

符号与代数计算的新阶段

——“符号计算系统的设计与实现”

(首次)国际研讨会参加记

徐家福 (南京大学)

我受国家教委派遣参加了1990年4月10日至4月12日在意大利卡布里召开的“符号计算系统的设计与实现”国际研讨会,今汇报如下。

一、一般情况

1. 这是符号计算领域内的首次国际会议。其目的是,提供这一领域的最新面貌,促进学术界、工业界、以及用户团体之间的学术交流。主题围绕符号计算系统的设计与实现的创新的方法与技术,例如,自动推理、几何模型构造与计算、自动程序设计等等。

2. 会议报告并讨论了二十五篇论文和十篇短文。内容涉及以下几个方面。(1)符号与代数计算:系统设计;(2)符号与代数

计算:实现方法与技术;(3)理论;(4)自动推理;(5)软件环境与语言;(6)软件环境与用户接口界面。

3. 此次会议国内只有我一人参加。会议主席意大利“La Sapienza”罗马大学信息学与系统学系Alfonso Miola教授和其它国家的一些代表态度都比较友好,和他们交流了情况,讨论了一些技术问题,特别是,和自动程序设计有关的问题。这是一个小型的研讨会,来自十多个国家约一百名代表参加了这次会议,规模不大,但文章质量高,讨论比较深入。今后有选择地参加一些这样的会议,“性能价格比”较高。

二、符号与代数计算:系统设计

1. 英国Bath大学J.H. Davenport的

事。因为工程化不是孤立的,它涉及到形式化,自动化等问题。国际上虽然发展迅速,但二十多年来却未见重大突破,目前仍然处于渐变阶段。形式化问题颇多,远未普及;自动化出现了一些辅助工具,高层次的自动化涉及到智能化与自然化等问题,国内外均仍处于实验、探索阶段。

3. 坚持远近结合,理论联系实际

发展我国软件工程,一方面,国外经验可资借鉴;另一方面,还必须根据我国国情,进行自主研究与开发,相信国内的力量,我们无理由妄自尊大,也不应该妄自菲

薄。从我们多次进行国际学术交往来看,一般说来,国外学术界与工业界人士是相互尊重、相互学习、取长补短的。正由于如此,人家发展才比较快。重要的是,学术界也好,工业界也好,既要有长远考虑,又要千里之行始于足下,要远近结合,注意理论联系实际。不重视实际工作当然不对,但对理论工作不予以足够的重视,将可能铸成大错,吃大亏,何况好的实际工作总是有理论基础的,好的理论工作也是有其实际背景的。

限于水平,欠妥之处,请批评指正。

“计算机代数系统设计中的当前问题”一文谈到, 计算机代数系统已有二十多年的历史, 但是, 对于“理想系统”的面貌尚无一般的认识。在移植性、功能性、以及功效性之间有多种权衡方案。作者讨论了这些问题, 如存储管理、引用计数、废区收集、语言、数值软件的接口界面、正文处理的接口界面、以及图形处理的接口界面等等。

2. 澳大利亚悉尼大学Greg Butler等人的“近世代数语言Cayley的设计”一文指出, 近世代数领域中已有经验形成了Cayley的设计, 适应了用户的需要。用户需求包括和公用数学记法的一般性; 诸如集合、序列、映射、代数结构、以及元素等合适的数据类型; 功效性; 可扩充性; 对已知算法的建入函数和过程的能力; 以及对代数结构公用例子的取接等等。讨论了这些方面对Cayley用户语言的影响。

3. 意大利Genova大学Alessandro Giovini等人的“CoCoA: 一个对用户友善的交换代数系统”报导了一个进行交换代数计算的小型专用系统CoCoA。这是一个在Macintosh个人计算机上开发的系统, 含有20000行Pascal代码。程序设计风格和数据结构的实现使得系统是开放的, 易于修改和改进, 便于对甚高级的优化技术进行维护, 强调严格数学意义下的算法优化。系统由内核和库组成, 内核包括两部分, 即用户接口内核和代数内核; 库直接建在代数内核的顶部。

4. 法国CEDEX信息学与自动学国家研究所A. Galligo等人的“符号与代数计算系统SISYPHE的设计”一文报导了INRIA和NICE大学自1988年以来合作开发的一个通用计算机代数新系统SISYPHE, 该系统可用作一个数学计算器, 利用程序设计语言可添加新的过程和包块。SISYPHE用Le-Lisp书写, 命令语言的语法和用户函数名和Macsyma系统类似。系统由一组编译过的模块组成, 包括一个预编译程序和一个编译程序。数学对象容许带相关转换函数的有型内部表

示。已经实现的新特点是: (1) 推广了RAT表示; (2) 推广了“简化”概念; (3) 往机器码中提供了分别编译; (4) 构造了一个简单的模块语言, 它和命令语言、Lisp都是不相交的; (5) 创建了一些和别的计算机代数系统通信的设施; (6) 为工作站开发了一个高级图形接口。

5. 英国Bath大学J.H. Davenport等人的“Scratchpad系统的代数观点: 基本交换代数”。作者认为, 虽然计算机代数系统处理多项式和整系数的有理函数已有多年历史, 但处理交换代数中更为一般的构造却是新近才出现的。该文解释了一个系统如何解决这一问题, 有必要引进哪些类型和算符, 简言之, 如何来构造一个交换代数的计算理论, 这一理论和传统的非构造性理论不同, 它和Seidenborg及其学派的理论也不同, 他们未曾讨论实际的功效问题。

三、符号与代数计算: 实现方法与技术

1. 法国INRIA Gerard Huet的“构造性数学的计算辅助环境的设计问题”报导了正在INRIA开发的、作为“证明协助软件”基础的“构造演算”及其当前的CAML实现。当前的系统包括一个证明检查程序, 构造性引擎, 一个针对数学理论的高级陈述性语言, 一个定理证明程序(用以在数学家指导下交互地综合证明, 并可根据用户自定义的策略扩充), 以及一个程序提取程序。所讨论的问题与基于逻辑的知识表示系统以及强类型模块化程序设计语言的程序设计环境的设计有关。

2. 西德GmbH IBM公司Albrecht Fortenbacher的“计算机代数的高效类型推理和类型迫转”一文指出, 像Scratchpad一类的新一代计算机代数系统的特征是, 具有很丰富的类型概念, 以此构造各种数学计算领域间的关系, 然而, 为了交互地利用这些系统, 用

户不应考虑类型信息。类型推理机制决定了要调用的合适函数。所有那些容许对类型推理定义语义的已知模型,都不能表示丰富的“数学”类型结构,因而,目前的类型推理都是启发式地进行。该文对其中一个子问题,即类型迫转,定义了语义(以重写法则为基础),根据这一定义,利用图技术构造了一个高效的类型迫转算法。

3. 意大利“La Sapienza”罗马大学信息学与系统学系Carla Limongelli等人的“数学结构和方法的抽象规格说明”一文讨论了用数据和方法的甚高级的抽象规格说明来处理数学对象的方法与技术。为了在最高抽象级上的设计与实现,定义了一个均匀一致的环境,从形式规格说明语言和实现性语言的观点来展开介绍。实际上,规格说明和程序设计都看成在同一个一般概念模型下的一致动作。概念模型完全由对象式方法学来支持。特别注意到,把算法逻辑作为一种逻辑规格说明语言,Loglan作为一种革新的对象式程序设计语言。

4. 民德Karl Marx大学信息学系Uwe Petermann的“符号计算系统的程序设计风范:案例分析”一文讨论了宜于符号计算系统设计的程序设计风范。在一个案例中,对象式程序设计语言Loglan已用于实现非交换多项式代数及其非对称商域中的算法。讨论集中在古典程序设计风范(如模块嵌套)和对象式程序设计风范的结合上。

5. 美国卡内基·梅隆大学计算机科学系Jean-Philippe Vidal的“在共享存储的多重处理器上Gröbner基的计算”一文介绍了在共享存储的多重处理器上计算Gröbner基。其基本思想是,每个处理器从一组未化简的临界对中选取一元,化简与之相关的S-多项式,根据结果对基和对组进行更新。这一算法的独创性表现在各进程间要求少量的同步。给出了在含16个处理器的Encore机器上实现的细节。

四、理论

1. 美国斯坦福(Stanford)大学Carolyn Talcott的“程序和数据类型规格说明理论”一文报导了一个研究项目的前期研究成果。该项目旨在发展一个宽谱形式体系,用以不仅支持程序等价性推理,而且支持程序和数据类型规格说明,支持对计算、程序运算、程序规格说明运算等特性的推理。在前期中,已经发展了一种程序等价和类成员的双层理论。低层是程序等价及定义性的理论,其中能处理的程序原始成份包括函数作用与抽象、条件、数、配对、连续性捕捉等。高层是带有定义类的通用理解原理的类成员的理论。很多标准类构造与数据类型都可以自然地表示出来,其中包括代数抽象数据类型、表、记录、以及函数类型构造等。

2. 西德Passau大学Rolf Henuicker的“前后文归纳:一种关于行为抽象的证明原理”一文介绍了一种称作前后文归纳的归纳原理,它宜于验证抽象数据类型的行为特性,该证明原理的有用性已由若干应用所证实,如行为规格说明上行为定理的验证,行为实施的验证,以及“忘记-限制-验明”实现的验证等。

3. 法国CEDEX Jean-Pierre Jouannaud等人的“以结合性、交换性、恒等性(ACI)为模的完备性”一文指出,长期以来,对结合性、交换性、恒等性的重写就是一个悬而未决的问题,新近,Baird等人引进了约束重写概念,避免了利用恒等所固有的不终止性问题。在此基础上,作者进行了两项工作,一是以这些公理为模的完备性的一组完备的法则;二是为了证明以结合性、交换性、恒等性为模的约束重写的完备性,构造了一些合适的序。

4. 美国SUNY计算机科学系Thom W. Frühwirth的“Prolog子类型的多型检验”一文对Prolog子类型的多型检验系统给出了一个

可执行的规格说明。作者指出,其实现可归结为,把已知的简单元解释程序修改成类型元解释程序,并把“生成-并-检验”方法用于Prolog程序设计。该项研究强调的是,Prolog宜于作为定义可执行规格说明的语言,并且Prolog可以添加一个简单但又强有力的类型系统。

5. 荷兰Amsterdam数学和计算机科学中心Roland N. Bol等人的“试论包含和前后文检验的能力”一文讨论了两类循环检验机制,即包含检验与前后文检验,分析了其坚固性、相对强度和其它相关概念而言的完备性,证明了坚固性以及某些逻辑程序类的完备性。包含检验的完备性定理利用了Kruskal树形定理的一个称作Higman引理的简单情形。

6. 法国Grenoble IMAG-CNRS Jean-Claude Reynaud的“若干代数成份的比较考察:一种非独立的类型方法”一文为了比较考察若干代数成份,定义了一个基于非独立类型的构架,它是用自由生成范畴来定义的。为了保持各个成份的初始的、松散的、受限的语义不变,引进了SPEC范畴概念,后者类似于特定的有限共完备范畴。一种包括参数化技术的构造性方法用来根据已定义的基本成份定义新的成份。

五、自动推理

1. 英国爱丁堡大学人工智能系Alan Bundy的“形式方法中Proofplans的使用”一文指出,Proofplans是控制自动定理证明中出现的搜索的一项人工智能新技术。已经将该技术用于软件工程中形式方法出现的证明义务的满足问题。首先描述了用作工具的特殊的形式方法技术,随后谈到,在这一技术内部如何利用Proofplans。

2. 美国加洲SRI International Mark E. Stickel的“一个Prolog技术定理证明器:一种利用Prolog的新解释和实现”一文指出,

Prolog技术定理证明器(PTTP)是Prolog的一个扩充,它对全一阶谓词演算而言是完备的,它和Prolog的区别在于,第一,为使搜索策略完备,在合一使用中采用坚固性的出现检验,深度优先的交互式深化搜索,而不是无界深度优先搜索;第二,把模型消去归约法则添加到Prolog推理中,以使推理完备。该文描述了PTTP的一个基于Prolog的新实现。

3. 美国IBM T.J. Watson 研究中心Jean-Louis Lassez的“参数询问、线性约束,以及变量消去”一文指出,在从CLP(R)到Mathematica 一系列语言中都引进了各种形式的陈述性线性算术。为了找出约束的规范表示,处理非线性约束的延迟,为了解决连累选择语言中的卫项以及其它有关问题,需要回答新的更为强有力的一类询问,即参数线性询问。该文提供了一个简单的数学形式体系来处理这一问题。

4. 法国CEDEX Mohamed Adi等人的“AC合一历程:系统求解方法及其实现”一文介绍了一个基于求解方程组的结合交换合一的算法与用C语言的实现。还提出了评估这一算法及其实现的一些基准。

5. 法国奥尔良大学Siva Ananthraman的“反驳定理证明中的启发式准则”一文指出,晚近利用熟知的不失败的Knuth-Bendix方法对自动演绎进行了一些尝试,发现这一方法的实际功效在如下情形可能会提高,即(1)如果可用一个面向目标的策略根据当前的规则表去选取或生成“下一个”规则;(2)如果用某种方法我们可知,在证明进程中,一个规则比其它规则更为有用。另一方面,理论上说,这种方法却要求按一种归约序(基类上的全序)工作,并且只通过简化就能达到最终的证明。虽然实际上这往往不是最有效的方法,但当我们把KUB推广到公理重写时,要克服这一点却是很烦琐的。该文寻求这些问题的解决方法,首先证明了,可以去掉对基本归约序的假设;其次,构造了一种

系统化的方法来解决前两个问题。

六、软件环境与语言

1. 意大利比萨大学 Giuseppe Attarch 的“知识库中标准的需求”一文指出, 由于知识库成份往往是整个应用的一小部分, 重要的是, 这样的成份要容易和系统的其余部分进行集成。解决这一问题的一种方法是采用一系列的标准, 以利于接口, 以及利用对象式程序设计, 提供必要的胶着。该文勾画了一个标准层次, 从资源层开始, 经过网络层, 库层、直到对象式工具包层。便于理解的知识库模型在这一构架中是重要的。该文建议了应该满足的一些需求, 还容许若干专用语言同时存在, 并举出了一种解决语言间相互操作的方法。

2. 英国Bath大学M.C. Dewar等人的“实际配置中调和符号和数值计算”一文指出, 在计算机上求解数值问题, 传统上有两个途径。一是以NAG Fortran库为代表的通用库; 二是以REDUCE为代表的途径。二者各有短长。作者认为, 结合这两种方法, 构造一个具有友善用户接口的计算机代数系统将是有益的。作者具体构造了一个称作IR-ENA的系统。

3. 法国南巴黎大学 M. Bidoit等人的“ASSPEGIQUE数据库的设计与规格说明”一文考察了代数规格说明环境 ASSPEGIQUE的数据库的设计, 并对这个数据库的管理系统提供了一个规格说明。为了利用代数规格说明, 重要的是要有一个规格说明语言 and 一组工具, 或者, 更好一些, 要有一个集成设计、验证、使用规格说明的各种工具的集成环境。在ASSPEGIQUE数据库的设计中, 严格区分了“静态”方面和“动态”方面。文中介绍了所用的规格说明语言PLUSS, 讨论了模块性的处理, 并给出了一个静态数据库管理系统的规格说明。

4. 西德Dortmund大学Michael Hanus

的“一种具多态类型的函数式与逻辑式语言”一文介绍了一种具有精确语义定义的有类型的语言, 这种语言集成了函数式与逻辑式程序设计风格。为了在编译时刻查出程序设计错误, 语言具有多态类型系统, 这一系统限制了函数与谓词的使用, 并确保一个函数或谓词在运行时刻只用合适的变元调用一次。和逻辑程序设计的很多别的类型系统相比, 这一类型系统具有模型论语义 (类型是解释域的子集), 并容许使用典型的逻辑程序设计技术。例如, 可以添加特定的子句, 并应用高阶程序设计技术。

七、软件环境与用户接口界面

1. 意大利 Olivetti 系统与网络公司的 Vin-Ceuzo Russi等人的“面向自动化的软件开发风范的对象式可执行的图形规格说明”一文介绍了一个基于高阶Petri网(称作PROT网)的对象式CASE系统(称作PROTOB系统)。这一系统由一组工具组成, 以支持面向自动化的软件生存期风范中对象式规格说明, 符号计算, 代码生成。其主要应用领域是各类分布式系统, 如实时嵌入系统, 通信协议, 以及生产控制系统。

2. 法国INRIA Norbert Kajler的“计算机代数系统(CAS)图形用户界面的构造”一文指出, 高质量的用户界面是计算机代数系统的要素之一。首先, 它使得学生用户与临时用户更容易利用系统的主要功能, 其次, 它使得专家用户更便于取接系统的全部功能, 并便于程序设计。在所有情形, 用户界面都能显著改善视觉, 在公式编辑中还能提供很多好处。该文的目的是简要研讨了CAS用户界面的需要。此外, 这一工作代表任何能行实现的重要的居先一步。

3. 美国Kent州立大学 Paul S. Wang的“和系统无关的数学函数图形包”一文报导了一个图形系统 SIG, 它用于显示符号系统中用数学公式定义的曲线和曲面, 该系统由

协议工程概论与进展^{*)}

肖军模 徐越彦 (南京通信工程学院)

摘要 The developmental process of the standard protocol described by the formal description techniques (FDT) and directed by the ideas of the software engineering, is known as the "protocol engineering". This is a developmental method of the protocol emerging in recent years. This paper puts emphasis on the basic ideas, history of the protocol engineering, and its main tasks in each phases. Finally, the advances for studying the protocol engineering in our institute is briefly presented.

一、什么是协议工程

在信息社会中,由于信息交换和处理的需要,建立各种类型的计算机网络(局域网、城域网和广域网)已势在必行。从目前的情况看来各厂家所提供的网络通常只包括下三层或下四层协议,运输层以上各层协议则需由用户自己开发。因此不论厂家还是用户都存在通信协议的开发问题。

协议开发的最终产品是可以在不同环境下运行的协议目标代码。早期的开发方法主

要是依据协议的非形式描述用手工方式实现的。由于对非形式描述的协议可能存在不同的理解,很难保证协议产品的正确性及其与标准协议之间的一致性。

近几年来,由于协议形式描述技术的日趋成熟,人们用软件工程的思想指导协议的开发,就形成了所谓的协议工程。支持协议工程的一整套软件与工具称为协议开发系统(或环境)。

协议工程通常包含以下几个阶段:

- 协议形式描述, ISO已经提供;
- 协议正确性验证;

^{*)} 本课题得到国家自然科学基金资助。

两部分组成: Xgraph和mgraph,二者作为并发进程运行。Xgraph是一个用C书写的独立图形设施,用X窗口系统工作;mgraph是和符号系统有关的部分。SIG通过X实现显示设备的独立性,通过mgraph达到可移植性。该文介绍了SIG的功能、设计与实现、以及进一步的开发计划。

4. 意大利C.N.R. CNUCE研究所G. P. Facouti等人的“图形系统的交互模型”一

文介绍一个宜于描述计算机图形系统基准模型(在ISO内部开发的)所定义的构架内部的交互式图形程序的模型。着重讨论其“陈述成份”,其中列出并讨论了来自各种不同方法的问题。随后介绍了由计算机图形基准模型所定义的系统结构。接着描述了作为逻辑设备或交互者内部一组进程的各个交互成份。

限于水平,欠妥之处,请批评指正。