第 6 卷第 2 期
1995 年 2 月

软 件 学 报
JOURNAL OF SOFTWARE

Vol. 6, No. 2
Feb. 1995

超越表达式类上的代数化简器*

李岳峰　刘大有

（吉林大学计算机科学系，长春 130023）

摘要　超越表达式的化简是很困难的计算机代数符号化简问题. 本文提出了一种新的超越表达式类上的代数符号化简方法，这种方法是基于爬山策略的和替换法；作为应用实例，本文还用该方法对标准三角表达式类进行了化简. 目前该种化简方法已经应用于计算机代数化简系统 CASS1.

关键词　代数化简，爬山法，和替换.

代数符号化简是计算机代数的一个重要内容，代数符号化简问题有两方面的含义：获得一个等价的简单体与计算等价对象的唯一表示.

设 T 是一个对象集，$\sim$ 是 T 上的一个等价关系，称 S 是一个关于 $\sim$ 的化简器，当且仅当 S 是从 T 到 T 的一个可计算映射并且满足下面两个条件：

对任意的 $t\in T$；　$S(t)\sim t$　　(1)

$S(t)\leqslant t$　　(2)

集合 T 的一个规范化简器 S 是一个有效的过程 S，且满足以下两条：

对任意的 $s,t\in T$；　$S(t)\sim t$　　(1)

$s\sim t\Rightarrow S(s)=S(t)$　　(3)

这时称 $S(t)$ 是 t 的规范形式.

定理 1. 设 T 是一个可判定的个体的集合，$\sim$ 是 T 上的一个等价关系. 则 $\sim$ 是可判定的，当且仅当存在一个关于 $\sim$ 的规范化简器 S（证明见文献[1]）.

定理 2. 设 T 是一个可判定个体的集合，R 是 T 上的可计算二元运算，$\sim$ 是 T 上的一个等价关系且是关于 R 的同余关系，S 是关于 $\sim$ 的规范化简器，定义：

$Rep(T):=\{t\in T: S(t)=t\}$，对任意 $s,t\in Rep(T)$ 令 $R'(s,t):=S(R(s,t))$，则有 $(Rep(T),R')$ 与 $(T/\sim,R/\sim)$ 是同构的，且 $Rep(T)$ 是可判定的，R' 是可计算的. 这里 $R/\sim(Cs,Ct):=C_{R(s,t)}$，其中 Ct 是关于 $\sim$ 的 t 的同余类（由定理 1 不难证明该定理）.

上述定理表明，等价关系的规范化简器同余对应于一个可计算操作，即能用算法描述该化简过程.

---

* 本文 1992－10－23 收到，1993－01－12 定稿

作者**李岳峰**，1964 年生，讲师，主要研究领域为计算机代数，算法分析，分布式人工智能. **刘大有**，1942 年生，教授，主要研究领域为专家系统及其开发环境，分布式人工智能，不精确推理，算法分析，神经元网络.

本文通讯联系人：李岳峰，长春 130023，吉林大学计算机科学系

条件(3)是一个很强的条件,实际中经常寻找一种比规范化简器稍弱的化简器.

S 称为关于～的零等价化简器,当且仅当 S 是从 T 到 T 的一个可计算映射且满足:

对任意的 $t\in T$;　$S(t)\sim t$　　(1)

$$t\sim 0\Rightarrow S(t)\equiv S(0)\qquad(4)$$

人们已经证明有理表达式类上存在规范化简器[1],非嵌套根式类上亦存在规范化简器[2]. CASS1 系统[3]中我们已实现了这两类化简器.

定理 3. (Caviness, Richardson, Matiyasetive 1968, 1970)设 $R2$ 是超越表达式类(包含超越函数符号 exp,log,sin 等的表达式称为超越表达式),$\sim_R$ 是 $R2$ 上的等价关系. 则对任意的 $t\in R2$,谓词"$t\sim_R 0$"是不可判定的.

由定理 3 和定理 1 我们看到,超越表达式类上不存在规范化简器. 目前为了找到较通用的化简方法往往要在表达式类上加一定的限制. 比如 Johnson 的归约方法,该方法只能处理那些在其上定义了"特征元素"概念的表达式类;基于代数扩张的结构理论法要求表达式类 T 能代数扩张成域 K,使 T 中的所有项可看成 K 里的一个有理项(比如非嵌套根式类上的化简[2]);而基于 Kunth 的重写规则的规则系统法则要求表达式类 T 中的等价关系～,关于 T 的关键对集是可判定的. 本文所给出的化简方法基于算法设计理论中的爬山策略,并要求所处理的表达式类上存在可用以替换的公理(可导出和替换公式),且公理替换的结果按定义的量度标准是可行的.

1　项、替换的基本定义

一个位函数是一个函数 $a:F\to N$,N 是非负整数集,F 的元素称为函数或运算符号. 如果对 $f\in F(a)$ 有 $a(f)=0$,则称 f 是一个常量,而 $F(a,n):=\{f\in F(a):a(f)=n\}$,即为 n 元函数符号的集合. 一个 a 代数是关于集合 M 的函数族 A,使得由函数符号 $f\in F(a)$ 定义的函数 $A(f)$ 是 $M^{a(f)}$ 到 M 上的映射. M 称为 A 的承载子,记为 $C(A)$.

设 X 是一个与 $F(a)$ 不相交的集合(X 通常被称为变量的集合). a 代数 $T(X,a)$(称为项代数)的承载子是满足如下条件的最小集合 T':

$x\in X\Rightarrow x\in T'$;

$t_1,\cdots,t_n\in T'$,且 $f\in F(a,n)\Rightarrow ft_1\cdots t_n\in T'$

(这里 $x,t_1,\cdots,t_n,ft_1\cdots t_n$ 表示符号串). 进一步对任意的 $f\in F(a,n)$,函数 $T(X,a)(f)$ 定义为:

$$T(X,a)(f)(t_1,\cdots,t_n)=ft_1\cdots t_n,$$

其中 $t_1,\cdots,t_n\in T'$. 以下称 T' 的元素为项.

一个替换 σ 是从 $C(T(X,a))$ 到自身的同余映射,且满足对任意的 $x\in X$ 有 $\delta(x)=x$.

如下我们给出函数等价的概念. 设 A 是一个 a 代数,则 A 上的一个赋值是 $C(A)$ 到 A 的同余映射 V. 如果 t 是项,则 $V(t)$ 称为 t 的值. 令 X 是一个可数的集合,t 是项,t 中出现的变量集为 $\{X_1,X_2,\cdots,X_n\}$. n 元函数 $Fun(A,n,t)$ 定义为:

$$Fun(A,n,t)(a_1,\cdots a_n)=V(t),$$

其中 $a_1,\cdots a_n\in C(A)$,V 是一个赋值,满足 $V(X_1)=a_1,\cdots,V(X_n)=a_n$.

设项 t_1 和 t_2 的变量集都为$\{X_1,X_2,\cdots X_n\}$则 t_1 和 t_2 函数等价，当且仅当：

$$Fun(A,n,t_1)=Fun(A,n,t_2).$$

2 基于和替换的化简器

表达式的化简过程中，经常使用如上节定义的替换 σ(例如重写规则). 本节我们仅考虑和替换，比如下式

$$AE+BB'+2C\to 0 \tag{5}$$

即为一个和替换(和替换的右端不一定总为零，下面我们总是假定和替换的右端比左端"简单"，这里的"简单"是相对于爬山法中的量度标准而言).

如上所述的和替换与规则替换是有区别的，规则替换要求严格匹配，而和替换只是近似匹配. 例如，考虑如下表达式的和替换.

$$(3X^3+2X^2-5)AED+(3X^3+4X^2+4X)BB'D+(6X^3+8X^2+5X-9)CD$$

如果对"最大模式"X^3 项使用和替换公式(5)，则所使用的替换是严格匹配，即

$$(3AED+3BB'D+6CD)X^3\to 3DX^3(AE+BB'+2C)\to 0$$

而考虑次"最大模式"X^2 项时，则和替换公式(5)就不满足严格匹配，而是采用所谓的近似匹配. 此时要根据"决策序列"和"权值"来处理表达式

$$2DX^2(AE+2BB'+4C).$$

这里的"决策序列"和"权值"是使用者根据实际情况提供的. 比如上述表达式，它的不匹配系数表为(1,2,4)，如果不考虑"决策序列"而仅考虑"权值"(如认为 AED 较复杂应优先去掉)，则和替换的结果为(0,1,2)，即：

$$2DX^2(AE+2BB'+4C)\to 2DX^2(BB'+2C);$$

如果"决序序列"要求去掉两项，则化简的结果为(−1,0,0)，即

$$2DX^2(AE+2BB'+4C)\to -2AEDX^2.$$

算法实现中，对于"决策序列"一般考虑两方面的因素. 一是来自用户的要求，可能用户需要形式化的化简结果(而不是从最优量度标准上考虑)；二是考虑下次循环时，处理较小模式时能得到更优的处理结构(这时和替换公式中通常包括变量，例如下节三角表达式的和替换公式).

设 T 是一表达式集，$\Sigma\alpha_i a_i(x_1,\cdots x_n)\to\Sigma\beta_j b_j(x_1,\cdots x_n)$是一个和替换，其中 $a_i(x_1,\cdots x_n)$ 和 $b_j(x_1,\cdots,x_n)$是项，α_i 和 β_j 是常量. 对于 T 上给定的一个量度标准，一般来说我们可用函数 $\Gamma:T\to R_{\geqslant 0}$($R_{\geqslant 0}$是非负实数集)表示. 对任意的 $s,t\in T$，如果 $\Gamma(s)<\Gamma(t)$，则称表达式 s 比 t 简单. 一般对 Γ 的限制如下：

对任意的非零常量 h 和 $s,t,t_1,t_2\in T$，且 $\Gamma(t_1)<\Gamma(t_2)$，$s\neq 0$；

$$\Gamma(hs)=\Gamma(s) \tag{6}$$

$$\Gamma(s*t_1)<\Gamma(s*t_2) \tag{7}$$

$$\Gamma(s\pm t)\leqslant\Gamma(s)+\Gamma(t) \tag{8}$$

定义 1. 设 t_1 和 t_2 属于 T，如果 $\Gamma(t_1)<\Gamma(t_2)$，并且 t_1 和 t_2 函数等价，则称表达式 t_1 是 t_2 的简化.

如下我们给出基于和替换化简的抽象化控制过程.

算法 GS(基于爬山策略的和替换化简器)输入的表达式 $t\in T$,本算法仅处理有一个和替换公式 $\Sigma\alpha_i a_i(x_1,\cdots,x_n)\rightarrow\Sigma\beta_j b_j(x_1,\cdots,x_n)$ 的情况,且要求 $\Gamma(\Sigma h_i a_i)=\Sigma\Gamma(h_i a_i)$,$h_i$ 是非零常数.

(1)如果不存在匹配模式,则算法结束.

(2)寻找最大模式项使 $t:=P_mQ_m+P_{m-1}Q_{m-1}+\cdots+P_1Q_1+P_0$. 其中 P_m 是模式,$P_j(1\leqslant j\leqslant m)$ 是关于 $\{x_1,\cdots,x_n\}$ 的项,$Q_j=\Sigma\gamma_{ij}a_i$ 是 $\Sigma\alpha_i a_i$ 的近似和式,并且 $\Gamma(P_m)\geqslant\Gamma(P_{m-1})\geqslant\cdots\geqslant\Gamma(P_1)$

(3)由权值和化简策略形成决策序列 φ_i,并化简 $P_mQ_m\rightarrow P_m*(h\Sigma\beta_j b_j-\Sigma\varphi_i a_i)$. 其中 h 是一常量

(4)$P_0\leftarrow P_0-P_m*\Sigma\varphi_i a_i$.

(5)处理 $P_m*\Sigma\beta_j b_j$,这里展成步骤 2 的形式要考虑 P_0 中的项.

如果 $P_m*\Sigma\beta_j b_j$ 能展成步骤 2 的形式,则 $GOTO$ 2;否则,对每个 $P_m*\beta_j b_j$,如果能合并到 $P_kQ_k(m-1\leqslant k\leqslant 1)$ 中,则合并,否则置 $P_0\leftarrow P_0+P_m*\beta_j b_j$.

(6)$m\leftarrow m-1$. 如果 $m\leqslant 0$,则算法结束,否则 $GOTO$ 3. □

算法 GS 的时间复杂性主要在于如何把 t 展开. 如果和替换公式有很多项,则确定 φ_i 也比较费时,并且处理 $P_m*\Sigma\beta_j b_j$ 可能使 m 变大,即可能出现组合爆炸问题,不过实际中这样的情况很少发生.

定理 4. 如果算法 GS 中的决策序列 φ_i 使得 $\Gamma(\Sigma\varphi_i a_i)<\Gamma(\Sigma\alpha_i a_i)-\Gamma(\Sigma\beta_j b_j)$,则过程 GS 是化简器,且如果使用过一次和替换,则 $GS(t)$ 即是 t 的简化.

证明:由 Birkhoff 定理知,替换不影响表达式的函数等价性,因此如果 $\sim$ 是 T 上的函数等价关系,则有 $GS(t)\sim t$,即(1)式成立.

如果算法 GS 没有使用和替换公式,则显然 $\Gamma(GS(t))\leqslant\Gamma(t)$,即(2)式成立.

又由(8)和(6)我们有

$$\begin{aligned}\Gamma(h\Sigma\beta_j b_j-\Sigma\varphi_i a_i)&\leqslant\Gamma(h\Sigma\beta_j b_j)+\Gamma(\Sigma\varphi_i a_i)=\Gamma(\Sigma\beta_j b_j)+\Gamma(\Sigma\varphi_j a_i)\\&<\Gamma(\Sigma\beta_j b_j)+\Gamma(\Sigma\alpha_i a_i)-\Gamma(\Sigma\beta_j b_j)=\Gamma(\Sigma\alpha_i a_i)=\Sigma\Gamma(\alpha_i a_i)=\Sigma\Gamma(a_i)\\&=\Sigma\Gamma(\gamma_{im}a_i)=\Gamma(\Sigma\gamma_{im}a_i)=\Gamma(Q_m)\end{aligned}$$

再由(7)式得 $\Gamma(P_m*(h\Sigma\beta_j b_j-\Sigma\varphi_i a_i))<\Gamma(P_mQ_m)$,即说明 $\Gamma(GS(t))<\Gamma(t)$,$GS(t)$ 是 t 的简化. □

上述算法隐含假定 $\Sigma\alpha_i a_i$ 的近似和式的各项系数非零. 有时还可要求近似和式的若干项系数是零,此时对决策序列的限制更严,一般作法是先配项,然后作类似的处理.

3　三角表达式的化简

定义 2. 数、任意变量、正余弦函数本身以及它们的乘积称为单项式. 单项式中正余弦次数之和称为单项式的度.

定义 3. 标准三角表达式是单项式或若干单项式按度降幂排列之和. 设 s 是具有 m 项单项式的标准三角表达式,则整数表 $(d_1,d_2,\cdots,d_m)$ 称为 s 的特征表,其中 $d_1,d_2,\cdots,d_m$ 是由小

到大(按度)单项式的度.

定义 4. 设 s 是一个标准三角表达式,$(d_1,d_2,\cdots,d_m)$是 s 的特征表,则称:

$$(\alpha(m)-1)+\sum_{i=1}^{m}(d_i+\alpha(d_i))$$

为表达式 a 的公式大小,其中:

$$\alpha(x)=\begin{cases}0 & \text{当 } x=0,1 \text{ 时}\\ 1 & \text{当}\geqslant 2 \text{ 时}\end{cases}$$

标准三角表达式的公式大小即为上节的 Γ,它用来量度表达式的复杂程度,且满足(6)、(7)和(8)式以及算法 GS 中的约定.

三角表达式的化简中,除了使用和角公式、诱导公式和计值公式之外,还要使用和替换公式 $\sin^2x+\cos^2x=1$. 即如下的和替换公式[4].

$$\text{sincos}(x,p,q)+\text{sincos}(x,p-2,q+2)-\text{sincos}(x,p-2,q)\rightarrow 0 \tag{9}$$

其中,$sincos(x,p,q)=\sin^px*\cos^qx$. 由于后一项的公式大小小于前面项的,故对如下的表达式 s 化简如下:

$$\begin{aligned}s&=\text{sincos}(x,6,0)+3\text{sincos}(x,4,2)+3\text{sincos}(x,2,4)\\&\rightarrow\text{sincos}(x,4,0)+2\text{sincos}(x,4,2)+3\text{sincos}(x,2,4)\\&\rightarrow\text{sincos}(x,4,0)+2\text{sincos}(x,2,2)+\text{sincos}(x,2,4)\\&\rightarrow\text{sincos}(x,2,0)+\text{sincos}(x,2,2)+\text{sincos}(x,2,4)\end{aligned}$$

这里,为了形式简单我们没有把 s 展成算法 GS 步骤 2 的形式. 而决策序列的选择主要考虑和替换后的表达式的 Γ 值能达到最小.

令 $s_1=\text{sincos}(x,2,0)+\text{sincos}(x,2,2)+\text{sincos}(x,2,4)=\sin^2x(1+\cos^2x+\cos^4x)$,$s_1$ 的公式大小为 15. 又 $s_2=1-\cos^6x$ 是 s 的简化,且 s_2 的公式大小为 $7<15$.

为了得到更好的化简算法,这里我们考虑公式(9)的变形公式(基于折叠思想):

$$\begin{aligned}&\text{sincos}(x,p,q)+3\text{sincos}x(x,p-2,q+2)+3\text{sincos}(p-4,q+4)\\&\qquad+\text{sincos}(p-6,q+6)\rightarrow\text{sincos}(x,p-6,q)\end{aligned}\tag{10}$$

和替换公式(10)的左端有四项,因而在表达式的和替换中很难出现该公式的近似公式,此时往往根据决策序列进行配项. 例如上述的 s.

$$\begin{aligned}s&=\text{sincos}(x,6,0)+3\text{sincos}(x,4,2)+3\text{sincos}(x,2,4)\rightarrow_{(\text{配项})}\text{sincos}(x,6,0)\\&\quad+3\text{sincos}(x,4,2)+3\text{sincos}(x,2,4)+\text{sincos}(x,0,6)-\text{sincos}(x,0,6)\\&\rightarrow\text{sincos}(x,0,0)-\text{sincos}(x,0,6)\\&=1-\cos^6x.\end{aligned}$$

标准三角表达式的化简中和替换公式(9)和(10)经常交替使用. 什么情况下使用公式(10)将由具体的表达式决定,而一般的策略是出现对称项(对称是对正弦和余弦而言)或缺少明显对称项时(需配项)使用.

4 结束语

处理三角表达式的化简,和替换法有明显的优点. 其它方法如规则系统法和基于代数扩张的结构理论法都有明显的缺点. 设 t 是一个标准三角表达式,如果作规则变换 $\sin x\rightarrow X$,

$\cos x \to Y$,则可建立标准三角表达式与二变元变项式之间的一一对应.

定义 5. 多项式 P 和 Q 形成 E 的关键对,当且仅当存在多项对$(a_1,b_1),(a_2,b_2)\in E$,使:$P=\delta_1 * b_1$ 且 $Q=\delta_2 * b_2$,这里 E 是一些多项式对组成的集合,δ_1,δ_2 是单项式且 $\delta_1 * a_1=\delta_2 * a_2$是 a_1 和 a_2 的最小公倍式.

定理 5. S_E 是 E 的规范化简器,当且仅当对所有 E 的关键对(P,Q)有 $S_E(p)=S_E(Q)$[8].

设 $E=\{(X^2,1-Y^2)\}$,则由定理 5 知 S_E 是规范化简器. 比如

$t=\text{sincos}(x,4,2)+\text{sincos}(x,6,0)\to X^4Y^2+X^6\to_E((1-Y^2)^2Y^2+(1-Y^2)^3\to 1-2Y^2+Y^4\to 1-2\text{sincos}(x,0,2)+\text{sincos}(x,0,4)=1-2\cos^2x+\cos^4x$;而 $GS(t)=\sin^4x$. 由此我们看到规则系统法虽然化简速度快,但容易出现正弦或余弦符号过多的情形. 而使用结构理论法即要找到适当的扩张域,如果用指数来代替正、余弦函数,这样做一方面把一个难的问题转换成另一个难的问题,另一方面也完全脱离了三角函数的自身特点. 目前,用基于爬山策略的和替换法化简三角表达式,我们认为是最好的方法.

参考文献

1 Buchberger B, Loos R. Algebraic simplification. Springer－Verlay, 1982.

2 Caviness B F, Fateman R. Simplification of radical expressions. SYMSAC, 1976.

3 李岳峰. 计算机代数化简系统 CASS1. 软件学报,1993,4(1):37－42.

4 Hornfeldt L. A sun－substituter used as trigonometric simplifier. EUROCAM'82, 1982.

5 Caviness B F. Computer algebra: past and future. Symbolic Computation, 1986.

6 Fateman R J. MACSYMA'S general simplifier: philosophy and operation. MACSYMA, 1979.

7 STEPHANE KAPLAN. Simplifying conditional term rewriting systems: unificaition, termination and confluence. J. Symbolic Computation, 1987.

8 Li Yuefeng, Chen Jianhua. A computer aided instruction system ASAC－2. BISYCP'89, 1989. 163－165.

9 李岳峰. CSCAS:规范化简的计算机代数系统. CJCAI'90, 1990. 149－154.

ALGEBRAIC SIMPLIFIER OF TRANSCENDENTAL EXPRESSIONS

Li Yuefeng　Liu Dayou

(*Department of Computer Science, Jilin University, Changchun* 130023)

Abstract　Simplification of transcendental expression is one of the difficult simplification problem in computer algebra. In this paper, the authors present an algebraic simplification method of transcendental expressions, and the method is a sum substitution strategy based on hill－climbing approach. An important example of the usefulness of this method is found in simplification of normal triangle expressions. Now this simplification method has already been used in computer algebraic simplification system CASS1.

Key words　Algebraic simplification, hill－climbing approach, sum substitution.