知识库系统的逻辑基础

许文艳[1)] 刘三阳[2)]
(西安电子科技大学理学院应用数学系 西安 710071)

摘 要 针对知识库系统的研究现状和存在的问题,分析了用经典一阶谓词逻辑作为知识库系统逻辑基础的不充分性,根据知识库系统的特点和需求,指出可计算性逻辑既能为知识库系统提供合理的逻辑基础,又能弥补经典逻辑作为知识库系统逻辑基础的不足.并在此基础上用证明论方法,建立了一个基于可计算性逻辑的完备子集 CL4 的知识库系统的公理系统,显示出易于表达、操作简单的特点.

关键词 可计算性逻辑;知识库系统;博弈;交互;资源
中图法分类号 TP301 **DOI 号**: 10.3724/SP.J.1016.2009.02123

Logic for Knowledgebase Systems

XU Wen-Yan[1)] LIU San-Yang[2)]
(*Department of Mathematic, Institute of Sciences, Xidian University, Xi'an* 710071)

Abstract According to the features and requirements of knowledgebase systems, the authors discuss the insufficiency of basing knowledgebase systems on classical first-order logic, and analyze the rationality and sufficiency of basing them on computability logic. The authors apply the method of proof theory and construct an axioms system of knowledgebase system, which is based on a complete subset CL4 of computability logic and shows the simplicity of knowledge representations and convenience of operating..

Keywords computability logic; knowledgebase system; game; interactive; resources

1 引 言

作为人工智能和数据库两项技术结合的产物,知识库系统已越来越受到人们的关注.它是利用人们已有的各种知识进行知识获取、组织和检索查询的计算机信息系统。在研究知识库系统时,其中很重要的问题是知识库的逻辑基础.传统的演绎知识库系统是以一阶谓词逻辑为基础的[1].但谓词逻辑表达的知识主要是静态的、浅层的知识,它不易表达交互过程和建设性知识,因此以一阶谓词逻辑作为知识库系统的逻辑基础是不充分的(本文将在第 3 节详细讨论).目前研究很多的基于描述逻辑的知识库,其逻辑基础——描述逻辑[2]——也只是一阶谓词逻辑的一个可判定子集,所以以它作为知识库系统的逻辑基础仍然是不充分的.另一方面,为了弥补经典逻辑表达能力的不足,许多学者从认知角度考虑,在其基础上添加模态操作[3-5],例如引入"know that"算子"□",但这种构造的结果是产生不必要的表达混乱和最终系统的非半可判定性.

2003 年,Japaridze 提出了"可计算性逻辑"[6](Computability Logic,CL).正如经典逻辑是关于真理的形式理论一样,可计算性逻辑是关于可计算性的形式理论,是一种交互的资源逻辑.从语义上说 CL 采取"博弈"语义,一个计算问题被理解为是计算机和用户之间的一种交互的"博弈"问题.问题的

收稿日期:2008-04-14;最终修改稿收到日期:2009-04-20.本课题得到国家自然科学基金(60574075,60674108)资助.**许文艳**,女,1978 年生,博士研究生,讲师,主要研究方向为非经典逻辑与近似推理. E-mail: xwykiki@126.com. **刘三阳**,男,1959 年生,教授,博士生导师,主要研究领域为现代优化理论与方法.

可计算性意味着存在一个机器总能赢得这个博弈，逻辑运算表示在计算性问题上的操作，一个逻辑公式的有效性表示它是“总是可计算的”问题. 值得一提的是 CL 具有很强的表达能力(本文将在第 3 节详细说明)，是经典逻辑的保守扩张. 经典逻辑只是 CL 的特殊部分——零交互问题部分. 另外文献[7-10]先后证明了 CL 的特殊部分 CL1、CL2、CL3、CL4 的可靠性和完备性，为建立基于可计算性逻辑的应用系统提供了理论基础.

基于上述原因，本文先在第 2 节简要介绍 CL 的语义和语构理论；然后在第 3 节分析知识库系统的特点和对其逻辑基础的需求，指出 CL 既能为知识库系统提供合理的逻辑基础，又能弥补传统逻辑作为知识库系统逻辑基础的不足；最后在此基础上，第 4 节用证明论方法建立了一个基于 CL 的完备子集 CL4 的知识库系统公理系统，显示出易于表达，操作简单的特点.

2 可计算性逻辑 CL

事实上，确切地说可计算性逻辑不是一种特定的理论体系，它具有开放的形式化体系，是旨在实现“从真理到可计算性”的一项伟大计划. 它完全包含并扩张了经典一阶谓词逻辑，在语义上采取“博弈”语义，是较经典逻辑“真值”语义的一次质的突破.

2.1 CL 的语法

目前为止 CL 体系中有以下符号：

(1) 个体常量符：$a,b,c,\cdots$；

(2) 个体变量符：$x,y,z,\cdots$；

(3) 逻辑常量符：$\top,\bot$；

(4) 简单谓词符：$p,q,r,s,\cdots$；

(5) 一般谓词符：$P,Q,R,S,\cdots$；

(6) 联结词符：$\neg,\vee,\wedge,\rightarrow,\sqcap,\sqcup,\Downarrow,\Uparrow,\Rightarrow$；

(7) 量词符：$\exists,\forall,\prod,\coprod,\bigvee,\bigwedge$；

(8) 括号及逗号：(,),“,”.

其中逻辑常量“$\top$”表示“总是可计算的”简单问题，“$\bot$”表示“总是不可计算的”简单问题. 符号 $\neg,\vee,\wedge,\rightarrow,\exists,\forall$，表示的运算和含义与经典逻辑的含义相同. 算子 $\vee,\wedge$ 称为平行联结词，$\bigvee,\bigwedge$ 称为平行量词，$\sqcap,\sqcup$ 称为选择联结词，$\prod,\coprod$ 称为选择量词，$\Downarrow,\Uparrow,\Rightarrow$ 称为重复联结词，且 $P\Rightarrow Q=(\Downarrow P)\rightarrow Q$. 选择联结词和选择量词统称为选择算子.

CL 要比经典逻辑更富表达力不仅因为它增加了许多新的逻辑运算符，可表达更细致的知识，而且因为它有两种谓词符：简单谓词符和一般谓词符. 其中简单谓词符代表零交互的问题，因而等同于经典逻辑中的谓词符，而一般谓词符代表一般(不一定零交互)的问题，因而是新引入的符号.

项(term). 个体常量和个体变量都是项. 记为 $t_1,t_2,\cdots,t_n$.

原子公式(atom). (1) 逻辑常量 $\top,\bot$ 是原子公式，称为逻辑原子公式；(2) 设 L 是 n 元简单或一般谓词符，t_i 是项($i=1,2,\cdots,n$)，则分别称 $L(t_1,t_2,\cdots,t_n)$ 为简单或一般(非逻辑)原子公式.

公式(formula). (1) 原子公式 $L(t_1,t_2,\cdots,t_n)$ 是公式；(2) 若 A,B 是公式，则 $\neg A,A\wedge B,A\vee B,A\rightarrow B,A\sqcap B,A\sqcup B,\Downarrow A,\Uparrow A,A\Rightarrow B$ 也是公式；(3) 若 A 是公式，x 是变量，则 $\exists xA,\forall xA,\prod xA,\coprod xA,\bigvee xA,\bigwedge xA$ 也是公式.

2.2 CL 的语义

首先我们需要解释以下几个基本概念：

(1) 博弈语义. 现实生活中，用户的大部分计算任务是用户与计算机之间错综复杂的多输入输出的交互过程，而非只有一个输入和一个输出的简单过程：计算任务的解决意味着计算机最终赢得了这个博弈，即输出了正确答案. 因此，传统的将计算问题理解为函数的观念是狭隘的，而应将其理解为“博弈”. 此后我们将计算问题等同于博弈问题.

图 1 用户与计算机之间的交互

(2) 静态博弈. 所谓静态博弈是指博弈的最终结果与博弈双方的反应速度无关. 若博弈的一方有一成功的策略，则该策略无论对方反应速度有多快仍然是好的、有效的；若没有成功的策略，则他不论对方反应速度多慢也无法成功. 即静态博弈只关心策略而不关心相对速度. 否则与双方反应速度有关的博弈称为动态博弈. CL 所研究的博弈都是静态博弈，而非动态博弈.

(3) HPM. CL 交互计算的基本计算机模型是 HPM(Hard-Play Machines). 它是在一种图灵机(TM)模型基础上经过修改得到的，如图 2 所示，其中赋值带通过列出对所有变量 $v_0,v_1,v_2,\cdots$ 的赋值来拼写某个赋值. 例如图 2 中的赋值带拼写赋值 e：$e(v_0)=20,e(v_1)=3,e(v_2)=62,e(v_i)=0(i\geq 3)$. 赋值带上的内容在 HPM 工作的整个过程中是不变的，

因此是“静态输入”.行程带通过列出计算双方(机器与用户)的行动值拼写计算问题的当前位置.如图 2 中行程带拼写的当前位置是:〈**400**,0,**1**〉.黑体的行动值由机器做出,非黑体的行动值由用户做出.

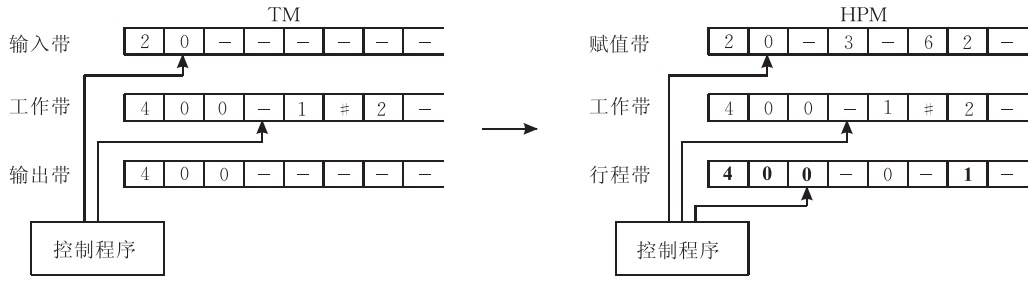

图 2　TM 与 HPM 之间的对比

HPM 的工作过程:最初,机器处于“开始”状态,赋值带拼写某个赋值 e,工作带和行程带空白,3 个扫描头均处于各带的最左端;计算开始后,机器由一个状态变到另一个状态,根据控制程序在工作带上拼写并移动扫描头;当机器处于“行动”状态时,工作带扫描头左边的字符串 α 被自动以黑体追加到行程带上,这意味着机器做出了行动 α;在每步计算期间,任何有限的非黑体字符串 $\beta_1,\cdots,\beta_n$ 都可被追加到行程带上,这意味着用户做出了行动 $\beta_1,\cdots,\beta_n$.这里不限制用户行动个数的设计正说明了我们不在用户的相对速度上强加任何条件的直觉.

设 A 为一个计算问题,$\mathcal{M}$为一个 HPM.若对任意的赋值 e,不论用户如何行动,$\mathcal{M}$都能对赋值 e 下的计算问题 $e[A]$最终输出正确答案,则称$\mathcal{M}$完成 A,或$\mathcal{M}$计算 A.

定义 1.　称函数 $*$ 为解释,若 $*$ 满足以下条件:

(1) $*$ 把每个 n 元简单谓词 p 映射到简单计算问题 $p^*(x_1,x_2,\cdots,x_n)$;

(2) $*$ 把每个 n 元一般谓词 P 映射到一般计算问题 $P^*(x_1,x_2,\cdots,x_n)$.

每个解释 $*$ 均可按经典的方法扩充到所有公式:$(L(t_1,t_2,\cdots,t_n))^*=L^*(t_1,t_2,\cdots,t_n)$,$(\neg A)^*=\neg(A^*)$,$(A\sqcap B)^*=A^*\sqcap B^*$,$(\prod xA)^*=\prod x(A^*)$等.通俗地说,解释 $*$ 给公式赋予了“意思”:一个公式 F 只是一串符号,而 F^* 是一个计算(博弈)问题.

定义 2.　一个计算问题 A 是可完成的(或可计算的)当且仅当存在一个 HPM $\mathcal{M}$完成 A.这样的 $\mathcal{M}$称为 A 的一个解决方案,记为$\mathcal{M}\models A$.

定义 3.　一个公式 F 是有效的当且仅当对 F 的任意解释 $*$,F^* 是可完成的.

进一步 CL 有比“有效性”更强的“统一有效性”.

定义 4.　一个公式 F 是统一有效的当且仅当存在一个 HPM $\mathcal{M}$,使得对 F 的任意解释 $*$,$\mathcal{M}\models F^*$.这样的$\mathcal{M}$称为 F 的一个“统一解决方案”.

公式 F 的统一解决方案$\mathcal{M}$是一个不依赖于 F 的解释而能保证完成 F 的策略.

2.3　CL 的语构

这里我们介绍可计算性逻辑的完备子集 CL4 的语构理论.因为它是目前为止 CL 的表达力最强的完备的推理系统.

CL4 系统的语法符号是在 2.1 节介绍的符号中去掉 $\vee$,$\wedge$ 及 $\Downarrow$,$\Uparrow$,$\Rightarrow$,其余符号均保留,原子公式仍分为简单和一般两种.一个子公式在偶数个(奇数个)$\neg$ 的辖域中的出现称为正出现(负出现).子公式的不在任何选择算子中的出现称为表面出现.不含一般原子和选择算子的公式——即经典一阶谓词逻辑中的公式——称为简单公式.一个公式 F 的“简化公式”是将 F 中所有形如 $G_1\sqcup G_2$ 或$\coprod xG$ 的子公式的表面出现用 $\bot$ 代换,所有形如 $G_1\sqcap G_2$ 或$\prod xG$ 的子公式的表面出现用 $\top$ 代换,所有一般原子的正表面出现用 $\bot$ 代换,所有一般原子的负表面出现用 $\top$ 代换.一个 CL4 公式 F 是稳定的当且仅当 F 的简化公式是经典逻辑中的重言式,即在经典逻辑中是可证的.否则 F 是不稳定的.

CL4 系统有以下 4 条规则:

A 规则. $\vec{H}\mapsto E$,其中 E 是稳定的且$\vec{H}$是一个公式集满足:

(1)若 E 中有子公式 $G_1\sqcap G_2$ $(G_1\sqcup G_2)$的一个正(负)表面出现,则对每个 $i\in\{1,2\}$,$\vec{H}$包含将 E 中该出现用 G_i 代换后的公式;

(2)若 E 中有子公式$\prod xG$($\coprod xG$)的一个正(负)

表面出现，则$\vec{H}$包含将E中该出现用$G(y)$代换后的公式，其中y是没在E中出现过的某个变量.

B_1 规则. $H \mapsto E$，其中H是将E中子公式$G_1 \sqcap G_2$($G_1 \sqcup G_2$)的一个负(正)的表面出现用G_i($i=1$或2)代换后的公式.

B_2 规则. $H \mapsto E$，其中H是将E中子公式$\prod xG$($\coprod xG$)的一个负(正)的表面出现用$G(t)$代换后的公式. 这里项t不在E中约束出现.

C 规则. $H \mapsto E$，其中H是将E中某个n元一般原子公式的两个——一正一负——表面出现分别同时用一个不在E中出现的n元简单原子公式代换后的公式.

当A规则中的$\vec{H}$是空集{}时，E是稳定的且不含$\sqcap$，$\sqcup$和$\prod$，$\coprod$，从而E是经典逻辑中的定理，此即CL4中的公理. 若公式F可由公理和4条推理规则推出，则F是可证的，记作$CL4 \vdash F$.

2.4 可靠性与完备性定理

文献[10]中已证明了：

CL4的可靠性完备性定理. $CL4 \vdash F$当且仅当F是有效的.

CL4的统一可靠性定理. 若$CL4 \vdash F$，则F是统一有效的，并且存在一个有效的程序可根据F的CL4证明找到F的一个统一解决方案.

CL4的强完备性定理，若$CL4 \nvdash F$，则对某个解释$*$，F^*是不可完成的.

定理 1. CL4中不含$\forall$，$\exists$的公式的有效性，统一有效性是可判定的.

3 可计算性逻辑CL作为知识库系统逻辑基础的优越性分析

可计算性逻辑本是关于“什么是可计算的”的形式理论. 但CL4的统一可靠性使得它不仅成为“什么是可计算的”逻辑，而且成为“如何计算”的逻辑. 也就是说CL可看作是一种“解决问题的工具”. 因此它有望成为诸多应用领域的理论基础，特别是可作为知识库系统的逻辑基础. 具体分析如下.

(1)作为知识库系统数学基础的逻辑应该是“建设性的”.

作为知识库系统数学基础的逻辑应该能够区分：问题的“真”和“能找到/辨别真”的能力. 例如，知识“任何人都有母亲”可表示为公式：

$$\forall x \exists y(y=Mother(x)) \quad (1)$$

这是非建设性知识，是一般人普遍具有的知识. 而建设性知识，例如以下公式：

$$\prod x \coprod y(y=Mother(x)) \quad (2)$$

表示“潜在的确实知道任何一个人的母亲是谁”的知识，这是一般人所不具备的知识.

但经典逻辑却没有区分这两种知识的能力，这使得以它作为知识库系统的逻辑基础是不充分的. 而CL却能准确区分这两种知识.

(2)作为知识库系统数学基础的逻辑应该是“交互性的”.

大多数知识库和信息系统是交互的. 但经典逻辑不是交互的，它是关于真理的逻辑，是零交互的. 而CL却能表达交互的过程，从而很好地满足了知识库系统的要求.

例如，我们想让系统做的是：对任意输入的一个病人x，输出诊断结果y，则系统需解决的问题可表示为

$$\prod x(Q(x) \rightarrow \coprod y(y=Diagnosis(x))) \quad (3)$$

其中$Q(x)$为关于x的病理征兆、血压、药理反应等表现的综合，可用“$\wedge$”把它们联结，或是更为复杂的结构，其中计算机提出什么问题可能依赖于用户提供的对先前问题的回答，从而产生一个交互的人机对话的过程.

例如式(3)中$Q(x)$为

$$\coprod sSmpt(x,s) \wedge \prod t(Pst(x,t) \sqcup \neg Pst(x,t)) \quad (4)$$

其中，$Smpt(x,s)$表示“病人x有征兆s”；$Pst(x,t)$表示“病人x做测试t的值为正”. 以下是该问题可能的一个交互对话过程：

1. 最初，系统处于等待用户输入一个病人名字的状态. 当用户输入一个病人名字X后，该问题演变为

$$\coprod sSmpt(X,s) \wedge \prod t(Pst(X,t) \sqcup \neg Pst(X,t)) \rightarrow \coprod y(y=Diagnosis(X)).$$

2. 系统继续处于等待状态，直到用户输入病人X的征兆S：

$$Smpt(X,S) \wedge \prod t(Pst(X,t) \sqcup \neg Pst(X,t)) \rightarrow \coprod y(y=Diagnosis(X)).$$

3. 基于用户提供的信息，系统选择测试T(要求对病人X做测试T)：

$$Smpt(X,S) \wedge (Pst(X,T) \sqcup \neg Pst(X,T)) \rightarrow \coprod y(y=Diagnosis(X)).$$

4. 用户对病人X进行测试T，并且反馈给系统测试结果的值为正：

$$Smpt(X,S) \wedge Pst(X,T) \rightarrow \coprod y(y=Diagnosis(X)).$$

5. 基于上述过程，系统做出判断：病人X的诊断结果为Y：

$$Smpt(X,S) \wedge Pst(X,T) \rightarrow (Y=Diagnosis(X)).$$

(3)作为知识库系统数学基础的逻辑应该是有“资源意识”的.

知识库中的事实与规则也可理解为是一种资源.事实资源(知识),如“王菲是女生”,“张三是学生”,“李刚是李磊的父亲”等,可分别表示为公式:

$$Female(wang\,fei),\ Student(zhangsan),$$
$$Ligang=Father(Lilei)$$

规则资源,表示一般知识,如“任何人都有母亲”,“若一个人是某人的母亲则她一定是女性”等,可表示为

$$\forall x\,\exists y(y=Mother(x)),$$
$$\forall x(\exists y(x=Mother(y))\rightarrow Female(x)).$$

上述例子是经典逻辑就能表示的资源(知识).但知识库的事实和规则资源中可能还会有这样的资源,如拥有一个一次性的测孕工具相当于拥有用下式表示的资源:

$$\prod x(Pregnant(x)\sqcup\neg Pregnant(x)) \tag{5}$$

而拥有两个一次性测孕工具相当于拥有资源:

$$\prod x(Pregnant(x)\sqcup\neg Pregnant(x))\wedge$$
$$\prod x(Pregnant(x)\sqcup\neg Pregnant(x)) \tag{6}$$

这两种资源显然是不一样的.但传统的知识库系统由于建立在经典逻辑基础上,不区分公式 P 和 $P\wedge P$,从而没有能力看到式(5)和(6)所表示资源的重要差异,这是传统知识库的一大缺点.

而基于 CL 的知识库系统,对可重复使用的资源可在其公式前加重复算子“$\Downarrow$”,如公式 $\Downarrow\prod x\coprod y(y=x^2)$,表示系统可重复计算任意一个数平方的能力;对消耗性资源,如上面提到的 $\prod x(Pregnant(x)\sqcup\neg Pregnant(x))$(不可重复使用),则其前不加算子“$\Downarrow$”.

(4)知识库系统应该解决的是“统一有效的”问题而不是仅仅“有效的”问题.

知识库系统作为知识查询系统,若在用户界面要查询(解决)问题 Q,事实上,系统真正要解决的是 $KB\rightarrow Q$.其中 KB 是系统的“知识库”,即知识库中知识的有限集合.形式上,KB 就是一个有限的公式集,公式间的关系为“$\wedge$”,即 KB 可认为是形如 $P_1\wedge P_2\wedge\cdots\wedge P_n$ 的公式.而知识库系统,确切地说,是建立在纯逻辑基础上的全局有效的软件,它不含任何非逻辑的知识.所以运用系统软件解决的应该是而且只能是“统一有效的”问题.

现设 $KB=R_1\wedge\cdots\wedge R_n$,则系统要解决的是 $R_1\wedge\cdots\wedge R_n\rightarrow Q$,即 $\neg R_1\vee\cdots\vee\neg R_n\vee Q$.所以系统不仅在 Q 中与用户交互,而且同时平行地在 $\neg R_1,\cdots,\neg R_n$ 中与系统的信息提供者(provider)交互(信息提供者可以是系统中的一个程序,可以是一个网络服务器,还可以是一组服务于系统的专业人员).只要信息提供者成功地完成他们的工作,系统就在 $\neg R_1,\cdots,\neg R_n$ 中失败.这意味着系统赢得整个博弈“$\neg R_1\vee\cdots\vee\neg R_n\vee Q$”当且仅当系统在 Q 中取胜,即成功解决 Q.因此,系统解决问题 Q 的能力事实上就是产生一个解决 $KB\rightarrow Q$ 的办法的能力.而这种能力可由其逻辑基础 CL,尤其是 CL 的统一可靠性定理给出.根据 CL 的统一可靠性定理,系统只要能找到一个 $KB\rightarrow Q$ 的证明,就能有效地构造一个统一解决方案 $\mathcal{M}$,使得 $KB\rightarrow Q$ 被完成.

注意是“统一可靠性”而不是简单的“可靠性”使得系统工作.简单的“可靠性”只能保证可证的公式是有效的,这是不充分的,因为:

(1)公式 E 的有效性意味着对任意解释 $*$,存在一个 E^* 的解决方案.很可能不同的解释相应有不同的解决方案,从而选择正确的解决方案需要具体解释的具体知识,即 E 的“含义”.而我们的假设是系统没有任何非逻辑的知识,或者说系统根本不知道解释 $*$.因此,系统若最终产生了一个 E^* 的解决方案,事实上,该方案对任何解释都是成功的,即它是 E 的一个统一解决方案.

(2)另一原因是,在找到 E 的一个证明后,简单的可靠性能保证对 E^*“存在”一个解决方案,但它可能没办法确实找到这个方案.而统一可靠性不仅能保证可证的公式必存在解决方案,而且能有效地“找到”这个方案.

综合以上 4 点,我们认为 CL 既能为知识库系统提供合理的逻辑基础,又能弥补传统逻辑作为知识库系统逻辑基础的不足.

4　基于 CL4 的知识库系统

以下我们用证明论方法建立一个基于 CL4 的知识库系统的公理系统.

我们用 $Alkali(x)$ 表示“溶液 x 是碱性的”,用 $Blue(x)$ 表示“石蕊试纸在溶液 x 中变蓝”,则拥有一张石蕊试纸,检测溶液酸碱性系统的公理系统建立如下:

(1)建立一组公理,即建立基于 CL4 的知识库系统的知识库 KB.它包含以下两个公理:

$$\forall x(Blue(x)\leftrightarrow Alkali(x)),$$
$$\prod x(Blue(x)\sqcup\neg Blue(x)),$$

其中第 1 个公式表示知识:溶液是碱性当且仅当石蕊试纸在其中变蓝;第 2 个公式表示知识(资源):系统拥有一张石蕊试纸,信息提供者可通过将试纸放

入任何溶液中而报告给系统试纸的颜色.

(2)建立一组推理规则.这里规则为 CL4 系统的 4 条规则:A 规则,B_1 规则,B_2 规则,C 规则(这里不再重复).

(3)由公理与推理规则最终得到定理(查询结果).

假设用户要查询:系统是否对任何溶液都能判断出它是碱性的还是非碱性的.该问题可表示为

$$\prod x(Alkali(x)\sqcup\neg Alkali(x)).$$

特别注意该问题绝不能表示为经典逻辑公式:$\forall x(Alkali(x)\vee\neg Alkali(x))$(因为这个公式表示"任何溶液是碱性的或者不是碱性的").

由以下证明过程我们将看到:$CL4\vdash KB\rightarrow\prod x(Alkali(x)\sqcup\neg Alkali(x))$.

(1) $\forall x(Blue(x)\leftrightarrow Alkali(x))\wedge\neg Blue(y)\rightarrow\neg Alkali(y))$ 由{},A 规则

(2) $\forall x(Blue(x)\leftrightarrow Alkali(x))\wedge\neg Blue(y)\rightarrow(Alkali(y)\sqcup\neg Alkali(y))$ 由(1),B_1 规则

(3) $\forall x(Blue(x)\leftrightarrow Alkali(x))\wedge Blue(y)\rightarrow Alkali(y)$ 由{},A 规则

(4) $\forall x(Blue(x)\leftrightarrow Alkali(x))\wedge Blue(y)\rightarrow(Alkali(y)\sqcup\neg Alkali(y))$ 由(3),B_1 规则

(5) $\forall x(Blue(x)\leftrightarrow Alkali(x))\wedge(Blue(y)\sqcup\neg Blue(y))\rightarrow(Alkali(y)\sqcup\neg Alkali(y))$ 由(2),(4),A 规则

(6) $\forall x(Blue(x)\leftrightarrow Alkali(x))\wedge\prod x(Blue(x)\sqcup\neg Blue(x))\rightarrow(Alkali(y)\sqcup\neg Alkali(y))$ 由(5),B_2 规则

(7) $\forall x(Blue(x)\leftrightarrow Alkali(x))\wedge\prod x(Blue(x)\sqcup\neg Blue(x))\rightarrow\prod x(Alkali(x)\sqcup\neg Alkali(x))$ 由(6),A 规则

由上述公理系统我们看到,基于 CL4 的知识库系统在建立知识库公理系统时,具有可表达更细致的知识或资源,能查询系统解决问题的能力以及根据 4 条推理规则操作简单等优点.

5 结 论

本文的主要贡献是深入分析了知识库系统的特点和它对其逻辑基础的需求,指出 CL 既能为知识库系统提供合理的逻辑基础,又能弥补传统逻辑作为知识库系统逻辑基础的不足.并在此基础上用证明论方法建立了一个基于 CL 的完备子集 CL4 的知识库系统公理系统,显示出表达知识细致准确,操作简单的优点.故 CL 有望代替经典逻辑作为知识库系统的逻辑基础.当然真正实现以 CL 作为知识库系统的逻辑基础,还有许多工作需进一步研究,如知识处理及搜索技术等.目前我们正考虑为以 CL4 为基础的知识库系统设计反向推理的推理机,使推理过程与人机交互过程合一,具体方法将另文讨论.

参 考 文 献

[1] Ullman J D. Principles of Database and Knowledge Base System. Vol. I, II. New York: Computer Science Press, 1988

[2] Baader F et al. The Description Logic Handbook: Theory, Implementation and Applications. Cambridge: Cambridge University Press, 2003

[3] Konolige K. On the relation between default and autoepistemic logic. Artificial Intelligence, 1988, 35(3): 343-382

[4] Levesque H, Lakemeyer G. The Logic of Knowledge Bases. Cambridge, MA: The MIT Press, 2000

[5] Moore R. A formal theory of knowledge and action//Formal Theories of Commonsense Worlds. Norwood, NJ: Ablex, 1985

[6] Japaridze G. Introduction to computability logic. Annals of Pure and Applied Logic, 2003, 123: 1-99

[7] Japaridze G. Propositional computability logic I. ACM Transactions on Computational Logic, 2006, 7(2): 302-330

[8] Japaridze G. Propositional computability logic II. ACM Transactions on Computational Logic, 2006, 7(2): 331-362

[9] Japaridze G. From truth to computability I. Theoretical Computer Science, 2006, 357(1): 100-135

[10] Japaridze G. From truth to computability II. Theoretical Computer Science, 2007, 379(1-2): 20-52

XU Wen-Yan, born in 1978, Ph. D. candidate, lecturer. Her research interests include classical logic, computability logic, and fuzzy reasoning.

LIU San-Yang, born in 1959, professor, Ph. D. supervisor. His research interests include optimization theory, non-smoothing analysis, and nonlinear analysis.

Background

As the result of combining artificial intelligence and data-base, knowledgebase systems have been one of the topics of most concern. When studying knowledgebase systems, one of considerably important problems is the logic on which the system is based. This paper focuses on this topic.

Traditional knowledgebase systems are based on classical first-order logic. But the language of first-order logic only expresses the static and shallow knowledge, not expressing interactive processes and constructive knowledge.

So it is not sufficient to base knowledgebase systems on first-order logic. The traditional approaches to knowledgebase systems try to mend this insufficiency by arguing the language of classical logic with special epistemic constructs, such as the modal "know that" operator □. But these epistemic constructs often yield unnecessary and very unpleasant complications such as messiness and non-semidecidability of the resulting logics.

Computability logic (CL), introduced recently, is a formal theory of computability. Computation and computational problems in CL are understood in their most general, interactive sense, and are seen as games played by a machine (computer) against its environment (user). Classical logic turns out to be a special fragment of CL. The classical concept of truth is nothing but a special case of computability—computability restricted to problems of zero interactivity degree.

This paper analyses the features of knowledgebase systems and the requirements of its logic, discusses the limitations of classical first-order logic and the superiorities of computability logic. Based on a system of computability logic, an axioms system of a knowledgebase system is constructed, which shows the simplicity and preciseness of knowledge representations, efficiency of queries to the system, and convenience of operating. The conclusion is that CL is an appealing alternative to traditional knowledge base. This paper apperceives it and makes an attempt on it. This research is supported by the National Natural Science Foundation of China (grant Nos. 60574075, 60674108).