第5卷第5期
1994年5月

软 件 学 报
JOURNAL OF SOFTWARE

Vol. 5, No. 5
May 1994

面向对象逻辑语言SCKE中的限制推理*

金 芝 胡守仁

(长沙工学院计算机系,长沙 410073)

摘要 限制理论是形式化常识知识并进行常识推理的一种重要方法.本文主要研究将限制理论转化为面向对象逻辑语言的可能性,并实现了完成这个转换的编译器.按面向对象逻辑语言的语义运行编译后的程序,可得到与原限制理论相同的结果.将该编译器嵌入面向对象逻辑语言解释器中,可以大大提高该语言的表达能力,特别是可以实现对常识知识表示和常识推理的支持.

关键词 面向对象的逻辑程序,限制理论,非单调推理,常识推理,继承系统.

限制理论(Circumscription)[1,4,6]由McCarthy于1980年首先提出,其最大贡献就是提供了一种形式化常识知识的手段和进行常识推理的方法.它的主要思想是利用"简单不规范理论"解决常识知识的表示和常识推理问题.所谓"不规范理论",是指任何个体事物都可能在某个方面是不规范的,引入谓词abnormal表示"不规范性",则可以通过最小化abnormal的外延达到进行常识推理的目的.非单调谓词abnormal是限制理论统一的表示基础.限制理论的另一个引人注目之处,在于它是建立在经典谓词逻辑基础之上的二阶理论,具备良好的数学理论基础.

面向对象逻辑语言SCKE(Structured Communicating Knowledge programming Entity)[2,3]是一种新颖的智能语言,它既具有逻辑语言语义清晰、表达能力强及自动推理能力等特点,又具有面向对象语言模块化结构、信息隐藏、继承以及过程和数据抽象等性质,对构造大型智能软件提供了良好的支持.在SCKE中,除了一般逻辑语言的一致化归约推理之外,继承是其又一种重要的推理方法.但一般面向对象语言都只提供了基于过程和数据抽象的继承机制,按继承的逻辑语义,这种继承推理是单调的.允许面向对象语言在继承过程中处理不规范问题,是提高它的表达能力,使它能用于形式化常识知识,进行常识推理的重要问题,也是使常识推理方法走向实用的一个重要手段.

本文主要研究限制理论与面向对象逻辑程序的关系,并实现了一个编译器完成限制理论到面向对象层次逻辑程序的转换.在SCKE中嵌入该编译器,可以使SCKE用于形式化基于限制理论的常识知识,并进行常识推理,从而提高了SCKE对常识的表达能力,也使限

* 本文1991-11-08收到,1992-01-30定稿

本课题受国家高技术计划资助.作者**金芝**,女,32岁,博士,主要研究领域为人工智能,逻辑程序设计,知识工程,智能机体系结构.**胡守仁**,68岁,教授,主要研究领域为计算机体系结构,人工智能.

本文通讯联系人:金芝,北京 100080,中国科学院数学研究所

制理论进一步走向实用.

1　限制理论基础

McCarthy 提出的限制理论作为一种非单调推理的形式化方法,受到了最充分的研究和重视. 直观上说,谓词限制是指从有穷子句出发,推导出的满足特定谓词的个体就是所有满足该谓词的个体. 对特定谓词的限制过程,实际上是一个最小化过程. 设 T 为一个理论,若 $P=\{p_1,\cdots,p_m\}$ 为需要最小化的谓词集,$Z=\{z_1,\cdots,z_n\}$ 为在最小化过程中按 P 发生变化的谓词,称为变量. 这里 $p_i,z_j(1\leqslant i\leqslant m,1\leqslant j\leqslant n)$ 都是 T 中的谓词符号,且 $p_i\neq z_j$. 则在 T 中对 P 的最小化过程将一阶理论 $T=T[P,Z]$ 转化为表达能力更强的二阶理论 CIRC(T;P;Z).

定义. 带有变量 Z 的理论 T,对最小化谓词 P 的限制 CIRC(T;P;Z)可定义为:

$$T(P,Z)\wedge\forall P',Z'(T(P',Z')\wedge(P'\rightarrow P)\rightarrow P'=P)$$

其中,$P'=\{p_1',\cdots,p_m'\}$,$Z'=\{z_1',\cdots,z_n'\}$ 分别为与 P 和 Z 类似的谓词,$P'\rightarrow P$ 表示对每一组变量 x 和所有 $1\leqslant i\leqslant m$,有 $p_i'(x)\rightarrow p_i(x)$.

该定义表明,在 T(P,Z)成立且 Z 中谓词的外延在最小化过程中可以变化的情况下,P 中谓词具有最小可能的外延.

例,设一个一阶理论 T 包括如下两个子句:

Bird(tweety).

Bird(tweety) ∧ Abnormal(tweety)→Fly(tweety).

其中,第一个子句表示 tweety 是鸟,第二个子句 tweety 只要没有出现任何异常情况,通常都会飞. 因为当前没有 tweety 的异常信息,按常识应该得出 tweety 会飞的结论,我们再看看限制推理的结论. 设 P={Abnormal}(我们打算最小化异常性),Z={Fly}(Fly 作为变量,它的外延依赖于谓词 Abnormal 的外延). 显然模型:

$$M=\{\text{Bird(tweety)},\text{Fly(tweety)}\}$$

为 T 的一个[P,Z]-最小模型且包括在 T 的任一个[P,Z]-最小模型中,则有:

$$\text{CIRC}(T;P;Z)\models\text{Fly(tweety)}$$

更进一步地,可以在谓词限制中引入序概念,以决定谓词最小化的次序,这种限制称作为优先化限制. 假设 P,Z 如上述定义相同,并将 P 划分为 $P_1,\cdots,P_k$.

定义. 带优先权 $P_1>\cdots>P_k$ 和变量 Z 的理论 T,其优先化限制 $\text{CIRC}(T;P_1>\cdots>P_k;Z)$可定义为:

$$\begin{aligned}&\text{CIRC}(T;P_2>\cdots>P_k;Z)\\&\quad=\text{CIRC}(T;P_1;\{P_2+\cdots+P_k+Z\})\\&\quad\&\,\text{CIRC}(T;P_2;\{P_3+\cdots+P_k+Z\})\\&\quad\&\cdots\&\,\text{CIRC}(T;P_k;Z)\end{aligned}$$

例,在上例中,假设 $P_1=\{\text{Abnormal}\}$,$P_2=\{\text{Fly}\}$,则:

$$\begin{aligned}\text{CIRC}(T;P_1>P_2;\{\})&=\text{CIRC}(T,P_1,P_2)\&\text{CIRC}(T;P_2,\{\})\\&=\{\text{Bird(tweety)},\text{Fly(tweety)}\}\cup\{\text{Bird(tweety)}\}\\&=\{\text{Bird(tweety)},\text{Fly(tweety)}\}\end{aligned}$$

2 面向对象逻辑语言 SCKE 简介

SCKE 是我们在 SUN 上开发的一种面向对象逻辑语言，其特点是在统一的语义解释基础上，既支持逻辑语言的描述性特征，又支持面向对象语言的结构化组织和信息隐藏、继承等性质，用一种自然而有效的方式集成这两种语言范例.

在 SCKE 中，一个程序由多个类对象(模块)组成，每个对象包含一组相关的方法定义(Prolog 子句). 对象之间通过消息进行通讯，向某一对象发送一条消息可解释为请求该对象证明一个目标. 类对象可以继承其超类对象的方法，以达到知识共享的目的，一个类对象可能有多个超类，多重继承通过回溯依次搜索这些超类，寻找到合适的方法. 定义超类的顺序决定了继承路径的搜索顺序. SCKE 的语法可形式地描述如下：

```
⟨Class⟩::=class (⟨atom⟩)
                ⟨interfaces⟩
                ⟨sentences⟩
          endclass
⟨interfaces⟩::=super(⟨atom⟩)|⟨interfaces⟩
⟨sentences⟩::=⟨clause⟩|⟨directive⟩|⟨circ_rule⟩
⟨clause⟩::=⟨head⟩:-⟨goals⟩|⟨head⟩
⟨head⟩::=⟨term⟩
⟨goals⟩::=⟨goals⟩,⟨goals⟩|⟨goals⟩;⟨goals⟩|⟨goal⟩
⟨goal⟩::=⟨term⟩|⟨atom⟩ send ⟨goals⟩
⟨directive⟩::=?-⟨goals⟩
⟨circ_rule⟩::=⟨antecedents⟩=>⟨conclusion⟩|⟨conclusion⟩
⟨antecedents⟩::=⟨antecedents⟩,⟨antecedents⟩|
                ⟨antecedents⟩;⟨antecedents⟩|
                ⟨antecedent⟩
⟨antecedent⟩::=⟨term⟩
⟨conclusion⟩::=⟨term⟩
```

其中，⟨circ_rule⟩表示限制理论规则.

除开其中的限制规则，SCKE 的操作语义可定义如下. 假设有如下约定：

P：由逻辑对象组成的 SCKE 程序

A：原子公式

g：目标

G：目标的合取式

|obj|={c|c 是对象 obj 中的子句}

Objects(P)={obj 是 P 中定义的对象}

O_i,O_j,O_s 为属于 Objects(P)的对象

定义. 程序 P 自顶到底导出目标 G 的过程，由目标子句序列：

$$G_0=G,G_1,G_2,\cdots\cdots$$

目标的求解区域序列：$O_{01}O_{02}$, $O_{11}O_{12}$, $O_{21}O_{22}$,……

以及最一般的一致化取代mgu序列:$\theta_0,\theta_1,\theta_2,\cdots\cdots$

组成,其中,$|O_iO_j|$表示对象O_i继承链上从O_i到O_j的组合公理集,G_{i+1}为G_i在环境$O_{i1}O_{j2}$上用mguθ_i导出.若导出过程有限,且最终目标为空子句,即$G_n=\square$,称该导出是成功的.

这里,SCKE的导出规则为:

(1)$\rightarrow\{true,\_\}_e$.

(2)$\{g,O_iO_j\}_k,\{G_k,O_iO_j\}_l\rightarrow\{(g,G),O_iO_j\}_{kl}$

(3)$\{G_k,O_iO_j\}_l\rightarrow\{A,O_iO_j\}_{k_l}$,其中,A是原子公式,求解区域$|O_iO_j|$中存在子句$A':-G$,使A与A′可一致化,且$\exists\ k=mgu(A,A')$.

(4)$\{A,O_iO_s\}_{kl}\rightarrow\{A,O_iO_j\}_{kl}$,其中$O_j$不是根对象,且$O_s$是$O_j$的超类.

(5)$\{A,O_iO_j\}_l\rightarrow\{O_i\ send\ A,\_\}_l$

3　SCKE引入限制理论的研究

从语义上说,限制理论与逻辑程序都是基于最小化模型的.两者的不同之处在于,逻辑程序最小化所有谓词的外延,且谓词最小化的次序完全由子句的定义决定,如子句:

$$H:-B_1,\cdots,B_n$$

其最小化的次序为$B_1,\cdots,B_n,H$.子句头和体的选择,以及子句体中谓词的次序不同,最小化的结果可能有很大不同.对整个逻辑程序同一谓词的最小化次序没有统一的全局约定.而限制理论仅最小化其中部分谓词的外延,并允许其它一部分谓词的外延随最小化过程而发生变化.在限制理论的最小化过程中,不具有逻辑程序中隐含的最小化次序,若有最小化次序问题则必须用谓词最小化优先权显式地给出(如优先化限制理论).

分层逻辑程序也将归结过程中谓词最小化次序显式地规定下来,与优先化限制理论有着异曲同功的效果.所谓分层逻辑程序是将程序T的所有谓词划分为k组$S_1,\cdots,S_k$,并定义程序中的每个子句

$$H:-B_1,\cdots,B_n\qquad(n\geqslant 1)$$

使,(i)若B_i为正文字,则$Stranum(B_i)\leqslant Stranum(H)$;(ii)若$B_i$为负文字$\rightarrow B_i'$,则$Stranum(B_i')<Stranum(H)$.其中,若$p\in S_j$,则$Stranum(p)=j$.可见,分层逻辑程序谓词最小化的次序为$S_1,\cdots,S_k$,与

$$CIRC(T;S_1>\cdots>S_k;\{\})$$

完全等价[4-6].

SCKE的语义是对一般逻辑程序语义的一种扩充,它将逻辑程序按抽象原则组织成一个逻辑理论的层次网络,网络上的结点是单个逻辑对象,弧则表示逻辑对象间的抽象/具体关系.

一个逻辑对象理论与逻辑程序理论的区别,是它能够与其它逻辑对象理论合作求解问题,即通过发送消息与其它逻辑对象理论通讯,或通过继承其超类对象理论来扩充自身的理论.如果继承作为理论的组合,消息目标作为外部条件,一个逻辑对象理论的语义可定义为带条件的组合理论的语义,这个组合的逻辑理论为该逻辑对象继承链上所有对象理论的集合,在最小化过程中,根据不同的外部条件(消息目标的成功或失败),得出不同的最小化

模型.

定义. 在SCKE中,一个对象O的继承链为一个对象序列

$$O_0=O,O_1,\cdots,O_n$$

其中,$O_{i+1}=\text{super}(O_i)$且$O_n$为根对象$(0\leqslant i\leqslant n-1)$.

定义. 在一个逻辑对象系统中,如果任一逻辑对象按继承链组合所得的逻辑理论都是分层的,则称该逻辑对象系统是分层的.

在分层逻辑对象系统中,每个逻辑对象的执行语义都与限制理论的执行语义等价.因此,要在SCKE中表示限制理论,并获得相应的推理结论,可以将限制理论编译为与之等价的分层逻辑对象系统.

4 SCKE引入限制理论的编译实现

假设限制理论 $\text{Circth}=\text{CIRC}(T;P_1>\cdots>P_n;Z)$

这里,T为分布在多个逻辑对象上的公理集,$P_1,\cdots,P_n,Z$为T中所有谓词的不交叉子集,其中,$P_1,\cdots,P_n$为最小化谓词,最小化优先次序为$P_1>\cdots>P_n$,Z为谓词变量,并进一步假设T中每条公理都至多包含一个属于Z的谓词.将Circth转化为分层SCKE程序,有3个方面的工作,第1,将各个类对象中的限制公理转化为分层逻辑程序规则,即将公理中最小化优先级最小的谓词作为规则结论,对谓词变量Z存在关系$P_n>Z$;第2,因为逻辑程序子句头必须为正文字,对那些结论为负文字的规则进行换名操作,将该负文字换名为一个正文字,(该正文字不与$P_1,\cdots,P_n,Z$中文字重名),形成逻辑程序子句;第3,执行换名操作后,有可能要丢失原有文字的信息,因为对逻辑程序来说,谓词名不同,其表示的含意也不同.为了保存这部分信息,可以在当前类对象的组合理论(含其继承链上所有逻辑对象理论)中对该文字进行展开操作,如同逻辑程序中的部分计算操作,一次展开所有被换名文字,使结果程序不依赖于丢失的谓词信息.展开方法为在当前类对象的组合理论中,寻找可与被换名文字所在的子句归结的子句,其中归结文字为被换名文字,归结后将结果转化为分层逻辑程序的规则.

限制理论到SCKE的编译实现算法可详细描述如下.算法的输入为单一逻辑对象中含有的限制公理及最小化优先关系,输出为编译后该逻辑对象中的分层子句集.在编译该逻辑对象之前,其所有超类必须已经经过编译.

算法1(编译算法)

```
Compile(T,P1>…>Pn,Z,T')
/* T为当前逻辑对象O所含的限制公理集,并假设限制公理形为
        L1,…,Lm=>A
/* T'为编译后对象O中的分层子句集,其中子句形为
        H:-B1,…,Bn
begin
    for T中每一条公理Circ_rule do
        circ_to_clause(Circ_rule,Stra_rule)
        if head_is_neg(Stra_rule)
```

```
            then rename(Stra_rule,Stra_clause)
                assert(T',Stra_clause)
            else assert(T',Stra_rule)
          endif
          resolve(Stra_rule)
        endfor
    endcompile
```

其中,circ_to_clause 将限制公理 Circ_rule 转化为分层逻辑程序规则 Stra_rule,head_is_neg 判断 Stra_rule 的规则头是否为负文字,rename 将 Stra_rule 的负规则头换名为正文字,从而将 Stra_rule 改为子句 Stra_clause,assert 将 Stra_clause 加入子句集 T',resolve 将当前逻辑对象所有超类中含被换名文字的子句用 Stra_rule 展开,对单继承系统和多继承系统,resolve 有不同的展开方法,分别描述如下.

算法2(单继承系统归结展开算法)

```
resolve1(Stra_rule,O)
/* 初始调用,O 为 Stra_rule 所在逻辑对象
begin
    S:=O
    repeat
        for S 中所有限制公理 Cric_rule do
            if Circ_rule 与 Stra_rule 可归结 &
              归结文字为 Stra_rule 或 Circ_rule 的换名文字
            then reso(Stra_rule,Circ_rule,Stra_rule')
                if head_is_neg(Stra_rule')
                then rename(Stra_rule',Stra_clause)
                assert(T',Stra_clause)
                else assert(T',Stra_rule')
                endif
                resolve1(Stra_rule',O)
            endif
        endfor
    P:=S,S:=super(S)
    until P=S
endresolve1
```

其中,reso 归结 Stra_rule 和 Circ_rule 得归结子句 Stra_rule',super 表示类/超类关系.

上述算法描述了单继承系统中归结展开方法,单继承系统中每个逻辑对象都只有一个继承链,要归结展开继承链上所有满足条件的子句,必须从当前对象开始逐步沿继承链向上,直到搜索到根对象.

SCKE 通过回溯实现多重继承,按继承链的概念,多继承系统中逻辑对象可能存在多个继承链,回溯过程则依次搜索这些继承链实现多重继承.而在同一继承链上的操作与单继承系统相同.由于在限制公理的编译过程中,只有在归结展开时要搜索其继承链,因此,对于多继承系统,归结时可以对各个继承链分别构造相应的归结子句以表示各继承链上的不同限制公理.多继承系统的归结展开过程如下所述.

算法3(多继承系统归结展开算法)

```
resolve2(Stra_rule,O_set,Path)
/* O_set 为当前需归结对象层上的对象集,初始调用时为[O]
/* Path 记录当前归结子句所属的继承链,初始调用时为[]
begin
    while O_set 不空 do
        First:=O_set 的第一个元素
        Rest:=O_set 的剩余元素
        for First 中所有限制公理 Circ_rule do
            if Cire_rule 与 Stra_rule 可归结 &
            归结文字为 Stra_rule 或 Circ_rule 的换名文字
            then reso(Stra_rule,Circ_rule,Stra_rule')
                if head_is_neg(Stra_rule')
                  then rename(Stra_rule',Stra_clause)
                       assert(T',Stra_clause,Path)
                  else assert(T',Stra_rule',Path)
                endif
                resolve2(Stra_rule',O_set,Path)
            endif
        endfor
            O_set1:=First 所有超类组成的表(不含循环超类)
            resolve2(Stra_rule,O_set1,[First|Path])
            O_set:=Rest
    endwhile
endresolve2
```

经过编译后的程序,其谓词最小化次序为 $P_1>\cdots>P_n>Z>\bar{Z}$($\bar{Z}$ 包含 Z 的换名谓词),与原限制理论的区别在于它对 Z 或 $\bar{Z}$ 也作了最小化处理.要与原限制理论所得结论一致,必须对结果程序的结论重新解释.若在分层程序 T′中证明公式 F,则有:

$$\mathrm{Ans}(\mathrm{CIRC}(T;P;Z),F)=\begin{cases}\text{yes} & \mathrm{Ans}(T',F)=\text{yes}\\ \text{no} & F\in P\ \&\ \mathrm{Ans}(T',F)=\text{no}\\ \text{unknown} & F\in Z\ \&\ \mathrm{Ans}(T',F)=\text{no}\end{cases}$$

5 实例分析

以下给出几个实例以说明上述限制理论的编译方法.

例1(单逻辑对象系统)

假设系统仅含有一个逻辑对象 O,O 中包含如下限制公理:

bird(tweety)

bird(tweety),¬ab(tweety)=>fly(tweety)

并假设 $P_1=[\text{bird}],P_2=[\text{ab}],Z=[\text{fly}]$

且限制优先次序为 $P_1>P_2$.

则其编译结果为分层逻辑程序:

bird(tweety)

fly(tweety):—bird(tweety),¬ab(tweety)

例2(单继承分类系统)

设有单继承分类系统,其继承关系为:

super(O_2)=O_1

super(O_3)=O_1

其中,O_1含限制公理

bird(x)

bird(x),¬ab(x)=>fly(x)

O_2含限制公理

penguin(x)

penguin(x),¬ab(x)=>¬fly(x)

并假设 P_1=[bird,penguin],P_2=[ab1],P_3=[ab],Z=[fly]

且限制优先次序为 $P_1>P_2>P_3$.

则其编译结果分别为分层逻辑程序 O_1':

bird(x)

fly(x):—bird(x),¬ab(x)

和分层逻辑程序 O_2':

penguin(x)

$\overline{\text{fly}}$(x):—penguin(x),¬ab1(x)

ab(x):—bird(x),penguin(x),¬ab1(x)

其中 $\overline{\text{fly}}$(x)为¬fly(x)的换名文字.

例3(多继承网络系统)

设有多继承网络系统,其继承关系为

super(O_3)=O_1

super(O_3)=O_2

其中,O_1含限制公理

quaker(x)

quaker(x),¬ab1(x)=>pacifism(x)

O_2含限制公理

republican(x)

republican(x),¬ab2(x)=>¬pacifism(x)

O_3含限制公理

repub_quaker(x)

repub_quaker(x),¬ab3(x)=>pacifism(x)

repub_quaker(x),¬ab4(x)=>pacifism(x)

并假设

P_1=[quaker,republican,repub_quaker]

P_2=[ab3,ab4],P_3=[ab1,ab2],Z=[fly]

且限制优先次序为 $P_1>P_2>P_3$.

则经编译后的分层逻辑对象系统为:

O_1:quaker(x)	[ ]
pacifism(x):−quaker(x),¬ab1(x)	[ ]
O_2:republican(x)	
pacifism(x):−republican(x),¬ab2(x)	[ ]
O_3:repub_quaker(x)	[ ]
$\overline{\text{pacifism}}$(x):−repub_quaker(x),¬ab3(x)	[ ]
ab2(x):−republican(x),repub_quaker(x),¬ab3(x)	[O_2]
$\overline{\text{pacifism}}$(x):−repub_quaker(x),¬ab4(x)	[ ]
ab1(x):−quaker(x),repub_quaker(x),¬ab4(x)	[O_1]

其中,每个子句后的对象表记录了该子句的继承路径条件,空表指任意路径.

结束语:限制理论是形式化常识知识的一种重要方法,本文在研究了限制理论语义和面向对象逻辑程序语义基础上,提出了用面向对象逻辑程序表示限制理论的方法,并实现了限制理论转化为面向对象逻辑程序的编译器.该编译器不仅能完成单继承系统的限制理论到面向对象逻辑程序 SCKE 的编译,而且能解决多继承系统的限制推理问题.运行实例表明,经过编译后得到的 SCKE 程序与原限制理论有一致的推理结论.该编译器为 SCKE 提供了表达常识知识的手段,大大提高了它的表达能力,为 SCKE 成为新一代知识系统的支撑语言提供了有力的支持.另一方面,由于限制理论是一种二阶理论,推理效率一般很低,这就限制了它在知识系统中的应用.将限制理论转化为具有分布特性的分层面向对象逻辑程序直接求解,是将限制理论推向实用的一种有效手段.

参考文献

1 McCarthy J. Circumscription ——a form of non−monotonic reasoning. Artificial Intelligence, 1985(**13**):27−39.

2 Jin Zhi, Hu Shouren. A new approach towards combining logic−with object−oriented paradigm for parallel knowledge processing. Proc. of the Int'l Conf. for Young Computer Scientists, Beijing, China, 1991:**7**.

3 金芝,胡守仁. SCKE:集成逻辑与面向对象范例的一种新模型.国防科技大学学报,1992:**3**.

4 Gelfond M, Przymusinska H, Przymusinski T. The extended closed world assumption and its relationship to parallel circumscription. Proceedings ACM SIGACT−SIGMOD Symposium on Principles of Database Systems, Cambridge, MA, 1986:133−139.

5 Gelfond M, Lifschitz V. Compiling circumscriptive theories into logic programs: preliminary report. Proceedings of the AAAI−88, 1988:445−459.

6 Gelfond M, Przymusinska H. On the rejationship between circumscription and negation as failure. Artificial Intelligence, 1989(**38**):75−94.

7 金芝,胡守仁.限制推理及其应用.计算机科学,1990(**6**).

THE CIRCUMSCRIPTIVE THEORIES IN THE OBJECT—ORIENTED LOGIC LANGUAGE SCKE

Jin Zhi and Hu Shouren

(*Department of Computer Science, Changsha Institute of Technology, Changsha* 410073)

Abstract Circumscription is an important theory for formalizing the commonsense reasoning. This paper mainly studies the possibility of transforming some special kinds of circumscription into the object—oriented logic language ——SCKE. A compiler has been implemented to complete the transformation. Running the result program in terms of the semantics of the object—oriented logic program, the same results as the original circumscriptive theory can be reached. So the nonmonotonic inheritance in the object—oriented logic language can be achieved to support the commonsense reasoning.

Key words Circumscription, object—oriented logic program, nonmonotonic inheritance, commonsense reasoning.