计算机科学 1995 Vol. 22№. 3

代数递归逻辑与人工智能*)

李爱中

(北方交通大学计算机系　北京 100044)

摘　要　融合代数和递归函数理论，本文建立了一个可以刻画知识表示、知识获取和自动推理的逻辑理论，给出了从示例中获取知识的多项式复杂性算法和自动推理的多项式复杂性算法。作为代数递归逻辑的应用，本文给出了人工智能中的约束满足问题的处理方案，探讨了人工智能的基本问题。

关键词　代数递归，逻辑理论，知识表示，知识获取，自动推理。

一、引　言

人工智能正面临着如何提高机器智能的严峻挑战，至今已提出了多种解决途径，例如人工神经网络、非单调逻辑或其它非标准逻辑、形象思维的研究、复杂巨型系统的研究等。本文提出另一种途径，结合经典代数理论与递归函数理论，建立一个用于知识表示、支持高效知识获取与自动推理的逻辑理论。这种想法的理由是：现有的形式理论，或表达能力太弱不足以应用；或表达能力太强(如一阶逻辑、Turing 机、递归函数等)带来了不可判定性和算法复杂性两个困难，阻碍了完美的理论之应用。吴文俊先生在使用代数方程表示几何定理及几何定理证明方面取得了巨大的成功，扩充"吴方法"到递归情形，会得到更加重要的结果。为此，作者结合代数与递归函数理论，进行了本文的研究工作，并得到了一些初步结果。

这些结果包括采用代数递归作为唯一的形式定义了一类广泛的函数和谓词足以满足实际应用需求；给出了代数递归函数和谓词的示例学习的多项式复杂性算法，给出了证明代数递归谓词永真性与否的多项式复杂性算法，进一步应用文中结果，作者给出了规划问题和约束满足问题的处理方案，探讨了人工智能的本质。

*)本文得到国家自然科学基金的资助，李爱中　博士后。

二、预备知识

本节介绍文中所需的背景知识，其中包括原始递归函数、代数方程和线性方程组的一些重要结果。除非特别声明，文中数及变量均取自于自然数集{0, 1, …}。

2.1　原始递归函数和原始递归谓词

定义 1　原始递归函数集 PRF 满足下列性质：

1°　PRF 含有下列函数作为其初始函数：(1)后继函数 $S(x)=x+1$；(2)零函数 $0(x)=0$；(3)投影函数 $P_j^n=(x_1,\cdots,x_n)=x_j(1\leqslant j\leqslant n)$。

2°　PRF 对于有限次使用合成运算和原始递归运算是封闭的。

合成运算：函数 $f(x_1,\cdots,x_n)$ 叫做由函数 $g_1(x_1,\cdots,x_n),\cdots,g_m(x_1,\cdots,x_n)$ 及 $h(x_1,\cdots,x_m)$ 通过合成运算得到的：

$$f(x_1,\cdots,x_n)=h(g_1(x_1,\cdots,x_n),\cdots,g_m(x_1,\cdots,x_n))$$

原始递归运算：函数 $f(x_1,\cdots,x_n)$ 叫做由函数 $g(x_1,\cdots,x_{n-1})$ 和函数 $h(x_1,\cdots,x_{n+1})$ 通过原始递归运算得到的：

$$\begin{cases} f(0,x_2,\cdots,x_n)=g(x_2,\cdots,x_n) \\ f(x_1+1,x_2,\cdots,x_n)=h(x_1,\cdots,x_n,f(x_1,\cdots,x_n)) \end{cases}$$

特别当 $n=1$ 时：

$$\begin{cases} f(0)=a \quad (a\text{ 为常数}) \\ f(x+1)=h(x,f(x)) \end{cases}$$

原始递归函数是一个最基本的函数类，从本质上包括了常用的大多数数论函数。特别具有重要含

义的是：合成运算反映了由子程序合成主程序的过程，原始递归运算反映了构造递归程序的过程。

上述定义反映了自底向上的组合过程，而且存在歧义性或不确定性。由于使用了两个算子不可能无回溯地进行反向归约或确定性归约，也就是说上述定义是组合式的，很难找到函数的唯一分解式。这种歧义性能否消除很难证明，而且可能是先天性的，这和形式语言的歧义性类似。非常幸运，作者在下文中找到了一个表达能力很强的运算——代数递归运算，它比上述两个算子加起来的功能还要强，而且代数递归运算是可分解的，对于组合函数可以很方便地找到其分解式，从而实现了确定性归约，这对提高演绎推理和归纳推理的效率至关重要。

定义 2 令 $P(x_1,\cdots,x_n)$是 n 元谓词，下列函数 $C_P(x_1,\cdots,x_n)$叫做 P 的特征函数：

$C_P(x_1,\cdots,x_n)=0$ iff $P(x_1,\cdots,x_n)$为真。

利用原始递归函数的概念可以定义原始递归谓词。

定义 3 若 n 元谓词 $P(x_1,\cdots,x_n)$的特征函数 $C_P(x_1,\cdots,x_n)$是原始递归函数，则称 $P(x_1,\cdots,x_n)$是原始递归谓词。

原始递归谓词对逻辑运算$\neg$，$\vee$和$\wedge$是封闭的。

2.2 代数方程

形如 $a_0x^n+\cdots+a_n=0$ 的方程称为关于 x 的 n 次代数方程，式中 n 为正整数，$a_0,\cdots,a_n$ 为整数，$a_0\neq 0$。设 ξ 为代数方程 $a_0x^n+\cdots+a_n=0$ 的任意一个根，那么 $|\xi|\leqslant\max\left(1,\frac{1}{|a_0|}\sum_{i=1}^{n}|a_i|\right)$。故此，一元 n 次代数方程是否有自然数根是可判定的，进而可以在存在自然根的前提下，求出最小的自然根。

设 $P_1(x_1,\cdots,x_k,y)=a_0y^m+\cdots+a_m$，$P_2(x_1,\cdots,x_k,y)=b_0y^n+\cdots+b_n$ 为两个多项式，$m>0$，$n>0$，$a_0,\cdots,a_m,b_0,\cdots,b_n$ 均为关于 $x_1,\cdots,x_k$ 的多项式，$a_0(x_1,\cdots,x_k)\not\equiv 0$，$b_0(x_1,\cdots,x_k)\not\equiv 0$。那么方程 $P_1(x_1,\cdots,x_k,y)=0$ 和 $P_2(x_1,\cdots,x_k,y)=0$ 在复数域上有公共的关于 y 的根的充分必要条件是右上角的结式。而 $R(P_1,P_2)$亦为关于 $x_1,\cdots,x_k$ 的多项式。

基于结式的上述结果，我们有下列归结引理。

引理 1（归结引理） 在自然数域上，若$\exists y(P_1(x_1,\cdots,x_k,y)=0\wedge P_2(x_1,\cdots,x_k,y)=0)$，那么存在代数方程 $P_3(x_1,\cdots,x_k)=0$ 成立。

$$R(P_1,P_2)=\begin{vmatrix} a_0\cdots\cdots a_m & & \\ \quad a_0\cdots\cdots a_m & & \\ \quad\cdots\cdots & & \\ \quad a_0\cdots\cdots a_m & & \\ b_0\cdots\cdots b_n & =0 & \\ \quad b_0\cdots\cdots b_n & & \\ \quad\cdots\cdots & & \\ \quad b_0\cdots\cdots b_n & & \end{vmatrix}\begin{matrix}\}\text{n 行}\\ \\ \\ \\ \}\text{m 行}\end{matrix}$$

[证明] 根据 $P_1(x_1,\cdots,x_k,y)=a_0y^m+\cdots+a_m$ 和 $P_2(x_1,\cdots,x_k,y)=b_0y^n+\cdots+b_n$ 中 m 和 n 的情形直接构造 $P_3(x_1,\cdots,x_k)$：

1° $m>0,n>0$：$P_3(x_1,\cdots,x_k)=R(P_1,P_2)$

2° $m=0,n>0$：$P_3(x_1,\cdots,x_k)=R(P_1^2+P_2^2,P_2)$

3° $m>0,n=0$：$P_3(x_1,\cdots,x_k)=R(P_1,P_1^2+P_2^2)$

4° $m=0,n=0$：$P_3(x_1,\cdots,x_k)=P_1^2+P_2^2$

显然，$P_3(x_1,\cdots,x_k)$满足此引理。 □

归结引理对本文十分重要，它相当于归结原理在归结证明的作用[2]。归结引理的证明实际上给出了实现归结的方法，该方法很简单，而未使用复杂的合一操作。我们称 $P_3(x_1,\cdots,x_k)=0$ 为从 $P_1(x_1,\cdots,x_k,y)=0$ 和 $P_2(x_1,\cdots,x_k,y)=0$ 中消去 y 的结果。

2.3 齐次线性方程组的约束解

设 $V=\{x_1,x_2,\cdots,x_n\}$，$NZV\subseteq V$，称 NZV 中变元为约束变元。整系数齐次线性方程组：

$$\begin{cases} a_{11}x_1+\cdots+a_{1n}x_n=0\\ a_{21}x_1+\cdots+a_{2n}x_n=0\\ \cdots\cdots\cdots\cdots\\ a_{m1}x_1+\cdots+a_{mn}x_n=0 \end{cases}\tag{1}$$

在什么条件下有约束解，即若 $x_i\in NZV$，则整数 $x_i\neq 0$？若有，如何求出一组约束解？

设

$$A=\begin{bmatrix} a_{11}\cdots\cdots a_{1n}\\ a_{21}\cdots\cdots a_{2n}\\ \cdots\cdots\cdots\\ a_{m1}\cdots\cdots a_{mn}\end{bmatrix}$$

对 A 实施矩阵变换得到 A'：

$$A'=\begin{bmatrix} b_{11} & & b_{1r+1}\cdots b_{1n}\\ & b_{22} & b_{2r+1}\cdots b_{2n}\\ & \cdots & \cdots\cdots\\ & & b_{rr}\quad b_{rr+1}\cdots b_{rn}\end{bmatrix}$$

其中诸元素 b_{ij}均为整数，且 $b_{kk}\neq 0(1\leqslant k\leqslant r)$，有：

$A'(x'_1,\cdots,x'_n)^T=0,\{x'_1,\cdots,x'_n\}=\{x_1,\cdots,x_n\}$

引理2 方程组(1)有约束解的充分必要条件是：

$\forall x'_i\in\{x'_1,\cdots,x'_r\}\cap NZV, b_{r+1}^2+\cdots+b_{in}^2\neq 0$

[证明](反证法)若存在 $x_i\in\{x'_1,\cdots,x'_r\}\cap NZV$，但 $b_{r+1}^2+\cdots+b_{in}^2=0$，则 $b_{ir+1}=\cdots=b_{in}=0$，显然 $x'_i=0$，而 $x'_i\in NZV$，显然方程组(1)无约束解。

直接给出求约束解的算法 LEA：

1° 若 $\{x'_1,\cdots,x'_r\}\cap NZV=\{\}$，则设

$$x'_j=\begin{cases}[|b_{11}|,\cdots,|b_{rr}|] & \text{当 } x'_j\in NZV \text{ 时}^{①}\\ 0 & \text{当 } x'_j\notin NZV \text{ 时}\end{cases}$$

其中 $j=r+1,\cdots,n$；

注①$[x_1,\cdots,x_n]$为正整数 $x_1,\cdots,x_n$ 的最小公倍数。

$x'_j=-(b_{jr+1}x'_{r+1}+\cdots+b_{jn}x'_n)/b_{jj}$，其中，$j=1,\cdots,r$；返回。

2° 若 $\{x'_1,\cdots,x'_r\}\cap NZV\neq\{\}$，且 $r+1=n$ 时，则设 $x'_{r+1}=[|b_{11}|,\cdots,|b_{rr}|]$

$x'_j=-(b_{jr+1}x'_{r+1})/b_{jj}, j=1,\cdots,r$；返回。

3° 若 $\{x'_1,\cdots,x'_r\}\cap NZV\neq\{\}$，且 $r+1<n$ 时，则先用此算法 LEA 解出关于 A''的约束解 $x',\cdots x'_r$ 及 $x'_{r+2},\cdots,x'_n$，其中 $NZV'=NZV-\{x_i/1\leqslant i\leqslant r, b_{ir+2}^2+\cdots+b_{in}^2=0\}$：

$$A''=\begin{bmatrix} b_{11} & & b_{1r+2}\cdots b_{1n}\\ & b_{22} & b_{2r+2}\cdots b_{2n}\\ & \cdots & \cdots\cdots\cdots\\ & & b_{rr} \quad b_{rr+2}\cdots b_{rn}\end{bmatrix}\begin{bmatrix}x'_1\\ \vdots\\ x'_r\\ x''_{r+2}\\ \vdots\\ x'_n\end{bmatrix}=0$$

然后修正 $x'_1,\cdots,x'_r$。

设 $a=\max_j\{|b_{jr+2}x'_{r+2}+\cdots\cdots+b_{jn}x'_n|\}$

$x'_{r+1}=(a+1)[|b_{11}|,\cdots,|b_{rr}|]$

$x'_i=-(b_{ir+1}x'_{r+1}+\cdots\cdots+b_{in}x'_n)/b_{ii}(i=1,r)$返回。

显然，$(x'_1,\cdots,x'_n)$为方程(1)的一组约束解。

故此，本引理得证。 □

由上述算法 LEA，容易得出下列结论：

引理3 算法 LEA 的时间复杂性为多项式 $P(n)$，其中 n 为变元个数。

三、代数递归逻辑

代数递归逻辑在结构上很简单。在语言部分只引入了一个算子(或运算)——代数递归运算来生成代数递归函数，进而用代数递归函数作为特征函数来定义代数递归谓词。在推理部分分别研究了归纳推理和演绎推理，给出了相应的算法。

3.1 代数递归函数

定义4 代数递归函数是指有限次使用下列代数递归运算而得到的函数 $f(x_1,\cdots,x_n)$：

$$\begin{cases}f(0,x_2,\cdots,x_n)=g(x_2,\cdots,x_n)\\ f(x_1+1,x_2,\cdots,x_n)\text{为 } P(x_1,\cdots,x_n,f(x_1,\cdots,x_n),t)=0\\ \qquad\text{关于 t 的最小自然数根。}\end{cases}$$

特别当 $n=1$ 时：

$$\begin{cases}f(0)\text{为 } P_0(t)=0 \text{ 的最小自然数根}\\ f(x-1)\text{为 } P(x,f(x),t)=0 \text{ 关于 t 的最小自然根}\end{cases}$$

其中 $g(x_2,\cdots,x_n)$为已知的代数递归函数，$P(x_1,\cdots,x_n,x_{n+1},x_{n+2})$为已知的多项式，$P_0(t)$为已知的多项式。

显然，上述定义的代数递归函数中包含了部分函数和全函数，但代数递归函数是可计算的。现讨论代数递归函数与原始递归函数的关系。

定理1 $O(x)$，$S(x)$和 $P_j^n(x_1,\cdots,x_n)(1\leqslant j\leqslant n)$为代数递归函数。

定理2 若 $g_1(x_1,\cdots,x_n)$，$\cdots$，$g_m(x_1,\cdots,x_n)$和 $h(x_1,\cdots,x_m)$均为代数递归函数且为全函数，那么，$h(g_1(x_1,\cdots,x_n),\cdots,g_m(x_1,\cdots,x_n))$也为代数递归函数且为全函数。

[证明] 施数学归纳法于 n。

1° 奠基 设 $f(x)=h(g_1(x),\cdots,g_m(x))$。设 $P_0(t)=t-h(g_1(0),\cdots,g_m(0))$。由于 $g_1,\cdots,g_m$ 和 h 均为代数递归函数，故有代数方程：

$$Ph(x_1,\cdots,x_m,h(x_1,\cdots,x_m),h(x_1+1,x_2,\cdots,x_m))=0 \quad (1)$$

$$Pg_i(x,g_i(x),\cdots,g_i(x+1))=0 \quad (2)$$

将(1)中 x_i 代以 $g_i(x)$得：

$$Ph(g_1(x),\cdots,g_m(x),h(g_1(x),\cdots,g_m(x)),h(g_1(x)+1,g_2(x),\cdots,g_m(x)))=0 \quad (3)$$

由(3)、(2)消去诸 $g_i(x)$得：

$$P_1(x,g_1(x+1),\cdots,g_m(x+1),h(g_1(x),\cdots,g_m(x)),h(g_1(x)-1,g_2(x),\cdots,g_m(x)))=0 \quad (4)$$

由(4)、(2)消去诸 $g_i(x)$得：

$$P_2(x,g_1(x),\cdots,g_m(x),h(g_1(x),\cdots,g_m(x)),h(g_1(x)+1,g_2(x),\cdots,g_m(x)))=0 \quad (5)$$

由(5)、(3)消去 $h(g_1(x)+1,g_2(x),\cdots,g_m(x))$得：

$P_3(x,g_1(x),\cdots,g_m(x),h(g_1(x),\cdots,g_m(x)))=0$ (6)

将(6)中 x 代入 x+1 并重写得：

$P_4(x,g_1(x+1),\cdots,g_m(x+1),h(g_1(x+1),\cdots,g_m(x+1)))=0$ (7)

由(7)、(2)消去诸 $g_i(x+1)$得：

$P_5(x,g_1(x),\cdots,g_m(x),h(g_1(x+1),\cdots,g_m(x+1)))$ (8)

由(6)、(8)消去 $g_1(x),\cdots,g_m(x)$得：

$P_6(x,h(g_1(x),\cdots,g_m(x)),h(g_1(x+1),\cdots,g_m(x+1)))=0$ (13)

即 $P_6(x,f(x),f(x+1))=0$

故此：

$\begin{cases}f(0)\text{为 }P_0(t)=0\text{的最小自然数根}\\ f(x+1)\text{为 }P_6(x,f(x),t)=0\text{关于 t 的最小自然数根。}\end{cases}$

所以 f(x)为代数递归函数。因此，1°成立。

2° 归纳 假设：若 $g'_1(x_1,\cdots,x_n),\cdots,g'_m(x_1,\cdots,x_n)$及对任意 m，$h'(x_1,\cdots,x_m)$为代数递归函数，则 $h'(g'_1(x_1,\cdots,x_n),\cdots,g'_m(x_1,\cdots,x_n))$为代数递归函数。当 $g_1(x_1,\cdots,x_{n+1}),\cdots,g_m(x_1,\cdots,x_{n+1})$及 $h(x_1,\cdots,x_m)$为代数递归函数时，设 $g_i(0;x_2,\cdots,x_{n+1})=gA_i(x_2,\cdots,x_{n+1})$为代数递归函数。由归纳假设知：$A(x_2,\cdots,x_{n+1})=h(gA_1(x_2,\cdots,x_{n+1}),\cdots,gA_m(x_2,\cdots,x_{n+1}))$为代数递归函数。 (a)

由于 h 和诸 g_i 为代数递归函数，故有：

$Ph(x_1,\cdots,x_m,h(x_1,\cdots,x_m),h(x_1+1,x_2,\cdots,x_m))=0$ (1)

$Pg_i(x_1,\cdots,x_{n+1},g_i(x_1,\cdots,x_{n+1}),g_i(x_1+1,x_2,\cdots,x_{n+1}))=0$ (2)

将(1)中 x_i 代入 $g_i(x_1,\cdots,x_{n+1})$得：

$Ph(g_1(x_1,\cdots,x_{n+1}),\cdots,g_m(x_1,\cdots,x_{n+1}),h(g_1(x_1,\cdots,x_{n+1}),\cdots,g_m(x_1,\cdots,x_{n+1})),h(g_1(x_1,\cdots,x_{n+1})+1,g_2(x_1,\cdots,x_{n+1}),\cdots,g_m(x_1,\cdots,x_{n+1})))=0$ (3)

由(3)、(2)消去诸 $g_i(x_1,\cdots,x_{n+1})$得：

$P_1(x_1,\cdots,x_{n+1},g_1(x_1+1,x_2,\cdots,x_{n+1}),\cdots,g_m(x_1+1,x_2,\cdots,x_{n+1}),h(g_1(x_1,\cdots,x_{n+1}),\cdots,g_m(x_1,\cdots,x_{n+1})),h(g_1(x_1,\cdots,x_{n+1})+1,g_2(x_1,\cdots,x_{n+1}),\cdots,g_m(x_1,\cdots,x_{n+1})))=0$ (4)

由(4)、(2)消去诸 $g_i(x_1+1,x_2,\cdots,x_{n+1})$得：

$P_2(x_1,\cdots,x_{n+1},g_1(x_1,\cdots,x_{n+1}),\cdots,g_m(x_1,\cdots,x_{n+1}),h(g_1(x_1,\cdots,x_{n+1}),\cdots,g_m(x_1,\cdots,x_{n+1})),h(g_1(x_1,\cdots,x_{n+1})+1,g_2(x_1,\cdots,x_{n+1}),\cdots,g_m(x_1,\cdots,x_{n+1})))=0$ (5)

由(5)、(3)消去 $h(g_1(x_1,\cdots,x_{n+1})+1,g_2(x_1,\cdots,x_{n+1}),\cdots,g_m(x_1,\cdots,x_{n+1}))$得：

$P_3(x_1,\cdots,x_{n+1},g_1(x_1,\cdots,x_{n+1}),\cdots,g_m(x_1,\cdots,x_{n+1}),h(g_1(x_1,\cdots,x_{n+1}),\cdots,g_m(x_1,\cdots,x_{n+1}))=0$ (6)

(6)中 x_1 改名后代入 x_1+1 并重写得：

$P_4(x_1,\cdots,x_{n+1},g_1(x_1+1,x_2,\cdots,x_{n+1}),\cdots,g_m(x_1+1,x_2,\cdots,x_{n+1}),h(g_1(x_1+1,x_2,\cdots,x_{n+1}),\cdots,g_m(x_1+1,x_2,\cdots,x_{n+1})))=0$ (7)

(7)、(2)消去诸 $g_i(x_1+1,x_2,\cdots,x_{n+1})$得：

$P_5(x_1,\cdots,x_{n+1},g_1(x_1,\cdots,x_{n+1}),\cdots,g_m(x_1,\cdots,x_{n+1}),h(g_1(x_1+1,x_2,\cdots,x_{n+1}),\cdots,g_m(x_1+1,x_2,\cdots,x_{n+1})))=0$ (8)

由 (6)、(8)消去 $g_1(x_1,\cdots,x_{n+1}),\cdots,g_m(x_1,\cdots,x_{n+1})$得：

$P_6(x_1,\cdots,x_{n+1},h(g_1(x_1,\cdots,x_{n+1}),\cdots,g_m(x_1,\cdots,x_{n+1})),(g_1(x_1+1,x_2,\cdots,x_{n+1}),\cdots,(g_m(x_1+1,x_2,\cdots,x_{n+1})))=0$ (b)

由(a)、(b)得 $h(g_1(x_1,\cdots,x_{n+1}),\cdots,g_m(x_2,\cdots,x_{n+1}))$为代数递归函数。故此 2°成立。

由 1°，2°得此定理成立。 □

定理 3 若 $A(x_2,\cdots,x_n)$和 $B(x_1,\cdots,x_{n+1})$是代数递归函数而且是全函数，则由原始递归式：

$\begin{cases}f(x_2,\cdots,x_n)=A(x_2,\cdots,x_n)\\ f(x_1+1,x_2,\cdots,x_n)=B(x_1,\cdots,x_n,f(x_1,\cdots,x_n))\end{cases}$

定义的函数 $f(x_1,\cdots,x_n)$也是代数递归函数而且是全函数。

[证明]：由于 $B(x_1,\cdots,x_{n+1})$为代数递归函数，故有代数方程：

$PB(x_1,\cdots,x_{n+1},B(x_1,\cdots,x_{n+1}),B(x_1+1,x_2,\cdots,x_{n+1}))=0$ (1)

(1)中(x_{n+1})代入 $f(x_1,\cdots,x_n)$得：

$PB(x_1,\cdots,x_n,f(x_1,\cdots,x_n),B(x_1,\cdots,x_n,f(x_1,\cdots,x_n)),B(x_1+1,x_2,\cdots,x_n,f(x_1,\cdots,x_n)))=0$

即 $PB(x_1,\cdots,x_n,f(x_1,\cdots,x_n),f(x_1+1,x_2,\cdots,x_n),B(x_1+1,x_2,\cdots,x_n,f(x_1,\cdots,x_n)))=0$ (2)

(1)中 x_1 改名后代入 x_1-1 并重写得：

$PB'(x_1,\cdots,x_{n+1},B(x_1-1,x_2,\cdots,x_{n+1}),B(x_1,\cdots,x_{n+1}))=0$ (3)

(3)中 x_{n+1}代入 $f(x_1,\cdots,x_n)$得：

$PB'(x_1,\cdots,x_n,f(x_1,\cdots,x_n),B(x_1-1,x_2,\cdots,x_n,f(x_1,\cdots,x_n)),B(x_1,\cdots,x_n,f(x_1,\cdots,x_n)))=0$

即：$PB'(x_1,\cdots,x_n,f(x_1,\cdots,x_n),B(x_1-1,x_2,\cdots,x_n,f$

$(x_1,\cdots,x_n)),f(x_1+1,x_2,\cdots,x_n))=0$ (4)

由(2)、(4)消去 $f(x_1,\cdots,x_n)$ 得：

$P_1(x_1,\cdots,x_n,f(x_1+1,x_2,\cdots,x_n),B(x_1+1,x_2,\cdots,x_n,f(x_1,\cdots,x_n)),B(x_1-1,x_2,\cdots,x_n,f(x_1,\cdots,x_n)))=0$ (5)

由(2)、(4)消去 $f(x_1+1,x_2,\cdots,x_n)$ 得：

$P_2(x_1,\cdots,x_n,f(x_1,\cdots,x_n),B(x_1+1,x_2,\cdots,x_n,f(x_1,\cdots,x_n)),B(x_1-1,x_2,\cdots,x_n,f(x_1,\cdots,x_n)))=0$ (6)

由(5)、(6)消去 $B(x_1-1,x_2,\cdots,x_n,f(x_1,\cdots,x_n))$ 得：

$P_3(x_1,\cdots,x_n,f(x_1,\cdots,x_n),f(x_1+1,x_2,\cdots,x_n),B(x_1+1,x_2,\cdots,x_n,f(x_1,\cdots,x_n)))=0$ (7)

由(2)、(7)消去 $B(x_1+1,x_2,\cdots,x_n,f(x_1,\cdots,x_n))$ 得：

$P_4(x_1,\cdots,x_n,f(x_1,\cdots,x_n),f(x_1+1,x_2,\cdots,x_n))=0$

故此，$f(x_1,\cdots,x_n)$ 为代数递归函数，本定理得证。□

定理4 原始递归函数集为代数递归函数集的真子集。

定理1～4表明了代数递归运算的功能至少比合成运算和原始递归运算加起来还要强。故此，代数递归函数具有很强的表达能力，足以满足一般应用的需求。

3.2 代数递归谓词

定义5 n元谓词 $P(x_1,\cdots,x_n)$ 为代数递归谓词当且仅当其特征函数 $C_P(x_1,\cdots,x_n)$ 为代数递归函数而且是全函数。

显然，原始递归谓词是代数递归谓词的真子集，而且代数递归谓词对逻辑连接词¬、∨和∧是封闭的。

至此，由3.1节和3.2节完成了代数递归逻辑的语言部分。

3.3 归纳推理

归纳推理研究递归代数函数和递归代数谓词的示例学习，关于归纳推理的综述性文章见文[4,5,6]。下面讨论多项式谓词的示例学习。

假设 $x_1,\cdots,x_n$ 为n个变量，考虑在封闭世界 $[S_1]\times\cdots\times[S_n]$ 的示例集E，其中 $[S_i]$ 表示集合 $\{i/i$ 是自然数，$i\leqslant S_i\}$。$E=E_+\cup E_-$，其中 E_+ 为E中正例，E_- 为E中反例，$E=[S_1]\times\cdots\times[S_n]$，也就是说封闭世界内的例必是正例或反例；另外，一个例不能既是正例又是反例，即 $E_+\cap E_-=\{\}$。

设 $f(x_1,\cdots,x_n)$ 为一多项式，规定 $(N_1,\cdots,N_n)$ 为其次数序列，其中 N_i 为 x_i 在 $f(x_1,\cdots,x_n)$ 中的最高次数。

多项式谓词的归纳关键在于反例的进一步实例化，因为已知 $(a_1,\cdots,a_n)$ 为 $P(x_1,\cdots,x_n)$ 的反例，只能得出 $P(x_1,\cdots,x_n)$ 的特征函数 $f(x_1,\cdots,x_n)$ 在 $(a_1,\cdots,a_n)$ 处的值 $f(a_1,\cdots,a_n)\neq 0$。若能把 $f(x_1,\cdots,x_n)$ 进一步实例化，即求出 $f(x_1,\cdots,x_n)$ 在 $(a_1,\cdots,a_n)$ 处的值，则可以用曲线拟合手段或其它方法求出多项式 $f(x_1,\cdots,x_n)$，即求出了多项式谓词本身。

康托编号函数：

$$\begin{cases}\pi(a,b)=\dfrac{(a+b)(a+b+1)}{2}+a\\ L(x)=\left\{\dfrac{(2x-\left\{\frac{(\sqrt{8x+1})-1}{2}\right\}^2)-\left\{\frac{(\sqrt{8x+1})-1}{2}\right\}}{2}\right\}\\ R(x)=\left\{\dfrac{(\sqrt{8x+1})-1}{2}\right\}-L(x)\end{cases}$$

其中 $\{x\}$ 表示不超过 x 的最大自然数。

康托编号函数具有下列性质：

1° $L(\pi(a,b))=a$；

2° $R(\pi(a,b))=b$；

3° 对任意一自然数 i，存在唯一的二元组(a,b)使得 $\pi(a,b)=i$。

利用康托编号可以对多元组进行编号，如

$$\begin{cases}G((N_1,\cdots,N_n)=\pi(N_1,\pi(N_2,\cdots,\pi(N_{n-1},N_n)\cdots))\\ G_1(N)=L(N)\\ G_i(N)=R^{i-1}(L(N)),1<i<n\\ G_N=R^n(N)\end{cases}$$

引入记号：

$\Delta_i f(x_1,\cdots,x_n)=f(x_1,\cdots,x_{i-1},x_i+1,x_{i+1},\cdots,x_n)-f(x_1,\cdots,x_n),1\leqslant i\leqslant n$

$\Delta_i^{k+1}f(x_1,\cdots,x_n)=\Delta_i^k f(x_1,\cdots,x_{i-1},x_i+1,x_{i+1},\cdots,x_n)-\Delta_i^k f(x_1,\cdots,x_n),(1\leqslant i\leqslant n)$

下面给出了从正反例中归纳多项式谓词的特征函数的算法 LPP(E,$x_1,\cdots,x_n$,f)：

```
LPP(E,x1,…,xn,f)
  begin
  if E=E+ then f(x1,…,xn)=0
  else if E=E- then f(x1,…,xn)=1
  else for i=1 to G(2S1,…,2Sn) do
  begin
    作出线性方程组 Δ^(N1+1) f(a1,…,an)=0, j=1,
    …,n, 其中若(a1,…,an)∈E+, 则 f(a1,…,an)=
```

0，否则把 $f(a_1,\cdots,a_n)$ 看成约束变量；若此方程组有约束解，则求出一组约束解进而使用曲线拟合或其它方法求出 $f(x_1,\cdots,x_n)$[3]并跳出循环；

end；

基于上述算法 LPP，给出了从示例中学习代数递归函数的算法 LRAf(E,$x_1,\cdots,x_n$,f)：

LRAF(E,$x_1,\cdots,x_n$,f)

if n=1 then begin

根据 f(0)作出相应方程 $P_0(t)=0$；

从函数 f(x)的示例中构造多项多谓词 B(X,Y,Z)的示例 $E'_{\perp}=\{(a,f(a),f(a+1))$

其中 $a+1\in S_1,a\geqslant 0\}$；

设 $S_2=\max_{x\in S_1}\{f(x)\}$；　$S_3=\max_{\substack{x\geqslant 0\\ x+1\in S_1}}\{f(x+1)\}$；

$E'_{-}=\{(x,y,z)/x\in S_1,y\in S_2,z\in S_3\}-E_{\perp}$；

规定 B(x,f(x),j),$j\neq f(x+1)$和 B(x,i,f(x+1)),$i\neq f(x)$为约束变元；

LPP(E',x,f(x),f(x+1),B)；

end；

if n>1 then begin

从已知示例中析取出 $f(0,x_1,\cdots,x_n)$的示例 E'；

LRAF(E',$x_2,\cdots,x_n$,f')；

从已知 $f(x_1,\cdots,x_n)$的示例中构造出多项式谓词 $B(x_1,\cdots,x_n,x_{n+1},x_{n+2})$的示例，

$E'_{\perp}=\{(x_1,\cdots,x_n,f(x_1,\cdots,x_n),f(x_1+1,x_2,\cdots,x_n))|x_1\in S_1,\cdots,x_n\in S_n,x_1+1\in S_1\}$；

设 $S_{n+1}=\max_{x_1\in S_1}\{f(x_1,\cdots,x_n)\}$；

$S_{n+2}=\max_{\substack{x_1\in S_1\\ x_1+1\in S_1}}\{f(x_1+1,x_2,\cdots,x_n)\}$

$E'_{-}=\{(x_1,\cdots,x_n,x_{n+1},x_{n+2})/x_i\in S_i\}-E'_{+}$；

规定约束变元，LPP(E',$x_1,\cdots,x_n$,f($x_1,\cdots,x_n$),f($x_1+1,x_2,\cdots,x_n$),B)；

end；

类似地，我们可以给出从示例中学习代数递归谓词的特征函数的算法。

至此，我们完成了递归逻辑的归纳推理部分的描述。

3.4 演绎推理

演绎推理主要研究代数递归谓词的永真性。若 $P(x_1,\cdots,x_n)$为一递归代数谓词，$P(x_1,\cdots,x_n)$为永真的，当且仅当$\forall x_i\in N$，$P(x_1,\cdots,x_n)$为真。关于递归代数谓词我们有下列结果。

定理 5　$P(x_1,\cdots,x_n)$永真当且仅当其特征函数 $f(x_1,\cdots,x_n)\equiv 0$。

这样，证明 $P(x_1,\cdots,x_n)$永真，即证 $f(x_1,\cdots,x_n)\equiv 0$。下面给出证明 $f(x_1,\cdots,x_n)\equiv 0$ 的算法 Proof(f,n)。

设

$$\begin{cases} f(0,x_2,\cdots,x_n)=A(x_2,\cdots,x_n)\\ B(x_1,\cdots,x_n,f(x_1,\cdots,x_n),f(x_1+1,x_2,\cdots,x_n))=0\end{cases}$$

Proof(f,n):Boolean;

if n=1 then

if f(0)≠0 then Proof(f,n)=false,

else if $B(x_1,\cdots,x_n,0,0)\equiv 0$ then

Proof(f,n)=true

else Proof(f,n)=false

else if Proof(A,n−1)=true and $B(x_1,\cdots,x_n,0,0)\equiv 0$

then Proof(f,n)=true

else Proof(f,n)=false

其中 $B(x_1,\cdots,x_n,0,0)\equiv 0$ 是可证明的，其证明过程如下；

用数学归纳法证明 $B(x_1,\cdots,x_n,0,0)\equiv 0$。设，

$$\begin{cases} B(x_1,\cdots x_n,0,0)=B_1(x_2,\cdots,x_n,0,0)\\ B(x_1-1,x_2,\cdots,x_n,0,0)=B(x_1,\cdots,x_n,0,0)\\ \quad +B_2(x_1,\cdots,x_n,0,0)\end{cases}$$

证明 $B(x_1,\cdots,x_n,0,0)\equiv 0$，即 $B_1(x_2,\cdots,x_n,0,0)\equiv 0$ 和 $B_2(x_1,\cdots,x_n,0,0)\equiv 0$。这样是一种无回溯的反向分解证法。

由于 Proof 和证明 $B(x_1,\cdots,x_n,0,0)\equiv 0$ 均采用了自顶向下的反向分解证法，故此证明效率很高，不产生无用的中间结果；而且是否永真均可证明。

3.5 高阶代数递归函数和高阶代数递归谓词

定义 6　规定代数递归函数为一阶函数，代数递归谓词为一阶谓词，那么(n+1)阶函数的定义为由下列递归式生成的函数：

$$\begin{cases} f(0,x_2,\cdots,x_n)=g(x_2,\cdots,x_n)\\ f(x_1+1,x_2,\cdots,x_n)\text{为 }h(x_1,\cdots,x_n,f(x_1,\cdots,x_n),t)=0\\ \qquad\qquad\text{关于 t 的最小自然根。}\end{cases}$$

当 n=1 时

$$\begin{cases} f(0)\text{为 }h_0(t)=0\text{ 的最小自然根}\\ f(x+1)\text{为 }h(x,f(x),t)=0\text{ 关于 t 的最小自然根}\end{cases}$$

其中 $g(x_2,\cdots,x_n)$为(n+1)阶函数，$h_0(t)$和 $h(x_1,$

…,x_{n-2})为 n 阶谓词。

谓词 $P(x_1,\cdots,x_n)$为(n−1)阶谓词,当且仅当其特征函数 $C_P(x_1,\cdots,x_n)$为(n+1)阶函数。

显然有 n 阶函数⊂(n−1)阶函数,n 阶谓词⊂(n−1)阶谓词。并且对任意 n,n 阶函数或 n 阶谓词均是可计算的。进而,可以研究 n 阶函数及 n 阶谓词的演绎推理与归纳推理,在分层递增的基础上研究推理的逐级进化方式。

四、约束满足问题的处理方案

大多数约束满足问题(Constraint Satisfaction Problem,CSP)可以在代数递归逻辑下得到解,其步骤如下:

(Ⅰ)把常规下的 CSP 转化成代数递归逻辑下的 CSP 问题表示,形式如下:

$$\begin{cases} f_1(x_1,\cdots,x_n)=0 \\ f_2(x_1,\cdots,x_n)=0 \\ \quad\vdots \\ f_m(x_1,\cdots,x_n)=0 \end{cases} \tag{1}$$

其中 $f_1,\cdots,f_m$ 均为代数递归函数。

(Ⅱ)化多约束条件为单约束条件,令 $F(x_1,\cdots,x_n)=f_1(x_1,\cdots,x_n)+\cdots+f_m(x_1,\cdots,x_n)$。显然,$F(x_1,\cdots,x_n)=0$ iff $f_1(x_1,\cdots,x_n)=0\wedge\cdots\wedge f_m(x_1,\cdots,x_n)=0$。

(Ⅲ)应用 3.4 节中的算法证明 $F(x_1,\cdots,x_n)\geqslant 1$,即 $1\dot{-}F(x_1,\cdots,x_n)\equiv 0$,其中 $a\dot{-}b$ 定义如下:

$$a\dot{-}b=\begin{cases} a-b & a>b \\ 0 & a\leqslant b \end{cases}$$

若 $F(x_1,\cdots,x_n)\geqslant 1$ 对所有 $x_1,\cdots,x_n$ 成立,则(1)无解;否则(1)有解并转至(Ⅳ)

(Ⅳ)采用深度优先的分解法求解。令 $F(x_1,\cdots,x_n)$定义式为

$$\begin{cases} F(0,x_2,\cdots,x_n)=A(x_2,\cdots,x_n) & (2) \\ F(x_1+1,x_2,\cdots,x_n)\text{为 } P(x_1,\cdots,x_n, \\ \quad F(x_1,\cdots,x_n),t)=0 & (3) \end{cases}$$

的最小自然根

若 $A(x_2,\cdots,x_n)=0$ 有解,则求之并结束;否则求解 $P(x_1,\cdots,x_n,F(x_1,\cdots,x_n),0)=0$ 并结束。

五、结论与进一步工作

基于代数和递归函数理论,作者在本文中总结了近十年来的工作,形成了一套体系,并称之为代数递归逻辑。代数递归逻辑的关键是代数递归运算的形式,它是可反向分解的,这样就使代数递归逻辑区别于传统的组合逻辑,后者的反向分解不唯一,这样导致了反向分解搜索或自顶向下的搜索不可避免地带有不确定性,回溯是不可避免的,因而必然产生组合爆炸。基于代数递归的归纳推理和演绎推理完全使用反向归约或搜索策略,避免了组合爆炸,从而给出了高效的算法。

致谢 在本文工作的研究过程中,得到了洪家荣教授、刘叙华教授、石纯一教授和黄厚宽教授的指点,得到了李忠凯博士和孙吉贵博士的建议与帮助,为此作者衷心感谢。

参考文献

[1] 李爱中,刘叙华,基于组合-分解的机器发现方法,软件学报(待发表)

[2] 刘叙华,基于归结方法的自动推理,科学出版社,1994

[3] 李爱中,模型发现和智能决策支持系统工具研究,哈尔滨工业大学博士学位论文,1991

[4] Kodratoff, Y., Recent Advances in Machine Learning, International Journal of Pattern Recognition and Artificial Intelligence, Vol. 6, No. 4, 1992

[5] Angluin, D., Inductive Inference: Theory and Methods, Computing Surveys, Vol. 15, No. 3, 1983

[6] Michalski, R. S., A Theory and Methodology of Inductive Learning, Artificial Intelligence, Vol. 20, 1983

(上接第 36 页)

[5] Katsuno H., Mendelzon A. O., On the difference between updating a knowledge base and revising it. In: Proceedings KR-91, 1991

[6] Grahne G., Updates and counterfactuals, same to [5]

[7] Nebel B., Belief revision and default reasoning: Syntax-based approaches, same to [5]

[8] Winslett M., Updating Logical Databases, Cambridge University Press, Cambridge, England, 1990

[9] Chen W. T., Liu L. L., A parallel approach for theorem proving in propositional logic. Information Sciences, 1987, 41