维普资讯 http://www.cqvip.com

计算机科学 1996 Vol. 23 №. 4

CO-LOGIC：一种支持约束演绎 OODB 语言的多类型逻辑*)

张奠成　李修华

（合肥工业大学人工智能应用研究室　合肥230009）

摘　要　We have designed a Constraint Deductive Object-Oriented Database Language CDOODL. This paper concentrated on formal aspects of CDOODL. We present a many-typed logic—CO-logic, which supports complex object identifer, strong typing methods, inheritance deductive and constraint solving. The proposed logic has a sound and complet proof procedure based on resolution and efficient constraint solving.

关键词　Deductive database, Object-oriented database, Constraint logic programming, Knowledge base system.

近年来，针对传统数据库技术在一些新的应用领域（如 CAD，CAE，CASE 等）所暴露的缺陷，提出和发展了面向对象方法，以适应这些新的要求。与此同时，演绎数据库也获得一些进展，但大都是在关系数据库提供的工具上进行逻辑程序设计的扩展，且缺乏对复杂对象进行推理的能力。因此，将演绎和面向对象方法结合起来是当前数据库研究的新方向。已提出一些系统：如 ORION 系统，POSTGRES 系统，基于逻辑数据库语言 LDL 等，但其中一些只注意用规则说明和规则调用来扩展面向对象数据库界面，另一些则只支持面向对象方法中一两个基本概念。根据我们的观点，将演绎和面向对象结合的关键在于如何提供一个统一的理论框架，将面向对象的核心概念结合到演绎方法中去。同时能保证一个有效的完备的消解证明过程，M. Kifer 等提出的 F-logic 逻辑[2,3]，是在这方面的一个好的尝试，但其中仍存在一些缺陷。首先，它包含一组对象构造函数，用于演绎过程中构造中间对象标识。显然，这些函数是语言的一部分，将由用户在定义规则时加以定义，会加重用户的负担。其次，在 F-logic 中，可能会出现同一语法符号由于位置不同被解释成不同的含义，导致语义的二义性。最后，F-logic 中同一子类的对象的属性个数可以不同，特别是方法可以作为对象的属性，以及因对象而异的不同动态模式也使数据库的一致性不易维护。总之，F-logic 过于复杂，不是一个实效系统。本文提出一种多态逻辑，作为 CDOODL[1] 系统的理论基础，在 CDOODL 系统中，我们试图将演绎数据库、面向对象方法和约束逻辑程序三方面的优点结合起来，我们将这种逻辑记作 CO-logic，它基于 F-logic，但作了大大的扩充。它具有如下特点：(1)支持面向对象的核心概念，实现面向对象与演绎的结合。(2)具有简明自然的规范描述。(3)它是一个多类型的逻辑系统。(4)具有约束处理能力，约束条件直接进入推理过程，实现约束同演绎、面向对象的结合。

一、CO-LOGIC 的语法

CO-logic 是一种多类型逻辑，系统中常量和变量都是带类型的，以小写字母 $x_1, y_1, a, b, \cdots$ 表之，CO-logic 的类型可以递归定义如下：

(1)原始类型（基本类型），如实型、布尔型等。

(2)如果 $T_1, \cdots T_n$ 是类型，则 $[T_1, \cdots T_i, \cdots, T_n]$ 称为元组类型。

(3)如果 T_i 是类型，则 $\{T_1, \cdots T_i \cdots T_n\}$ 称为集合

*)本文受国家教委博士点专项科研基金资助。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。李修华　博士，研究领域：人工智能应用、演绎数据库、面向对象数据库。

维普资讯 http://www.cqvip.com

类型。这里，必需注意，集合类型中的(元素)子类型必需是相同的。数据库中的对象有不同类型，但属于同类的对象有相同类型。

我们系统中的函数分为用户定义函数和系统构造函数，当用户对派生对象的标识不感兴趣时则这些新对象的标识就由系统构造函数自动产生和维护，可减少用户负担。

类型 T 的项可以递归定义如下：

(1)T 的常量是 T 的项。

(2)T 的变量是 T 的项。

(3)如果 $t_1,\cdots,t_n$ 分别是 $T_1,\cdots,T_n$ 的项，则 $f(t_1,\cdots,t_n)$是 T 的项，这里 f 是映射到 T 的函数，$T_i(i=1,\cdots,n)$可以是 T，但不能是集合类型。

CO-logic 系统的复合项定义如下：

(1)简单的复合项形式为 P:p，此处 p 是项，而 P 是类名；

(2)复杂复合项形式为 P:p[$t_1,\cdots,t_n$]，这里 p 是一个类型为[$T_1,\cdots,T_n$]的项，t_i 对应于 $T_i(i=1,\cdots,n)$，它们或是复合项，或是复合项的集合，P 仍是一类名。

系统中所有类组成一个格，其中具有最小限制元件 ALL，即所有对象的类；另一个是最大限制元件 NONE。类型中的序用 $\leqslant_c$ 来表示。这样，类的限制愈强，相对于 $\leqslant_c$ 来说愈大，特别是 $ALL\leqslant_c NONE$。

设 D^n 表示域 D 上构成的多元组集合$(a_1,\cdots,a_n)$，一个 n 元操作 f 是从 D 的子集 E 到 D 的映射。CO-logic 中的操作包括布尔操作、数值操作、集合操作和串操作。如上所述 D 的意义下，则一个 n 元的关系 r 是 D^n 的子集 E。在 CO-logic 中，我们选用下述关系：标识关系(=和<，>)，布尔关系(→)，数值关系(<，>≤，≥)，类关系($\leqslant_c$)，串关系(In)和集合关系(=，⊆，∈，∉)。

约束是一种关系，其形式为 $C(a_1,\cdots,a_n)$，这里 C 为关系符，$a_i(i=1,\cdots,n)$是参加约束的项，约束关系往往是成组出现的，任何有限的一组约束称之为约束系统。

一复合项也是一复合公式。系统中规则有形式：$t_0:-t_1,\cdots,t_n,\{c_1,\cdots,c_m\}$，其中 $t_i(i=0,1,\cdots,n)$为复合公式，而$\{C_j\}(j=1,\cdots,m)$称为规则的约束系统，式中 n 和 m 可以为零，当它们为零时，规则转化为事实。这样 CO-logic 系统中的查询可以定义为?-$t_1,\cdots,t_n,\{c_1,\cdots,c_m\}$。CO-logic 系统的一个数据库由规则的有限集合组成。

二、CO-LOGIC 的语义

一个逻辑系统的语义可用一个二元组来表示，即 $\mathscr{U}=(U,I)$，这里 U 为解释域，I 为一解释，可以分别定义如下：设 $U_{i,0}$是类型 $T_i(i=1,\cdots,n)$所有简单的自由变量的项的集合，这里 $U_{i,0}$可能有无限个，U_i 为类型 T_i 的变量取值域，可定义如下：

$U_{i,k}=U_{i,k-1}\cup\{f(t_1,\cdots,t_n),|t_j\in U_{j,k-1}(j=1,\cdots,n)$，其中 f 为函数，f 的结果类型为 T_i)

$U_i=\bigcup_{k=0}^{\infty}U_{i,k}$

$U=\bigcup_{i=1}^{n}U_i$

上式 U 即所需的解释域，以上定义的 U 与传统的 Herbrand 域对应，但它是一个多类型的域。

给定一个 CO-logic 数据库，我们可以类似 Herbrand 解释给出解释 I 的定义如下，用 I(t)表示 t 的解释：

(1)U 为解释域；(2)类型 T_i 的常量(如 C)，被解释为它自己，即 I(C)=C；(3)如果 f 是一个 n 元函数，则 f 被解释为从 U^n 到 U 的映射；(4)如果 f 是一个 n 元操作，则同样被解释为从 U^n 到 U 的映射；(5)对一个 n 元关系 r，被解释为 U^n 空间的一个子集，当 n=1时，即为 U 的子集；(6)对任一类型 P，P 被解释为 W 空间的子集，这里 W 定义如下(设 S 为集合，令 P(s)表示该集合的幂集)：

$W_1=U\cup P(U),\cdots,W_{2n}=\bigcup_{i=1}^{\infty}W_{2n-1}^i$，

$W_{2n+1}=W_{2n}\cup P(W_{2n}),\cdots,W=\bigcup_{i=1}^{n}W_i$。

应该注意的是：这里对类名 P 的解释与传统一阶谓词逻辑对谓词的解释有较大的不同。传统逻辑的谓词变元(或演绎数据库中的属性)个数是固定的；而在我们的逻辑系统中，类格中一个子类的所有对象也属于其超类，即每一类中的对象可以有不同数目的属性，子类无条件地继承其超类的所有属性。我们对类名 P 作的解释正好体现了这一特点。

一个变量指派 V 是从对象变量到对应类型的变量域的映射，我们将 V 扩充到项，以 V' 表示，其条件是：(1)V'(d)=V(d)，d 为常量，(2)V'(d)=V(d)，d 为变量，(3)V'(f(…,t,…))=I(f)(…,V(t'),…)，f 为函数符(或操作符)，t 为项。

我们说一组变量的指派 V 满足约束关系 $C(a_1,a_2,\cdots,a_n)$，如果向量$(V'(a_1),V'(a_2),\cdots,V'(a_n))$满足关系 C，即 $C(V'(a_1),\cdots,V'(a_n))$成立。

维普资讯 http://www.cqvip.com

设S为一约束系统.若指派V满足所有约束关系,则称V为S的一个解.两个约束系统有同样的一组解,则称它们是等价的.应该注意的是,所有可解的约束系统对变量的空集是等价的,所有无解系统也是等价的.所谓可解系统,是指这种系统最少有一个解。

我们也可以按下式将变量指派V扩充到项:

$V(P:p[\cdots,t,\cdots])=V(p)[\cdots,V(t),\cdots]$

一个U-基例是一个形式为$P:p[t_1,\cdots,t_n]$的对象,式中p和所有项$t_i(i=1,\cdots,n)$属于U。所有U-基例构成CO-logic的解释基(类似于传统的Herbrand基)。

给定一解释I和一变量指派V,可以定义如下函数$M_{I,V}$,赋予每一复合项的意义:对任一简单复合项,$P:p$,$M_{I,V}(P:p)$为真,当且仅当$V(p)\in I(p)$.对任一复杂复合项,$t=P:p[t_1,\cdots,t_n]$,$M(t)$为真.当且仅当(1)$M(t_i)$为真,$i=1,\cdots,n$;(2)存在$K,K\subseteq I(p)$,并且$K=(K_0,K_1,\cdots,K_n)$,使得$\langle V(p),V(t_1),\cdots,V(t_n)\rangle\leq_* K$,即必需满足下述条件:(a)$K_0=V(p)$;(b)如果$t_i(i=i,\cdots,n)$是一简单复合项,则$K_i=V(t_i)$;(c)如果$t_i(i=1,\cdots,n)$是一复杂复合项,则$(V(t_i))\leq_*(K_i)$;(d)如果$t_i$是复杂复合项的集合,则$\{V(t_i)\}\subseteq_*\{K_*\}$.这里$\subseteq_*$定义如下:对任一元素$s\in\{V(t_*)\}$,存在另一元素$s'$,$s'\in\{K_i\}$,使得$s\leq_* s'$。

对任一规则$r:t_0:-t_1,\cdots,t_n,c_1,\cdots,c_m$,$M_{I,V}(r)$为真,当且仅当$M_{I,V}(t_i)(i=1,\cdots n)$为真,并且v为$\{C_i\}(j=1,\cdots,m)$的一个解时,意味着$M_{I,V}(t_0)$也为真。

根据上述规定,显然一个基复合项的意义与变量指派无关,我们可以将它记为$M_I(t)$.如果$M_I(t)$为真,则解释I是t的模型,如果规则集R中每一规则在I解释下为真,则称I为规则集R的模型。

象传统逻辑一样,在我们的系统中,可以将解释表作U-基例的子集。例如有规则集R如下:$S:s[N:3,N:1]$,$P:[N:2]$,$P:x:-S:x$.则集合$\{N:1,N:2,N:3,S:s[N:3,N:1],P:p[N:2],P:s[N:3,N:1],S:s,P:p,P:s\}$为R的一个模型。

从以上所述CO-logic系统的语法和语义结构可看出,这系统比传统一阶逻辑有了较大的扩充,.在文[1]中,我们举出若干示例,说明了基于传统一阶逻辑的关系模型的谓词形式很容易映射到本系统的复合形式。同时,在CAD中一些用现有数据库工具难于表示的关系,可以较容易用CO-logic系统进行形式描述.另外一些例子表明本系统较之LDL语言、COL语言和FLogic有较强的语义表达能力和清析、简便的语法结构.

三、CO-LOGIC模型的理论基础

作为演绎系统最重要的是它的理论基础.传统演绎理论的支柱是最小模型的唯一性、模型相交定理、最小模型的极小不动点性质以及完备合理的过程特性.

3.1 最小模型

为了给出"最小"概念.我们在U-基例空间定义一个关系$\leq_b$如下:对任意两个基例t和t',令$t=C_1:c_1[t_1,\cdots,t_n]$.和$t'=C_2:c_2[t_1',\cdots,t'_m]$,我们说$t\leq_b t'$,当且仅当下列条件成立:(1)$c_1=c_2$;(2)$c_1\leq_b c_2$;(3)$n=0$,或$n=m$,此时对每一$i(i=1,\cdots,n)$有:(a)如果$t_i$和$t_i'$是元组类型,则$t_i\leq_b t_i'$;(b)如果$t_i$和$t_i'$是集合类型,则$\{t_i\}\subseteq_b\{t_i'\}$,这里的$\subseteq_b$定义是:对任意元素$s\in\{t_i\}$,存在一元素$s'\in\{t_i'\}$,使得$s\leq_b s'$。

对任意两个解释I和I',我们说$I\leq I'$当且仅当对每一元素$a_1\in I$,都存在一个元素$a_2\in I'$使得$a_1\leq_b a_2$,我们说$I=I'$,当且仅当$I\leq I'$并且$I'\leq I$。我们称模型I是最小的,如果不存在另一模型J,使得$J\leq I$,但使$I\leq J$不存在。

假设I_1和I_2都是模型,我们定义I_1和I_2的交是这样的集合,即它包括所有在I_1和I_2下均为真的那些U-基例。据此,容易证明下列定理(参见文[1]):

定理1 如果D是CO-logic系统的一个数据库,则D有唯一的最小模型。

定理2 如果D是CO-logic系统的一个数据库,I_1和I_2是D的模型,则两者的交也是D模型。

3.2 不动点语义

为了建立不动点语义,必需对CO-logic的每个数据库D,用传统逻辑重定义函数T如下:

$T(s)=\{d\in P(\text{U-基例空间}):$在D中存在一规则:

$t_0:-t_1,\cdots,t_n,c_1,\cdots,c_m$,和一指派V使得:(1)$V(t)=d$;(2)V是$\{c_i\}$,$i=1,\cdots,m$,的一个解;(3)$(V(t_j))\subseteq_b s$,$j=1,\cdots,n\}$。

从定义中可以看出$T(s)$函数是基于约束系统的可解性,而不是基于U-基例空间的可台一性,这一点

维普资讯 http://www.cqvip.com

同传统逻辑有较大不同。

在传统逻辑中，存在众所周知的最小模型的极小不动点性质，即 $M_p=lft(T_p)=T_p \uparrow w$。从文[5]，可以证明，在一约束系统中，对一新的约束域 D，这性质仍是正确的，只要下列条件能满足：(1)D 是完全可满足的，即每一约束或证明是可解的，或证明是不可解的。(2)D 有紧凑解，即域中每一元素能用若干个(可能是无穷个)约束来说明，任一约束的补也能用若干(可能无穷个)约束来说明。

3.3 过程语义

我们已知，复合项可以嵌套在另外复合项中，并可嵌套到任一深度，实际上，每一复合项等效于其它复合项的平合取，例如，$P_1p[S_1s[\{N_1:1,N_1:2\}],N_1:3]$ 等效于 $P_1p[S_1:s,N_1:3]$ 和 $S_1s[\{N_1:1,N_1:2\}]$。

给定具有变量集 $V=\bigcup_{i=1}^{n}V_i$(所有 V_i 对应于类型 T_i 的变量)和项的数据库，则代换 θ 是一映射：$V\rightarrow$ {类型 T_i 的项}。照常，我们可以作如下扩展，即令 θ 与函子及类名置换，即若 $t=P_1p[t_1,\cdots,t_n]$，则 $\theta(t)=P_1\theta(p)[\theta(t_1),\cdots,\theta(t_n)]$。

我们能简单地将次序关系 $\leq_b$ 扩展到非基本复合项：给定一对复合项(可能是非基本的)t_1 和 t_2，当且仅当对应于 t_1 和 t_2 的每一基本置换，$\theta(t_1)\leq_b\theta(t_2)$。当且仅当 $\theta(t_1)\leq_t\theta(t_2)$ 时，置换 θ 是 t_1 和 t_2 的一种合一。若对 t_1 和 t_2 的每一合一 θ_1，存在合一 θ_2 使得 $\theta=\theta_2\theta$，则这合一是最一般的合一子(mgu)。

CO-logic 演绎系统由两部分组成，即消解推理规则和约束求解器。CO-logic 子句隐含地是指复合项，或其非，或约束等的全称量化的析取(有时子句表作集合形式)。消解规则是利用最一般合一子来求以正复合项消解负复合项，这时假设约束是满足的。

一个演绎序列由一些有可解约束的目标所组成，当最后目标仅包含约束时，演绎序列成功。这些响应的约束构成查询的结果。当序列的最后目标不能再扩展时，认为是有限地失败。

对一确定的 CO-logic 数据库 D 和一查询 Q 来说，当3.2小节中条件满足时，我们可得到下列结果(证明见文[1])：

定理3 (CO-logic 演绎的合理性) 令 D 为 CO-logic 数据库，Q 为查询，若 $D\vdash Q$，则 $D\models Q$

定理4 (CO-logic)演绎的完备性) 令 D 为 CO-logic 数据库，Q 为查询，若 $D\cup\{Q\}$ 是不可满足的，则存在一个成功的演绎推理序列。

四、结 论

本文提出一种多类型逻辑，在统一的一阶逻辑的形式框架下实现了面向对象、演绎推理和约束处理的有机结合，并保证了模型具有完备合理的形式语义，CO-logic 系统对传统一阶逻辑作了较大的扩充，它不仅具有面向对象的自然的数据表示、演绎模型的知识推理等特点，还具备显式的约束处理能力，作为演绎系统的理论基础，本文为系统建立一整套语义理论，包括最小模型、不动点语义和过程语义等。有关继承问题，我们在系统中通过引入类关系演绎机制来实现继承同逻辑相结合。多类型系统的另一特点是在演绎推理过程中，通过类型检查可排除大量的不必要的消解过程。本文着重研究面向对象，演绎推理和约束处理的结合，进一步工作将在以下方面进行：如何使模型中允许出现否定文字，如何更多吸收知识工程和人工智能的思想，以及如何进一步融入其它的面向对象概念，如封装性和复载等。

主要参考文献

[1]Xiuhua Li, The Design and Implementation of CDOODL (Constraint Deductive Object-Oriented Database Language). Ph. D. thesis, Hefei University of Technology 1993

[2]M. Kifer and G. Lausan, F-logic: A Higher-Order Logic for Reasoning about Object, Inheritance, and Schema, Proc. ACM SIGMOD(1989)

[3]M. Kifer and J. Wu, A Logic for Object-Oriented Logic Programming, Proc. ACM, PODS(1989)

[4]J. Cohen, Constraint Logic Programming Language, CACM, Vol. 33, No. 7, 1990

[5]J. Jaffar et al., Constraint Logic programming, Proc. 14th ACM POPL 1987