, 1999, 5: 156～187

8 Kohlhase M. Towards an Knowledge-Centered Infrastructure for Web-Based Mathematics. Availabe at: http://www. cs. cmu. edu/～kohlhase, February 14, 2001

9 Homann K, Calmet J. Combining Theorem Proving and Symbolic Mathematical Computing, Integrating Symbolic Mathematical Computation and Artificial Intelligence. In: Proc. of 2nd Intl. Conf. on Artificial Intelligence and Symbolic Mathematical Computing (AISMC-2), Karlsruhe, Springer, LNCS 958, 1995. 18～29

10 Franke A, Kohlhase M. System Description: MATHWEB, an Agent-Based Communication Layer for Distributed Automated Theorem Proving. FB Informatik, University des Saarlandes. In: Harald Ganzinger, ed. Automated Dedution | CADE－16, LNAI 1632, Springer Verlag, 1999. 217～221

11 Noy N F, Musen M A. SMART: Automated Support for Ontology Merging and Alignment. In: Proc. of the Twelfth Workshop on Knowledge Acquisition, Modeling and Management, Banff, Canada, July 1999

12 Sure Y, Staab S, Angele J, et al. OntoEdit: Guiding Ontology Development by Methodology and Inferencing. In: Prestigious, Applications of Intelligence System (PAIS), ECAI 2002, Lyon, France, 2002

13 陆汝钤. 世纪之交的知识工程与知识科学. 北京：清华大学出版社，2001

14 Carlisle D, Ion P, Miner R, et al. Mathematical Markup Language (MathML) version 2. 0. W3C Recommendation, World Wide Web Consortium, 2001. Available at: http://www. w3. org/TR/MathML2

15 Extensible Markup Language (XML). Available at http://www. w3. org/XML/

16 Bray T, Paoli J, Sperberg-McQueen C M. Extensible Markup Language (XML). W3C Recommendation TRXML, World Wide Web Consortium, December 1997. Available at: http://www. w3. org/TR/PR-xml. html

17 Raggett D, Hors A L, Jacobs I. HTML 4. 0 Specification. W3C Recommendation REC-html40, World Wide Web Consortium, April 1998. Available at: http://www. w3. org/TR/PR-xml. html

18 Kohlhase M. Creating OMDoc Representations from LATEX. Internet Draft. Available at: http://www. mathweb. org/omdoc, 2000

19 Caprotti O, Cohen A M. Draft of the OpenMath Standard. The Open Math Society. Available at: http://www. nag. co. uk/projects/OpenMath/. mstd/, 1998

20 Kohlhase M. OMDoc: An Open Markup Format for Mathematical Documents (Version 1. 1). Availabe at: http://www. cs. cmu. edu/～kohlhase, 2001

21 Kohlhase M. OMDoc: An Open Markup Format for Mathematical Documents: [Seki Report SR-00-02]. Fachbereich Informatik, Universit"eat des Saarlandes, 2000. Available at: http://www. mathweb. org/omdoc

22 Kohlhase M. OMDoc: An Infrastructure for Openmath Content Dictionary Information. Bulletin of the ACM Special Interest Group on Symbolic and Automated Mathematics (SIGSAM), 2000, 34(2): 43～48

23 Kohlhase M, Franke A. MBase: Representating Knowledge and Context for the Integration of Mathematical Software System. Availabe at: http://www. cs. cmu. edu/～kohlhase

24 Asperti A, Padovani L, Coen C S, et al. Towards a Library of Formal Mathematics. Short presentation at TPHOLS2000. Available at: http://helm. cs. unibo. it

25 中国数学机械化研究中心. Available at: http://www. mmrc. iss. ac. cn/mmrc-il. html

26 Cao Cungen. Technology Focus of 21st Century. Computer World, 1998, D1-D3 (in Chinese)

27 Cao Cungen. Medical Knowledge Acquisition from the Electronic Encyclopedia of China. Lecture Notes in Computer Science, 2001, 2101: 268～271

28 Cao Cungen, et al. Progress in the Development of National Knowledge Infrastructure. Journal of Computer Science & Technology, 2002, 17(5): 1～16

29 Si Jinxin, Cao Cungen, et al. An Environment for Multi-Domain Ontology Development and Knowledge Acquisition. EDCIS 2002, Beijing, LNCS2480, 2002. 104～116

30 Zeng Qingtian, Cao Cungen. Ontology-base Mathematical Knowledge Multipurpose Representation, Acquisition, Analysis and Management: [Technical Report 2002]. Institute of Computing Technology, Chinese Academy of Sciences, 2002

31 曾庆田，曹存根，眭跃飞，等. 基于本体的数学知识获取与知识继承机制研究. 微电子学与计算机，2003，20(9)：19～27

32 陈传璋，朱学炎，金福临，欧阳光中. 数学分析(第二版). 高等教育出版社，2003

33 数学百科全书. 中国大百科全书出版社，1998