关于在演绎数据库系统研究中引入一种新的关系的一些探讨*

徐洁磐

(南京大学计算机系,南京 210093)

摘要 本文提出了在演绎数据库系统研究中的一种新的关系叫虚－实关系.虚－实关系集成了实关系与虚关系的一些优点,克服了它们的一些缺点,在演绎数据库系统中它具有特别的性质.将虚－实关系引入演绎数据库系统后我们可以扩大演绎数据库系统的功能.目前,虚－实关系已在SUN－3工作站上的演绎数据库系统UNIFY中实现并正在用于CAD图形领域中.

关键词 演绎数据库,虚－实关系,dUNIFY.

在演绎数据库中有两种关系,一种叫实关系(Real relation),它构成了演绎数据库中的外延数据库(External database),另一种叫虚关系(Virtual relation),它构成了演绎数据库中的内函数据库(Internal database).实关系即是关系,它实际的存在于演绎数据库中,它提供了演绎数据库中的基础数据,而虚关系则用规则表示,它的数据并不实际存在于演绎数据库中,仅当需要时通过对规则的演绎推理以取得虚关系中的数据.虚关系的存在构成了演绎数据库的一大特色.虚关系与实关系相比有很多特点:虚关系中数据不必直接录入;虚关系不占用存储空间;虚关系能参予演绎推理.

但是我们也要看到,虚关系的一些优越性,特别是不占用存储空间的"空间"优越性是用"时间"换取的.即虚关系中的数据库获取是用演绎推理得到,而演绎推理是需要时间的,这种时间概念不但包括执行1次演绎推理所需的时间,而且考虑到虚关系中数据库每取得1次均要执行1次推理,因此就整体而言,其时间损失是很严重的,虚关系的这个弱点导致演绎数据库的时间效率低,这样并影响了它的实际使用.因此如果对虚关系能有一定的改进,能根据不同时刻的不同需要随时能进行"时－空"转换.这种需要引出了在演绎数据库中增添一种新的关系的必要性.

在本文中我们拟引入此种关系并命名为虚－实关系(Virtual－real relation).本文第1节讨论虚－实关系的形成与特性,第2节讨论虚实关系中的一些算法,第3节讨论虚－实关系的实现,第4节讨论虚－实关系应用,最后是小结.

* 本文1992－07－06收到

作者徐洁磐,59岁,教授,主要研究领域为数据库,人工智能.

本文通讯联系人:徐洁磐,南京 210093,南京大学计算机系

1 虚－实关系的形成及其特性分析

1.1 虚－实关系概述

首先,我们对虚－实关系作一个概括性的描述:

(1)虚－实关系是一个关系.

(2)虚－实关系是通过规则形式建立的,而不是通过数据录入方式建立的,刚刚建立的虚－实关系具有虚关系形式(或称虚形式).

(3)虚－实关系在建立后根据需要可以形成实关系形式,实际将数据存贮于演绎数据库内,此时虚－实关系具有实关系形式(或称实形式),而由虚形式转变成实形式的过程称为实化过程.此时,虚－实关系在演绎数据库内同时具有虚的版本(规则形式)与实的版本(数据形式),并以实版本为正本,虚版本为副本.

(4)具有虚、实两种版本的虚－实关系根据需要可撤消实版本而仅保留虚版本,这种过程可称为退化过程,而此时虚－实关系返回至虚形式.

根据以上4点描述,我们可以对虚－实关系概括为:虚－实关系是一种可以具有虚形式及实形式两种形式的关系,它的初始形式为虚形式,其后可根据需要在两种形式中根据一定条件进行相互转换.

1.2 两种形式的转换

虚－实关系中虚形式与实形式之间的相互转换是根据一定的条件而进行的,条件一旦满足转换即进行.所谓"条件"有两种表示方法,一种是人工的,另一种是自动的,人工方法即是由操作人员识别条件,当他认为条件满足时即用一专用的"形式转换"命令以完成虚形式与实形式之间的形式转换,而自动的方法则是设法用一种计算机内的软件装置以实现转换,它设有阀门,当满足一定条件时阀门自动打开从而引起转换的自动实现.

下面我们对自动转换作更为详细地探讨.我们设置两个阀门函数,一个叫实化函数,一个叫退化函数.先讨论实化函数.当虚－实关系处于虚形式时,有下列因素可促使它转换成实形式:

(1)使用频率 u :虚－实关系使用频率高可促使其转化成实形式;

(2)使用时间 t :虚－实关系每次使用时间长可促使其转化成实形式;

(3)环境因素 e :外界使用的特殊要求可促使其转化成实形式;

(4)占用空间 s :虚－实关系实化后所占存贮空间大小可影响其转化成实关系;

(5)规则形式 r :虚－实关系的规则形式如递归/非递归,合取符多/少均较大程度影响演绎推理时间,此因素可促使其转化成实形式.

(6)变动频率 m:虚－实关系实化后其基础实关系的变化(如插入,修改等)频率对转化成实关系产生影响.

因此,实化函数 f 是一个具有下列形式的函数:

$$f(u,t,e,s,r,m)$$

设置阀门值为 p,则第一个阀门:实化阀门 F 可形式地表示如下:

$$F(u,t,e,s,r,m)=\begin{cases}1 & \text{当 } f(u,t,e,s,r,m)\geqslant P \text{ 时}\\ 0 & \text{其余情况}\end{cases}$$

当 F 为 1 时,表示实化阀门打开.

类似地,讨论退化函数,当虚一实关系处于实形式时,有下列因素影响其转换成虚形式:(1)使用频率 u;(2)使用时间 t;(3)环境因素 e;(4)占用空间 s;(5)规则形式 r;(6)变动频率 m. 因此,退化函数 f' 是一个具有下列形式的函数:

$$f'(u,t,e,s,r,m)$$

设置阀门值为 P',则第二个阀门:退化阀门 F' 可形式地表示如下:

$$F'(u,t,e,s,r,m)=\begin{cases}1 & \text{当 } f'(u,t,e,s,r,m)\geqslant P \text{ 时}\\ 0 & \text{其余情况}\end{cases}$$

当 F' 为 1 时,表示退化阀门打开.

当虚一实关系处于虚形式时,实化函数处于计算状态(此时,退化函数处于封锁状态),在每次查询后自动计算,一旦实化阀门为 1 时自动引发规则演绎并获得实关系,当虚一实关系处于实形式时,退化函数处于计算状态(此时,实化函数处于封锁状态),在每次查询后自动计算,一旦退化阀门为 1 时自动撤消实关系在外存的存储.

1.3 虚一实关系与数据库

在演绎数据库中实关系建立了外延数据库,虚关系建立了内涵数据库,而虚一实关系在虚形式时建立了内涵数据库,在实形式时则建立了外延数据库. 因此,虚一实关系根据其表示形式而建立不同数据库.

1.4 虚一实关系特性

根据上面讨论,我们可以总结虚一实关系的一些优点与特性如下:

(1)虚一实关系用规则形式建立,可避免繁琐而容易出错的数据录入方式;

(2)虚一实关系可在演绎数据库内同时以虚、实两种形式存在,既具有快速查询的效果,又能参与规则的演绎推理,还可以返回虚形式以减少存储开销.

(3)虚一实关系的形式转换可有人工与自动两种方式,转换方便.

虚一实关系具有虚关系与实关系的共同优点且又避免了各自的缺点,引言中所述的一些困难(如时间效率低等问题)在使用了虚一实关系后均可得到顺利地解决. 虚一实关系的上述特点还使它在实际应用中有着广阔的前景.

2 虚一实关系的增量算法

虚一实关系中的实形式具有关系的形式. 此种关系是由一些实关系通过规则演绎而产生,这种演绎过程较费时间,一经产生后不希望其数据有经常变动. 但是,不幸的是当产生虚一实关系的实关系遇有删除、插入、修改等数据变动时,则虚一实关系的数据也应变动,此时需重新演绎产生新的数据. 一般而言,实关系的增、删、改操作是经常的,因此演绎过程也会经常重复,从而消耗大量时间,特别是对递归规则尤为明显,因此需采取必要的策略以减少此类变动的时间消耗. 在本节中提出一种算法叫增量算法,它可使在实关系变动后对递归规则重新演绎的时间大为减少.

这个算法的大致思想是这样的. 即如有实关系 $R_1,R_2,\cdots,R_n$ 及用它们所构成的规则或规则集 r,经过演绎可产生 VR,如 R_i 有变动,它包括删除与插入(修改是插入加删除,因此

可以不予考虑),这种变动可用$\triangle R_i$表示,这样,$R_1,R_2,\cdots,R_{i-1},R_i\cup\triangle R_i$(或$R_i-\triangle R_i$),$R_{i+1},\cdots,R_n$及$r$可产生$VR'$,若令$|VR'-VR|=\triangle VR$,则在实关系$R_i$中增加(减少)了$\triangle VR$,因此实际上我们只要计算由增量$\triangle R_i$而引起的增量$\triangle VR$就行了.这种算法称为增量算法.增量算法分插入与删除两种,插入时R_i变为$R_i\cup\triangle R_i$,而VR变为$VR\cup\triangle VR=VR'$,删除时$R_i-\triangle R_i$,而VR则变为$VR-\triangle VR=VR'$.

我们假设所讨论的数据库是无环的,其算法都采用目前常用的Semi-naive方法以计算方程.现在开始讨论插入的增量算法.

插入增量算法.

算法输入:

(1)实关系:$R_1,R_2,R_i,\cdots,R_n$

(2)规则(集)r以及由r所构成的关系代数方程:

$$X=F(R_1,R_2,R_i,\cdots,R_n,X) \tag{1}$$

(3)关系VR;

(4)增量$\triangle R_i$;

算法输出:

关系VR'

算法步骤:

(1)对(1)作代入而得如下的方程:

$$X=F(R_1,R_2,\cdots,R_i\cup\triangle R_2,R_{i+1},\cdots,R_n,VR\cup X) \tag{2}$$

(2)用Semi-naive法对方程(2)求解,所得的解即为VR'.

下面证明此算法的正确性.

定理 1. 增量算法是正确的.

证明:按照一般方法,求解VR'的关系代数方程是方程(3),对(3)用Semi-naive算法即得VR',方程(3)的形式如下:

$$VR'=F(R_1,R_2,\cdots,R_{i-1},R_i\cup\triangle R_i,R_{i+1},\cdots,R_n,VR') \tag{3}$$

因为$VR\subseteq VR'$,故有$VR'=VR'\cup VR$,将此式代入方程(3)中的右端即得:

$$VR'=F(R_1,R_2,\cdots,R_{i-1},R_i\cup\triangle R_i,R_{i+1},\cdots,R_n,VR\cup VR')$$

将此式的VR'用X替换即得方程(2),对(2)用Semi-naive法即得VR'.证毕.

由于本文应用Semi-naive算法,因此算法计算时间与计算层次有关,下面定理给出了增量算法在计算层次上的优越性.

定理 2. 设增量算法的方程(3)的计算层次为n,一般算法的方程(2)的计算层次为m,则必有$n\leqslant m$.

证明:方程(3)用Semi-naive方法,其第一次计算即可获得VR的参与而取得较多的VR'的结果,而方程(2)用Semi-naive方法则由于无VR的参与,因此VR'的结果较少,此后每次计算由于前次的VR'结果均有(3)多于(2),因此下次计算VR'结果也总有(3)多于(2).由于方程(3)与(2)的VR'结果是收敛且一致,因此方程(3)计算层次必小于等于方程(2)的计算层次.证毕.

定理1与定理2告诉了我们增量算法的正确性与时间上的优越性.下面举一例说明.

例1:设有实关系双亲关系 $P(x.y)$(见表1),我们用规则定义祖先与子孙关系 $A(x,y)$ 如下:

$$A(x,y):--P(x.y)$$

$$A(x,y):--P(x.z),A(z,y)$$

$A(x,y)$经实化后获得实关系如表2所示.当实关系 $P(x.y)$变动时,如插入元组 $P(h,i)$即$\triangle P=\{(h,i)\}$,则此时可用增量算法求解而获得结果 $A'(x,y)$:

$$A'(x,y)=P(x,y)\cup\triangle P(x,y)\cup(P(x,z)\cup\triangle P(x,z))\ X\ (A(z,y)\ A'(z,y)) \quad (4)$$

对方程(4)用 *Semi—Naive* 方法求解:

第1层即得:

$\{(h,i),(a,c),(c,f),(c,g),(d,h),(h,i),(f,j),(a,d),(g,k),(a,f),(a,g),(a,j),(a,k),(a,h),(c,j),(c,k),(d,j),(a,j),(c,i),(g,i)\}$

用第1层即可求得解.

按一般算法,求解方程为:

$$A'(x,y)=P(x,y)\cup\triangle P(x,y)\cup(P(x,z)\cup\triangle P(x,z))\ X\ (A(z,y)\ A'(z,y)) \quad (5)$$

用 Semi—Naive 算法可得

第1层:$\{(a,c),(a,d),(c,f),(c,g),(f,j),(g,k),(d,h),(h,j),(h,i)\}$

第2层:$\{(a,f),(a,g),(a,h),(c,j),(c,k),(d,j),(g,i)\}$

第3层:$\{(a,j),(a,k),(c,i)\}$

第4层:$\{(a,j)\}$

按一般算法 $A'(x,y)$的计算层次为4,在此例中增量算法的计算层次比一般算法的计算层次要少3层.

关于删除的增量算法与插入的类似,有关定理也类似,此处从略.

增量算法从根本上为虚—实关系的提高(时间)效率提供了保证.

表1

x	y
a	c
a	d
c	f
c	g
d	h
h	j
f	j
g	k

表2

x	y
a	c
a	d
c	f
c	g
d	h
h	j
f	j
g	k
a	f
a	g
a	j
a	k
a	h
c	j
c	k
d	j

3 虚—实关系的实现

我们已在SUN—3工作站上开发了1个演绎数据库,叫dUNIFY.该演绎数据库是建立在Sun UNIFY上的.dUNIFY采用了实关系,虚关系与虚—实关系3种关系,这使得dUNIFY的功能比一般演绎数据库强,我们目前正用它于CAD的图形系统中,还准备扩充至图象与语音处理中.

dUNIFY的虚—实关系实现是通过下列命令完成的:

(1)虚—实关系定义:用一组规则以建立一个虚—实关系;

(2)虚—实关系显示:用此命令以显示虚—实关系的虚形式及实形式;

(3)虚—实关系删除:用此命令以删除指定的虚—实关系,连同它的虚形式表示与实形式表示;

(4)虚—实关系的生成:用此命令以实化一个虚—实关系,实化后该虚—实关系在

dUNIFY 中生成一个实关系,此命令为虚一实关系的实化提供了人工方式;

(5)虚一实关系的返回:用此命令以退化一个虚一实关系,退化后该虚一实关系仅保留其虚形式(即规则形式),此命令为虚一实关系退化提供了人工方式.

命令 4 与 5 还提供了由系统自动执行的方式,它由两个模块:虚一实关系实化模块与退化模块完成.

在实化模块中,系统利用实化函数以启动生成命令产生一个实关系.在 dUNIFY 中,实化函数为:

$$f(u,t,e,s,r,m)=\frac{a\cdot u+b\cdot t+c\cdot e+d\cdot r}{j\cdot s+k\cdot m} \tag{6}$$

其中 a,b,c,d,j,k 为参数,它可由用户根据情况自由选择.u,t,e,s,r,m 为函数 f 的自变量,其中 u,t,s,r,m 由系统计算或由系统统计完成,而 e 则由用户给出.

在退化模块中,系统利用退化函数以启动返回命令退化一个虚一实关系,在 *dUNIFY* 中退化函数为:

$$f'(u,t,e,s,r,m)=\frac{j'\cdot s+k'\cdot m}{a'\cdot u+b'\cdot t+c'\cdot e+d'\cdot r} \tag{7}$$

其中 a',b',c',d',j',k' 为参数,它可由用户根据情况选择,u,t,e,r,s,m 为函数 f' 的自变量,其中 $u,t,,r,s,m$ 由系统计算或由系统完成统计,而 e 则由用户给出.

上述 5 条命令的使用过程是这样的,首先可用命令 1,以建立一个虚一实关系,然后用命令 4,5 以实现虚一实关系的实化与退化,可用命令 2 以显示关系的虚形式与实形式,最后可用命令 3 以删除此关系.

在 dUNIFY 中当实关系的插入(或删除,修改)操作进行时,系统同时需检查与此相关的虚一实关系,若为实形式时用增量算法以修改虚一实关系.

虚一实关系与虚关系一样,不能对其作插入、删除及修改操作,但能对其作查询操作.在查询时,当为实形式时,则相当于对实关系的查询,若为虚形式时则相当于对虚关系的查询.

有关此方面详细情况,读者可参阅文献[8].

4 虚一实关系的应用

虚一实关系有很多特点,它可有很多应用,一般特别适合于下面几个方面:

(1)适合于建立新关系,而这些新关系的全部或部分数据来自旧关系,这些新关系可用旧关系通过规则形式建立并通过虚关系实化最终形成实关系,这样做可以避免了一般建立一个新关系所需的大量数据录入,以及录入时所不可避免带来的数据错误.

(2)适合于经常查询的关系.关系数据库中某些窗口经常需要查询,此时可用虚一实关系以提高其查询效率.

(3)适合于扩充,减缩关系.关系的扩充与减缩即是在关系中扩充一些属性或删除一些属性,这些扩充与减缩在一般关系数据库中都需用专门模块实现,它们大多结构复杂,开销大,而用虚一实关系实现则较为方便,其整个实现过程如:①用规则扩充与减缩;②实化;③删除旧关系;④用规则改名;⑤实化.

(4)虚一实关系在演绎过程中可加快演绎推理.

上面 4 点表示了虚－实关系在功能上与速度上的优越性,在演绎数据库中增加了虚－实关系后,使演绎数据库的功能更强,应用更广泛.

5 结束语

本文在研究演绎数据库应用的基础上提出了一种关系,叫虚－实关系.它弥补了演绎数据库系统在功能上的某些不足,加入了此种关系后,演绎数据库系统具有实关系,虚关系及虚－实关系等 3 种不同性质的关系,从而能较全面地反映不同的实际要求.

我们已在演绎数据库系统 dUNIFY 中实现了虚－实关系的功能,并且经过运行证明性能良好,实际可用.虚－实关系在实际应用中具有良好的应用前景.在 CAD 中,在图象处理中均有应用价值.目前我们正在用 dUNIFY 作为电路 CAD 的支撑环境,用虚－实关系以表示电路板等取得了较为满意的效果,有关此方面的应用将另文介绍.

参考文献

1 Gallaire H, Minker. Logic and database. New York: Plenum Press, 1978.

2 Gallaire H, Minker. Advance in database theory. New York: Plenum Press, 1981.

3 Reter R. Towards a logical construction of relational database, on conceptual modelling. Springer－Verlag, NewYork, Inc., 1984.

4 Nicolas, Yazdanian. BDGEN: deductive DBNS. Proc. of 9th International Computer Conference, Paris, 1983.

5 徐洁磐.演绎数据库及其现状.计算机科学,1987(1).

6 Ullman J D. Principles of database and knowledge base systems. Computer Science Press, 1989.

7 Morsman A. Knowledge representation in CAD field. Proc. of ISSM International Conference, USA, 1990.

8 dUNIFY 使用手册.1991.

SOME EXPLORATION ABOUT INTRODUCING A NEW RELATION IN RESEARCH OF DEDUCTIVE DATABASE SYSTEM

Xu Jiepan

(*Department of Computer Science, Nanjing University, Nanjing* 210093)

Abstract In this paper, a new relation called virtual－real relation in research of deductive database system has been proposed, virtual－real relation can integrate some advantage of real relation and virtual relation, overcome some disadvantage of them, it has a special character in DDBS. After introducing virtual－real relation to DDBS the authors can extend the function of DDBS. Now a virtual－real relation has been realized in DDBS －dUNIFY on SUN－3 workstation and is being applied to CAD and graphic fields.

Key words Deductive database, virtual－real relation, dUNIFY.