演绎的面向对象数据库研究

王 浩 张奠成
(合肥工业大学人工智能应用研究室 合肥 230009)

摘 要 文中从复杂对象逻辑语言、面向对象逻辑基础及面向对象数据库演绎查询语言等三个方面研究了一些典型的演绎面向对象数据库(DOOD)的成果,对它们进行了分析和比较,并就目前研究中一些重要的尚待解决的问题作了进一步讨论.

关键词 面向对象数据库,演绎面向对象数据库,演绎查询语言

中图法分类号 TP311. 13

RESEARCH ON DEDUCTIVE OBJECT-ORIENTED DATABASES

Wang Hao and Zhang Diancheng
(*Institute of Artificial Intelligent Application, Hefei University of Technology, Hefei* 230009)

Abstract Some typical achievements of deductive object-oriented databases (DOOD) are reviewed from three aspects: complex-object logic language, the logic basis of OODB, and deductive query language of OODB. These achievements are analyzed with emphasis on the discussion about some important problems that have to be solved completely.

Key words object-oriented database, deductive object-oriented database, deductive query language

Class number TP311. 13

1 引 言

演绎数据库(DDB)和面向对象数据库(OODB)自 80年代以来,它们分别沿着不同的道路独立地发展,在各自领域的理论、技术及系统等方面都取得了许多成果. DDB的研究是典型的理论驱动的,而 OODB的研究是典型的应用驱动的,这两方面的研究具有很大的互补性. OODB的主要问题是缺乏一个形式语义,而 DDB使用的是关系模式,不支持对象标识和数据抽象概念,因此将两方面研究结合起来,以复杂对象建模和数据抽象能力为核心,以基于规则的 OODB查询语言作为接口语言是新一代数据库的基本特征.

演绎的面向对象数据库(DOOD)的研究开始于对复杂对象增加演绎推理的研究,这方面工作从理论和实践两方面开展,理论方面主要是从 DDB理论出发,在其数据模型中增加了对复杂数据项支持,进而明确结合了对象标识、继承及方法等面向对象的特征,并提出了有关的对象查询的语言. 在实践方面,主要是在

原稿收到日期: 1996-09-19;修改稿收到日期: 1997-04-14. 本课题得到国家教委博士点专项科研基金资助. 王浩,讲师,博士,研究领域: 人工智能应用、面向对象方法、面向对象数据库. 张奠成,教授,博士生导师,研究领域: 知识工程、人工智能应用、智能 CAD软件工程.

OODB或扩展的关系数据库基础上增加语义规则.如 Kim等人在其研制的 ORION系统上增加了演绎推理能力.而由 Stonebraker等设计研制的 POSTGRES系统则是一种嵌入式规则系统.不过,这些实践方面的探索,正如 Kim本人所说的,其实质不能说是 DOOD,因为它们不能直接支持系统维护、归结算法等逻辑概念,只是在已有数据库之外增加了一个演绎规则处理层[1].

本文从复杂对象逻辑语言、面向对象逻辑基础及面向对象数据库演绎查询语言等三个方面简要分析和比较了一些典型的 DOOD研究成果,对对象标识、继承性、方法、约束和实现等有关 DOOD研究的重要问题作了进一步的讨论.

2 典型 DOOD模型与语言介绍

2.1 复杂对象逻辑语言和模型

扩充关系模型以表达结构化数据的研究早在关系模型提出不久就开始了[2],大部分工作在两个方面开展,即如何对实际的数据结构模型化和如何在这样的模型上定义查询语言.传统的 DDB查询语言是以 DATALOG为基础的,无法表示复杂的结构数据.虽然利用 PROLOG可以表示某些类型的数据结构,但对聚集操作(如集合操作)则不支持.

Kupper等人提出了 LPS语言[3],该语言以多类型逻辑中的二类逻辑为基础,具有两种类型的对象:单个对象(individual objects)和集合对象.LPS在规则体中使用全称量词来存取集合,它被看作是在集合元素上的一个非过程迭代.在 LPS中,只能处理"平坦"的集合对象(即集合元素不能是集合),这被认为是 LPS的一个主要缺点.

Tsur等人研制一个类似于 LPS的逻辑语言 LDL[4,5],其功能更强一些.与 PROLOG不同,LDL程序员不需要显式控制规则中的谓词执行次序及程序中规则的执行次序.在 LDL中提出了一系列新的概念,使用集合成组操作(set-grouping)来构造集合,使得语言表达能力增强.但为了满足最小模型唯一性语义,必须增加一些语法上的限制.例如在 LDL中,一个简单的数据库 P: $[(P((X)) <- P(X)),(P(1))]$就不存在模型(称之为集合悖论).而数据库 $[(P(X) <- Q(X))]$可能具有两个模型,而它们的交则不是模型.这些与传统 LP理论的不一致,要求 LDL用户只能按某些特殊的约定设计应用系统,因而带来了不便.

Abiteboul提出的复杂对象操作语言 COL[6]是在 DATALOG之上通过引入元组和集合来表示复杂对象的定义和操作,与 LDL和 LPS不同,COL语言引入了值为集合的数据函数来处理复杂对象.COL中的数据函数与 LDL中的集合成组操作相对应,数据函数引入可以使得 COL能处理否定问题.COL程序的语义也是建立在最小模型基础上的,与 LDL类似,由于引入集合和数据函数,对某些 COL程序可能有不只一个最小模型.

LPS,LDL和 COL对集合构造和检索方面的能力是基本相同的,但均存在以下不足:① 对复杂对象的子结构存取方法不一致,其文字的语法结构和语义依赖于它所要处理的对象的层次结构.② 仅能处理复杂对象,不具有基本的面向对象特征,如没有对象标识,不能处理 IS-A关系.③ 缺少有效的计值方法,自底向上的计值方法由于引入太多的概念,在很多情况下不再适用.例如 LDL的规则无范围限制导致域的非独立性,可能引起不安全的查询.在 LPS中全称量词的实现显得过于复杂.

2.2 面向对象的逻辑基础

复杂对象模型和语言的研究为 DOOD数据模型和语言打下了基础,以美国 SUNY石溪分校的 Kifer为代表的一些学者从逻辑的角度研究面向对象数据库,描述了面向对象数据库的逻辑基础,提出了面向对象逻辑语言.这方面的工作可追溯到 Maier在 1986年发表的论文[7],在那篇文章中,Maier提出了对象逻辑(O-LOGIC),不过在 O-LOGIC中复杂对象不包含集合,另外在演绎方面也缺少处理不一致信息的能力.

在 O-LOGIC基础上,Kifer等人提出了一个扩充的面向对象逻辑(OO-LOGIC)[8],在 OO-LOGIC数据模型中增加了对集合、类型以及对象标识的支持.在 OO-LOGIC中的对象构造是语言中显式表达部分,它不区分原子数据值和其它基本对象.原子值(如数值)既是一个对象又是其对象标识.OO-LOGIC由于引入类

型和对象标识,使得对集合定义和处理比 LPS、LDL和 COL更为有效,它的类型机制消除了 LDL集合悖论.集合中的元素可以是对象标识,使得集合可以有任意深度有嵌套并可表示无限集合.

与此同时,W. Chen等人也在 O-LOGIC基础上提出了一个关于复杂对象的逻辑 C-LOGIC[9]. C-LOGIC的策略重点放在复杂对象的建模上,语言的设计由模型需要来指导.它支持复杂对象的一些基本特征,如对象标识、多值属性及动态类型等概念. C-LOGIC虽然包含高阶构造,但可将 C-LOGIC公式转换为一阶逻辑公式,使其具有一阶语义.其它高层特征如单值属性、静态类型概念可加在 C-LOGIC顶层之上.实际上,C-LOGIC可看成是 OO-LOGIC中去掉单值属性后生成的一个子集[8].

值得指出的是,无论是 LPS、LDL、COL,还是 O-LOGIC、OO-LOGIC或 C-LOGIC,上述系统对继承机制均未作全面考虑.这些方法所涉及的只是简单的类-子类关系,但子类并不能继承超类的值.另外对方法的研究在这些系统中也未体现,因此在全面考虑了面向对象和基于框架语言(FL)的特征之后,Kifer等人提出了 F-LOGIC[10]. F-LOGIC是一个全面的逻辑解决方案,满足上面所提到的要求. F-LOGIC也适用于 AI中基于框架的语言,实际上 AI中的框架可看成是一个缩小的复杂对象,它具有标识、继承和推理特征.

对于继承和模式的推理需要用到高阶逻辑,许多研究表明,高阶逻辑的某些部分通过谓词演算可具有一阶语义.但这只是间接的语义,并与 OOP和 FL风格不一致.与 HILOG[11]一样,F-LOGIC具有高阶语法,但具有一阶语义.另外,F-LOGIC具有明确的模型论语义和有效且完备的基于消解的证明过程. F-LOGIC的提出者试图为 OODB建立一个坚实的逻辑基础,并在此基础上建立对象查询语言[12,13].从某种意义上来说,F-LOGIC和 OOP之间的联系正如经典的一阶逻辑和关系程序设计的联系一样[14].

2.3 面向对象的演绎查询语言

DOOD研究的另一个方面是从面向对象数据库系统本身出发,研究其演绎查询语言.在这方面,Abiteboul提出的 IQL[15]可以说是面向对象的演绎查询语言的先驱.

Abiteboul认为对象标识(object identify,OID)是面向对象数据库查询语言基本原语(primitive),对象标识是一个具有丰富类型的数据模型、继承机制及强大的查询语言的中心.从直观上理解,OID是一个类型化指针,在 IQL中,OID用作有向(可能有环)图的编码、集合的操作等,使得查询语言表达能力增强. IQL语言具有三个基本性质:① 它是基于规则的,是大多数基于规则的逻辑数据语言,尤其是 COL在 OODB情形下的推广;② 它可以进行静态类型检查,因而可以控制指针的使用;③ 在严格表达数据库操作所必须的性质基础上,它是完备的,IQL语句的语义具有膨胀不动点(inflationary fixpoint)特性,可以用自底向上的方式计值.

IQL将模式与实例概念加以区别,并具有程序设计的强类型机制,这在其它语言中是不具备的,如 Bancilhon提出的对象演算[16]、OO-LOGIC等均没有将实例与模式分离,这些查询语言只能看作是对无类型的 PROLOG语言的扩充.进一步,在结合了 COL和 IQL语言的特征并增加了对方法的支持后,Abiteboul又提出了一个较为完全的 DOOD语言[17].这个语言的模型在 IQL模型基础上增加了两方面内容:方法和表(LIST)类型.方法是满足某一类型约束的函数,它可以以内涵或外延方式来表达,方法可以被继承,并可重载.不过为了满足面向对象和面向值两方面特征,Abiteboul提出的模型保留了关系及关系操作,带来了较大的冗余,另外在实现上还存在将核心语言的不动点语义与外部控制的过程语义相协调的问题.

面向对象演绎查询语言的一个实际例子是 Cacace等人提出的 LOGRES[18]. LOGRES的数据模型是建立在 OO方法上的,它支持对象类、分类层次和对象共享特性,在实例层次上引入对象标识. LOGRES的数据库静态结构是建立在类型等式基础上的.类型等式满足了多种数据库结构特性,特别是完整性约束,它是由类型相同建立起来的. LOGRES的语言支持集合、多集(multisets)、序列(sequences)及受控的否定形式.通过基于规则的范式,可以完成查询和更新. LOGRES系统支持模块,模块是一组类型等式和规则的集合.将一个模块应用到某个 LOGRES数据库时,用户可以指定应用方式,因而可以控制模块应用的副作用.应用方式同时也指定了规则应有的特殊语义.利用这个机制,LOGRES模块和数据库与它们支持的规则之间有对应的参数化语义性质.

LOGRES与 IQL在很多方面上是相似的,但对 IQL的最大改进是使用数据函数,它有两个用途:一是

执行集合方面的操作(nesting和 unnesting操作),二是允许直接应用继承多态.IQL中用对象标识进行集合类型的操作,这降低了程序的易读性和说明性.在 LOGRES中,没有联合类型(union types),可直接处理继承性,而在 IQL中继承性是通过联合类型间接处理的.

3 DOOD研究中几个问题的讨论.

3.1 对象标识

OID概念在早先的一些扩展的关系模型中被提出来,如在文献[19]中称为代理(surrogates),在文献[20]中为 L-值(L-values).作为 OODB的基本特征,对象标识仍有许多问题没有圆满解决,包括实例类型化问题和对象创建后的命名问题.在 IQL和 LOGRES中,OID来源于一特有的无限类(sort),对象创建时,就从那个类中取一个新的 OID,OID是用户不可见的.而在 OO-LOGIC和 F-LOGIC中,新对象是用斯科林函数来表示的,在定义规则时需显示说明此函数,即可直接控制它们的名字.OID的引入带来了许多问题,如重复消除[15],增大用户负担[21]等.应当指出,创建对象(不论是来自无限类还是基于斯科林函数)的使用明显偏离了传统数据库的方向,因为这样会产生潜在模型,而关系数据库理论在很大程度上可看成是有限模型理论的一个子集[22].

3.2 继承性

在自动推理研究中,继承是作为一种获取特定信息(即分类信息)的方法,是由演绎推理实现的.在语义网络的研究中关于继承性和分类信息的处理人们提出了一些方法,但这些工作要么回避了如何改进可操作推理过程,要么其形式公式的语义丢失了像 PROLOG那样简单完美的表达.在逻辑程序语言内部建立继承机制的早期系统是 LOGIN[23],在 LOGIN中提出了一个简单有效的范式,允许将继承性从 PROLOG逻辑推理中分离出来.LOGIN给出了一个可表示记录类型结构的 ψ项,以及其类型合一算法,其中 ψ项可用于复杂对象的演绎推理.但 LOGIN不适合于数据库查询语言[10].

OO方法的封装性要求在一个既是面向对象同时又是逻辑的语言中对象应包括一个逻辑程序模块.这意味着可具有一些导出的属性,而这些属性是由诸如 DATALOG子句等形式的一个逻辑规则集来定义的.因此 OO方法中的继承性,就对逻辑对象如何重载继承的规则提出了要求.目前有两种方法处理这个问题:一是语法的方法,即将相关的规则替换或重新定义,这样就必须对规则进行显式命名.这种方法除了要记住规则名的不便外,如果仅需要改变一部分(如增加一些条件,例外情况处理)继承规则,而不是全部替换它们的话,这种方法是不允许的.另一个方法就是语义的方法,即将继承的规则看作为是"缺省"的,它可以通过附加一些信息,甚至有一点冲突的信息来进行重载.

缺省推理已在人工智能中进行了研究,同时在演绎数据库中也有应用,但只是一些固定的缺省:如否定的基文字,或在分层逻辑程序的文字层次上.实际上,这些都是"闭世界假设(CWA)"的要求,即肯定的知识显式地存在数据库中,而否定的知识被认为是缺省的.而现在则要求显式地指定缺省,即提出"用户指定的缺省",而不是系统指定的缺省[24].在许多 DOOD研究中,继承性仅用于对象的属性和方法名(及定义上),推理规则只是在对象间引入,即在对象模块的层次之上,而不是在对象内部.而对象内部的继承性和重载才是逻辑程序设计与 OO范式集成的焦点[17].

3.3 方法

演绎语言没有与过程语言中方法相对应的概念,已提出的演绎的面向对象查询语言,很多都忽略了方法及方法继承.F-LOGIC使用了非基 ID项作为方法定义的标志.在 F-LOGIC中,一个子类可以是超类的一个实例,类可以作为对象来使用,由于这种概念重叠,它混淆了一个类定义的方法是作为其子类上的方法还是此类中单个对象的方法之间的区别.在对象和类之间定义方法也有一些限制,例如关系模型中象连接这样的操作也无直观的定义.IQL和 ILOG[25]提出了对象标识符创建的概念,但也仅讨论了数据继承,没有清晰的方法概念,在文献[17]中,提出了方法是满足某种类型限制的函数,并通过示例讨论了方法继承和重载,但并未给出理论保证.另一方面,HILOG[11]讨论了具有一阶语义的高阶语法,虽然没有直接讨论 OO特征,但已

有了定义方法的机制,并且参数化多态性可以容易地实现.不过,HILOG没有说明如何实现继承.

方法继承的多态性已成为将逻辑查询模式转换为 OO查询模式的主要障碍,在 HORN 子句框架下,类和其子类结构上的差异不允许一个类的方法应用到其它子类上.在这方面,LLO[26]引入了元变量概念.元变量可用于数据抽象、信息隐藏和继承性.元变量的实例化建立了类型继承层次.在超类型中其子类型中部分结构隐藏在元变量中,这使得一个类型的实例具有统一的结构,因此定义在超类上的方法可由子类共享.进而方法继承可以由通过对与方法相关类型的检查一致地实现.所要做的工作就是在合一算法内加入处理有关类型 /类信息的部分,而封装和抽象数据类型可以由函数 ID项完成.

3.4 约束

在设计和规划领域,OODB以其可以自然表述复杂结构对象的能力得到了广泛应用,但在这些领域中,存在着大量的不完全信息,需要对求解的问题加以约束[27].约束逻辑程序设计[28]将传统的逻辑程序设计的语义域由 Herbrand域推广到特殊的数据域,将传统的解决约束问题的专用过程或方法引入到逻辑程序设计中,使得系统不但具有逻辑语言的说明性语义及关系描述的方便性,而且兼有专用程序的高效率的运行,大大提高了逻辑语言的描述能力和推理效率.

将约束引入演绎的面向对象数据库,可以解决相当一部分不完全信息和全局约束的处理,但由于问题的复杂性(约束问题通常具有非多项式复杂度),还有大量问题难以解决.在实现实际的约束求解器时,需要解决三个重要问题:① 域压缩,即通过约束传播使未知变量的搜索空间减小.② 使用域相关的启发式知识可降低复杂度,但困难在于如何将启发式知识与约束程序的说明性特征相结合.③ 编译技术,即将约束编译成简单的、易管理的单元.在大型应用中解释的约束求解器不能更好地使用.

Caseau在 LAURE语言中,提出并实现了一些技术,并应用到一些大型应用中[29].在 LAURE中,使用关系代数表示约束,因而支持已有的查询优化中发展起来的编译技术.采用产生式规则和约束混合的形式,使启发式知识的表达自然且有效,而域压缩则通过关系代数的抽象解释来实现.LAURE将 CHIP涉及的整数域扩展到了一般的有序域(order-sorted domain).其数据模型是 OO-LOGIC的一个子集.

文献[30]将约束引入演绎的面向对象数据库模型,建立了一个约束演绎面向对象的数据模型(简称为 CDOOD模型),这种模型不仅拥有面向对象的自然的数据表示、演绎模型的知识推理等优点,而且由于显式的约束处理技术使系统具有高效的运行特性.在 CDOOD中使用的约束求解器代替消解合一,并对魔集计值方法在约束求解上的应用进行了推广.不过 CDOOD模型也存在着一些限制,如继承性、封装、重载等面向对象特征如何与演绎、约束相结合等问题,还有待进一步研究.

3.5 实现

DOOD实现的途径有[22]:① 扩充关系系统;② 扩充 OODB系统;③ 扩充 DDB系统;④ 扩充 KBMS和 ES(专家系统);⑤ 从头设计.从目前提出的理论的模型来看,大量的工作是采用扩充 DDB方法来进行的,不过这方面工作也存在以下问题:

(1) 没有考虑一般数据库语言的基本要求,例如,F-LOGIC的研究不是着眼于面向对象数据库查询语言的,在其模型中实例和类型概念重叠,因而没有数据库模式的概念.

(2) 演绎查询语言过于复杂,如为了使面向值和面向对象特征能够统一,IQL保留了关系概念,将给实现带来了困难.另外用 OID的创建来完成查询,这也不符合人们对数据库查询理解.LOGRES集数据查询与更新于一体,使得语言过于复杂.

(3) 对 OODB数据模型缺少统一认识,各个研究都是基于各自的数据模型,没有考虑信息建模能力以及便于实现等具体要求.如 F-LOGIC未提供一个有效的计值策略[31],而 IQL中关于程序语义的膨胀不动点计算的实现和优化也还没有满意的解决方案[14].

另一个值得指出的方面是面向对象逻辑程序设计(OOLP)[32],OOLP与 DOOD关系犹如 LP与 DDB关系.OOLP有关对象表示、继承性、方法及重载等方面处理可供 DOOD研究中参考.例如,与将对象作为逻辑语言的一个结构项不同,在 Dalai等人提出的 OOLP+ 中[33],将对象作为一组命名的关系断言.这使得 OOLP+ 程序很容易地转换为 PROLOG程序而无需引入任何元谓词,并使用 DATALOG中已有查询优化

技术.

最后,必须指出是,由于目前还未有成熟的商品化的 DDB系统[21],因此大多研究只是一些原型系统,在这些原型系统上进行面向对象的扩充需要大量的改写工作且必须保持原系统的性能,因此可以断言,DOOD离实用化还有很长一段路要走.

4 结束语

DOOD研究一方面可为 OODB建立一个坚实的基础,因为目前 OODB还没有公认的数据模型,没有形式化基础,因此在不同的系统中,对 OODB定义和操作语义差别很大.另一方面可在统一的形式化框架下,将 OODB, KDB(知识库),复杂对象(包括 NF²,嵌套关系)等数据库新领域的研究成果统一起来. 这些方面的研究成果的结合,可望逐渐明确并统一新一代数据库的数据模型、查询语言、体系结构以及其它特有问题和技术的认识,开辟研究的新局面.目前 DOOD研究仍处于实验探索阶段,还有许多问题需要深入研究.本文的目的是旨在对当前 DOOD研究现状及存在的问题进行分析和讨论,为进一步研究和开发智能化的面向对象系统奠定基础.

参 考 文 献

1 Kim W.面向对象数据库系统的研究方向.计算机科学, 1991, 18(3): 1— 12

2 石桥林.复杂对象数据库研究:历史和现状.计算机科学, 1991, 18(2): 21— 27

3 Kupper G. Logic programming with sets. In: Proc of the 6th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems. San Diego, 1987. 11— 20

4 Naqi S, Tsur S. A Logic Language for Data and Knowledge Bases. Roclville, MD: Computer Science Press, 1989

5 Beeri C *et al*. Sets and negation in logic database language (LDL1). In: Proc of the 6th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems. San Diego, 1987. 21_ 37

6 Abiteboul S, Grumbach. COL: A logic-based language for complex objects. In: Schmidt JW, Ceri S, Missikoff eds. Advances in Database Technology— EDBT'88. Venice: Springer-Verlag, 1988. 271_ 293

7 Maier D. A logic for objects. In: Proc of Workshop on Foundations of Deductive Database and Logic Programming. Washington D C, 1986. 6_ 26

8 Kifer M, Wu J. A logic for object-oriented logic programming (Maier's O-LOGIC Revisited). In: Proc of the 8th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems. Philadelphia, 1989. 379_ 393

9 Chen W, Warren D. C-LOGIC of complex objects. In: Proc of the 8th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems. Philadelphia, 1989. 369_ 378

10 Kifer M, Lausem G. F-LOGIC: A higher-order language for reasoning about objects, inheritance, and scheme. In: Proc of ACM SIGMOD on Management of Data. Portland, Oregon, 1989. 134_ 146

11 Chen W, Kifer M, Warren D. HILOG: A first-order semantics for higher-order logic programming constructs. In: 2nd Int Workshop on Database Programming Languages. Cleveland, 1989. 1090_ 1114

12 Gardarn G, Valdurez P. ESQL2: An extended SQL2 with F-LOGIC semantics. In: Proc IEEE Int Conf on Data Engineering. Phoenix, 1992. 294_ 366

13 Frohn J, Lausen C, Uphoff H. Access to objects by path expressions and values. In: Proc 20th Int Conf of Very Large Data Base. Santiago, 1994. 273_ 284

14 周傲英,施伯乐等.基于规则的对象数据库查询语言.计算机科学, 1995, 22(3): 65_ 68

15 Abiteboul S, Kanellakis P. Object identity as a query language primitive. In: Proc of ACM SIGMOD on Management of Data. Portland, 1989. 159_ 173

16 Bancilhon F, Khoshafian S. A calculus for complex objects. In: Proc of the 5th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems. Cambridge, Massachusetts, 1986. 53_ 59

17 Abiteboul S. Towards a deductive object-oriented database language. In: Proc Int Conf on Deductive and Object-Ori-

ented Database. Kyoto, 1989. 453_ 472
18 Cacace F, Ceri S *et al*. Integrating object-oriented data modeling with a rule based programming paradigm. In: Proc of ACM SIGMOD on Management of Data. Atlantic City, NJ, 1990. 225_ 236
19 Codd E F. Extending the database relational model to capture more meaning. ACM Trans on Database System, 1979, 4(4): 397_ 434
20 Kupper G, Vardi M. A new approach to database logic. In: Proc of the 3th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems. Waterloo, 1984. 86_ 96
21 Ullman J. A comparison between deductive and object-oriented database systems. In: Proc Int Conf on Deductive and Object-Oriented Databases. Munich, 1991. 263_ 277
22 施伯乐等.演绎的面向对象数据库.计算机应用, 1995, 15(3): 1_ 5
23 Ait-Kaci H, Nasr R. LOGIN: A logic programming languange with semantics of inheritance. J Logic Programming, 1986, 3(3): 185_ 215
24 Brass S, Lipeck U. Semantics of inheritance in logical object specifications. In: Proc Int Conf on Deductive and Object-Oriented Databases. Munich, 1991. 417_ 430
25 Hull, Yoshikawa R. ILOG: Declarative creation and manipulation of object identifiers (extended abstract), In: Proc 16th Int conf on Very Large Data Base, Brisbane, 1990. 455_ 468
26 Lou Y, Oisoyoglu Z. LLO: An object-oriented deductive language with methods and method inheritance. In: Proc of ACM SIGMOD on Management of Data. Denver, 1991. 198_ 207
27 Morgenstern M *et al*. Constraint-based systems: Knowledge about data. In: Kerschberg L ed. Proc of 2nd Conf on Expert Database Systems. Virginia, US: George Mason University, 1989. 23_ 43
28 Jaffar J, Lassez J. Constraint logic programming. In: Proc of 14th ACM Symposium on the Principles of Programming Languages. Munich, 1987. 111_ 119
29 Caseau Y. Constraint in an object oriented deductive database. In: Proc Int Conf on Deductive and Object-Oriented Databases. Munich, 1991. 292_ 311
30 李修华.约束演绎面向对象的数据模型研究 [博士论文].合肥工业大学, 合肥, 1993
31 Heuer A, Sander P. Semantics and evaluation of rules over complex objects. In: Proc Int Conf on Deductive and Object-Oriented Databases. Kyoto, 1989. 473_ 492
32 徐殿祥,郑国梁.对象式逻辑程序设计.计算机研究与发展, 1996, 33(1): 17_ 23
33 Dalai M, Gangopadhyay D. OOLP: A translation approach to object-oriented logic programming. In: Proc Int Conf on Deductive and Object-Oriented Databases. Kyoto, 1989. 593_ 605

全国科学技术名词审定委员会推荐的部分网络名词(1997年 7月发布)

internet	互联网	TCP(transmission control protocol)	传输控制协议
internetwork	互联网		
interconnection network	互联网	Telnet	远程登录
Internet	因特网	FTP(file transfer protocol)	文件传送协议
WWW(world wide web)	万维网	E-mail	电子函件
Web	万维网	browser	浏览器
hypertext	超文本	Archie	阿奇 [工具]
hypermedia	超媒体	ISP(Internet service provider)	因特网服务提供者
home page	主页	directory service	名录服务
DN(domain name)	域名	fire wall	防火墙
WAIS(wide area information server)	广域信息服务系统		