

可演绎系统的研究

李磊 (中山大学岭南学院计算机系 广州510275)

J. Kouloumdjian (法国里昂INSA信息科学实验室)

koulo, J

TP311.13

摘要: 本文综述了可演绎系统目前的研究情况, 认为以DATALOG为代表的演绎数据库系统进化为实用系统是困难的。文中简介了一个基于PROLOG的可实用的可演绎系统, 还对进一步的研究内容及其原因进行了讨论。

考虑计算机的应用历史大致可分为三个阶段: 科学计算、数据处理和知识处理。每个阶段都有对应的工具系统, 它们分别是: 算法语言, 数据库系统和知识处理系统。目前, 知识处理系统还仍然处于发展阶段, 并且成为计算机科学中非常活跃的研究领域。

处理对象不同决定了处理工具特性的不同, 例如, 算法语言和关系数据库系统分别对于科技计算和数据处理是十分适宜的, 但是目前对于知识处理来说还缺乏如此适宜的工具系统。五花八门的知识处理系统和干脆用传统工具来处理知识就说明了这一点。

知识的最基本操作之一是演绎, 因此知识处理系统应具有的最基本功能之一也应当是演绎, 我们把这种具有演绎功能的系统称为可演绎系统。本文首先介绍这种系统的研究与发展, 然后介绍一个实际的可演绎系统。

一、概况

知识系统的广泛应用大大地刺激了可演绎之间的时序约束所组成。

区间解释模块 (IPM): 在网络中利用基于三角形的约束传递算法生成一个不完全解, 即对那个可能世界中的每个区间进行解释, 使得网络中任意一条弧上都有一个非空的原语关系集。这时也许会报告矛盾出现的信息。

完全解生成器 (CSG): 它的主要功能

是研究系统的研究与发展, 主要方向集中于: 演绎数据库和逻辑程序设计。

关系数据库有一个逻辑解释, 还可以定义视图(View), 因此可以认为关系数据库系统是非常初等的可演绎系统。这一点强烈地引诱着数据库领域的专家来扩充这种演绎能力, 直至成为一个实用的可演绎系统。这就是目前研究的演绎数据库, 在这一领域中人们已经取得大量结果。

可以认为演绎数据库领域最基本的研究内容是对递归查询的研究。历届重要的关于数据库方面的国际会议上都有大量的论文讨论这一问题, 例如: 1990年8月16日的VLDB上发表的九篇关于演绎数据库的论文中, 有八篇是讨论递归查询的。这一状况说明了此问题的重要性, 同时也说明了这个问题还没有很好地获得解决, 虽然人们已经提出各

是根据不完全解生成网络的完全解 (可能有多个), 或者测试出整体的不相容性。

ATMS: 基于假设的真值维护系统。

· 本文所描述的时序推理系统 TRS 正在 VAX II/75 机上进行模拟试验, 效果良好, 预计能达到预定的设计目标。(参考文献 12 篇略)

种方法,如:纯真(Naive),半纯真(Semi-naive), Q/SQ, 魔集(Magic set)等。

在已实现的许多演绎数据库系统(或原型)中,最有影响的是DATALOG。它的作者之一, Cardatin, 曾声称DATALOG不但在知识处理方面,而且在传统的数据处理领域中都具有广泛的应用前景,例如取代COBOL语言。但事实并非如此。虽然此系统在学术研究上产生了重要的影响,但在实际应用上却表现逊色,距设计者的设想相差甚远。我们认为其主要原因是:系统提供的用户语言为一个简单的查询语言,它设有函数;系统原语也不丰富;自底向上的计算需要大量的空间等。

我们知道,虽然关系数据库系统在命令方式下提供了一个良好的用户界面,如用户数据录入,数据查寻,自动报表等。但在实际应用中往往还是要嵌套在某一程序设计语言中使用,如COBOL, C等。这说明在实际应用中仅使用查询语言是不够的。一个仅提供查询语言,而不能提供程序设计语言或不能嵌套于某一程序设计语言使用的数据库系统是不太实用的。

我们认为演绎数据库的用户语言进化为一个灵活的程序设计语言是困难的。因此演绎数据库的基本出路可能还是要嵌套于某一程序设计语言中使用,而且这一语言应当足够高级。

与关系数据库相反, PROLOG系统是一个功能很强的可演绎系统,同时又是一个程序设计语言。但是同演绎数据库的命运相似,它也没有在实际工程中获得广泛的应用,其主要原因是:早期的PROLOG系统低效、封闭和缺少高效外存管理机构。为了改善PROLOG系统性能,人们做了大量努力,并且已取得丰硕成果:基于Warren抽象机的PROLOG编译技术和部分求值技术改善了PROLOG效率;同C, FORTRAN, PASCAL语言的接口改善了PROLOG的封闭性;同关系数据库的连接技术可解决外存管理问题。前

两点都已实现,最后一点还在研究中。

由于PROLOG有一个数据库解释,而关系数据库有一个逻辑解释,同时它们的优缺点恰恰是互补的,因而把这两个系统融为一体是非常自然的。但是由于PROLOG和关系数据库采用了不同的推理方式(正向和反向链接),及不同的求值策略(一次一个元组,一次一个集合),因此融合这两个技术并不容易。考虑这些困难,曾有人得出这一方向是没有前途的结论。但是这一悲观的论点并没有阻止这一领域的发展,人们已提出了耦合这两个系统的一系列方法,如:编译,解释,半编译和半解释方法,其中半解释方法具有较好的实用价值。

二、一个实用的可演绎系统

由中山大学计算机系信息科学实验室同法国里昂 INSA 信息工程实验室合作研制的 PROLOG-DBMS 系统是一个实用的可演绎系统,因为它具有如下特性:

1. 一个好的界面。提供一个完整的 PROLOG 语言; PROLOG 和 DBMS 系统都可完全独立地运行;对数据库访问的组织,执行和对提取数据的消耗完全对用户透明。

2. 系统的正确性。DBMS 中的元组完全等价于 PROLOG 程序中的事实,或者说 DBMS 中存贮的是 PROLOG 子程序而不是数据。

3. 系统的效率。由于采用了部分求值技术,使得系统极大地利用了 DBMS 提供的连接操作,因而大大地减少了 PROLOG 与 DBMS 间通讯次数;由于采用的是半解释方法,使得用户程序中变量充分地例化,因而所有从 DBMS 中提取的元组恰是生成 SLD 归结树时所用到的全部事实;由于采用了子句间优化技术,使得数据库中的任何一个元组都不会被多次提取,这不但对提高效率而且对保证系统正确性都是必要的;由于 PROLOG 和 DBMS 间通讯界面是系统用 C 和 ES-QL 语言自动生成,并且每一用户程序对应于一个专用通讯界面,因此减少了每次数据库

访问的代价，因而提高了效率；由于 PROLOG 和 DBMS 进程并行执行，信息交换和同步通过管道完成，减少了每次通讯的代价，因而提高了效率，并且在多机或者分布环境中可进一步提高效率。

4. 系统灵活性。用户可完全独立地使用 PROLOG、DBMS、部分求值器和 PROLOG-DBMS 系统。系统提供了 PROLOG 与 DBMS 间多种通讯方式，直接方式（一个可执行文件）或者间接方式（多个程序并行执行），后者为 PROLOG 系统同时同多个或多种 DBMS 高效连接提供了便利，例如：同 INFORMIX, INGRES, ORACLE 等关系数据库系统，这对于分布式应用是十分重要的。系统是在 SUN 工作站，UNIX 环境下完成，用 PROLOG 和 C 语言（各占 99% 和 1%）完成，移植方便。系统还提供了同 DBMS 通讯的中间语言，可以方便地同任意以 C 语言作为宿主语言的关系数据库系统进行连接。

三、进一步的研究

文中提到过，演绎数据库可实际应用的基本出路可能是嵌套于某一高级程序设计语言中使用。我们认为融合 PROLOG 和演绎数据库技术也是十分自然的，因为它们的优缺点恰恰是互补的，正向和反向链接技术的结合可大大地提高系统的时空效率。

为了保证 DBMS 系统的效率，目前的 DBMS 系统对数据表示作了较强的限制，至少要求数据库中的关系是平面的。但是，对下关系：

R	A ₁	A ₂	
	a	b	表示为
	a	c	
	a	d	

R	A ₁	A ₂	
	a	{b, c, d}	

是十分自然和高效的，这点不但对数据库而且对于逻辑程序设计也是如此。目前，这一问题已经引起了计算机科学家广泛的注意，LPS (Logic Programming with Set) 和 Col 等逻辑程序语言被相继提出，但是距实用还十分遥远。我们在这方面也进行了探索，实现一个改进了的 LPS 语言解释器原型，进一步我们将为实现一个可实用系统而努力。

新的环境和工具可能会对递归查询的研究产生影响，我们希望得到一个较实用的方法解决递归查询问题。

数据库系统处理的是数据之间的关系，而可演绎系统处理的是关系之间的联系，因而一个知识处理系统首先是一个数据库系统，例如，PROLOG, DATALOG 等，所以以数据库作为研制可演绎系统的基础是十分自然的。

目前由于演绎数据库的研究（例如 DATALOG）在提供程序设计语言方面遇到了极大的困难，进展缓慢，因此改善 PROLOG 性能，使之成为一个实用的可演绎系统势在必行，并可能是真正的出路。

人们对于 PROLOG 与 DBMS 的连接技术已进行了广泛的讨论，例如松或紧耦合，其中紧耦合可以理解成把数据库访问程序同 PROLOG 系统连接成一个可执行文件，但这只能解决每次数据库访问的代价问题。同时也带来新的问题，例如：并行处理，同多个、多种和分布式数据库相连问题。这样的系统实现也是困难的。

在我们的系统中 PROLOG 与 DBMS 系统是并行运行的，实验结果令人满意，我们认为已达到实用水平。

此项目是在 J. Kouloumdjian 教授（法国）的指导和霍英东基金的资助下完成的。（参考文献略）