计算机代数化简系统 CASS1

李岳峰

(吉林大学计算机科学系,长春 130023)

COMPUTER ALGEBRAIC SIMPLIFICATION SYSTEM CASS1

Li Yuefeng

(*Jilin University*, *Changchun* 130023)

Abstract The computer algebra system CASS1 is designed by algebraic simplification method. In this survey, we only discuss some important algorithms for the CASS1, including rational expression canonical simplification algorithm, unnested radical expression canonical simplification algorithm and one species transcendental expression simplification algorithm. As an attempt at computer algebra system canonical construction, CASS1 was implemented by algorithm description language ALDES[5].

摘要 CASS1是用代数化简方法设计的计算机代数系统. 本文讨论CASS1的几个重要算法,包括:有理表达式的规范化简算法、非嵌套根式表达式的规范化简算法以及一类超越表达式的化简算法等. CASS1是用算法描绘语言ALDES[5]语言实现的,是计算机代数系统建造规范化的一种尝试.

§1. 引 言

计算机代数系统的出现已有二十多年的历史,目前最有代表性的计算机代数系统是[1]: MACSYMA, REDUCE, ALDES/SAC－2和mu－MAIH.

MIT研制的MACSYMA系统是当今能为数学提供最强有力工具的计算机代数系统. 因为MACSYMA能处理初等数学,因此,用户通过它能展开极复杂的问题.

REDUCE是最早强调可移植性的计算机代数系统,它处理常、偏微分方程的问题很有特色.

计算机代数系统ALDES/SAC－2有更高的可移植性和更大的可用性. 例如,我们已把该系统移植到吉林大学计算机科学系的VAX－11/750上. ALDES/SAC－2主要处理多项式和矩阵运算,这个系统最重要的系统特征是包含很多详细算法和完整文件的一个形式系统.

本文1990年10月25日收到,1991年2月8日定稿. 作者李岳峰,讲师,1988年硕士毕业于吉林大学,主要研究领域为计算机代数、算法.

建造一个计算机代数系统是相当艰辛的，例如MACSYMA的建造就花费了一百多人年的时间，由大约三十万行的编译LISP码组成.计算机代数系统中算法的实现，多采用"试探性"的方法[9]或符号代数计算的办法(如ALDES/SAC—2)，并且相应的算法仅可应用于一类问题的某些实例.计算机代数化简系统CASS1的建造中，感兴趣的是寻找一类问题的解算法.这里，我们以计算机代数化简的方法，寻找相应表达式类的规范化简器，来实现各子系统的功能.以这种方式建立的各子系统具有一种框架的性能，达到了更大的可用性.

代数化简问题有两个方面的意义：获得一个等价的简单体与计算等价对象的唯一表示.

设集合T是一个表达式集，$\sim$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T到T的一个可计算映射，并且满足下面两个条件：

对任意$t\in T$；

$$S(t)\sim t \tag{1}$$

$$S(t)\leqslant t \tag{2}$$

这里"$\leqslant$"是简化的意思.

获得一个等价的简单体的问题就是要寻找满足(1)、(2)的过程S.

定义过程S是集合T的一个规范化简器，当且仅当对任意的$s,t\in T$满足：

$$S(t)\sim t \tag{3}$$

$$s\sim t\rightarrow S(s)=S(t) \tag{4}$$

计算等价对象的唯一表示问题是要寻找一个规范化简器.

§2. CASS1的系统结构

CASS1系统由四个子系统组成，它们是CASS1的基本系统、有理表达式系统、根式表达式系统和超越表达式系统.在计算机代数系统ALDES/SAC－2的基础上，我们扩充了ALDES/SAC－2的一些功能，实现了CASS1的基本系统[9].

ALDES/SAC－2是由11个子系统组成，这些子系统与CASS1的子系统之间的调用如图1所示.这里，系统的整数要求是无限精度的，即任意整数A有如下形式：$A=\sum_{i=1}^{n-1}\alpha_i\beta^i$.其中，$|\alpha_i|<\beta,\beta\geqslant 2$是此无穷精度算数的基数.我们的实现中$\beta=2^{32}$.

CASS1基本系统由45个子算法组成，它使用ALDES/SAC－2的基本系统，表处理系统和符号处理系统，CASS1的基本系统为其它子系统提供符号演算的功能.

§3. 有理表达式及非嵌套根式的规范化简器

文[2]中，证明了有理表达式类上存在规范化简器.规范化简器RECSAL是有理表达式系统的算法，它的实现过程如下：

算法*RECSAL*(有理表达式规范化简器)s是任意给定的有理表达式，有理表达式t是s的规范形式.

(1)[化s为两个多项式的商，不妨设$sp=p'(X_1,X_2,\cdots,X_n)/q'(X_1,X_2,\cdots,X_n)$.$p'$、$q'$是有理多项式]

$sp=SRETPO(s)$.

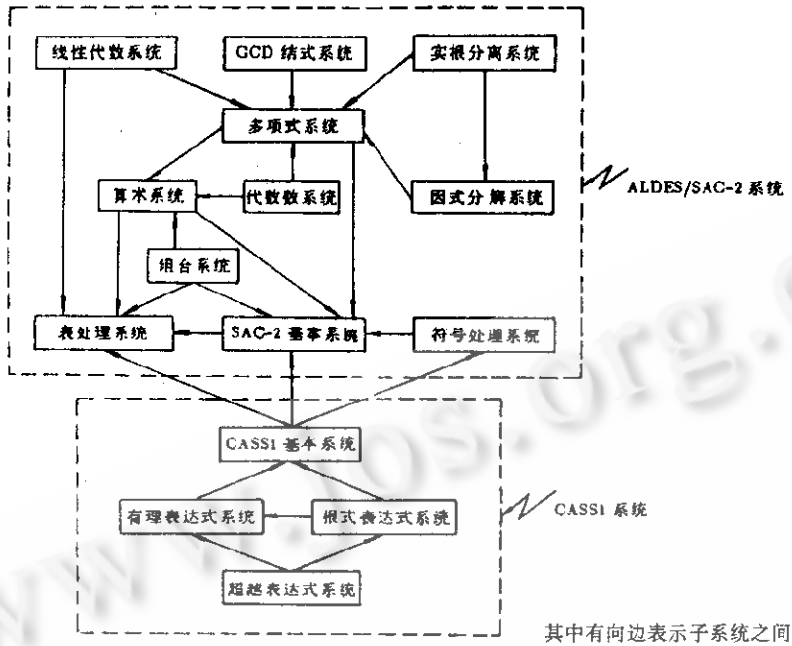


图 1　CASS1 与 ALDES/SAC－2 之间的关系图

(2)[求 p'、q'的分布式多项式]

$p=POEXP(p')$. $q=POEXP(q')$.

(3)[求 p 和 q 的 GCD]

$IPGCDC(n,p,q.c,p_1,q_1)$.

(4)[c 为公因式,$p_1=p/c,q_1=q/c$]

$t=(p_1,q_1)$. ■

这里,过程 $SRETPO$、$POEXP$ 以及 $IPGCDC$ 都是有理表达式系统的算法;令 R 是一个系数环,$A(X_1,X_2,\cdots,X_r)=\sum_{i=1}^{m}a_iX^{e_i}\in R[X_1,X_2,\cdots,X_r]$,$r\geqslant 1$,且每个 $a_i\neq 0$,$e_i=(e_{i1},e_{i2},\cdots,e_{ir})$. 如果假定 $e'_1>e'_2>\cdots>e'_m$,其中 e'_i 是 e_i 的逆,则多项式 A 的分布式多项式定义为$(a_1,e'_1,\cdots,a_m,e'_m)$. 例如:

$$(X-X/X^2)/(X+1)\rightarrow(X/1-1/X)/((X+1)/1)\rightarrow$$
$$((X^2-1)/X)/((X+1)/1)\rightarrow(X-1)/X.$$

根式表达式系统调用有理表达式系统.

定义 3.1:设 Q 是有理数集,RE 是关于变量$(X_1,X_2,\cdots,X_n)$的有理表达式集,则称集合$\{s^r:s\in RE,r\in Q\}$为$(X_1,X_2,\cdots,X_n)$上的根式表达式类. 如果根式表达式的根号内没有根号作为其子项,则称为非嵌套根式.

非嵌套根式的等价关系是通过函数的亚纯相等建立起来的. 用结构理论法,Caviness 和 Fateman 于 1976 年给出了非嵌套根式的规范化简算法[2]. 非嵌套根式的规范化简器 NEC-SAL 由下面算法描绘.

算法 *NECSAL*(非嵌套根式的规范化简器)$t_1,t_2,\cdots,t_l$ 是关于$(X_1,X_2,\cdots,X_n)$的非嵌套根式. $s_1,s_2,\cdots,s_l$ 是 $t_1,t_2,\cdots,t_l$ 的规范形式.

(1)*FOR* $i=1$ *TO* l *DO* $s_i=t_i$.

(2)[用规则$(p/q)^{-r}\to(q/p)^r$及有理化简处理 s_i]

FOR $i=1$ *TO* l *DO* $s_i=TRANSFORM1(s_i)$.

(3)[求 $s_1,s_2,\cdots,s_l$ 中所有形如$(p/q)^r$项的集合]

$R=RADICALS(s_1,s_2,\cdots,s_l)$.

(4)[求 R 中项的被开方多项式的集合]

$P=PRAS(R)$.

(5)[求 P 的一个基]

$B=ABFP(P)$.

(6)*IF* $ISENP(P)$ *THEN* $B=B\cup\{-1\}$.

$M=NUMBER(B)$.

(7)*FOR* $i=1$ *TO* l *DO* $s_i=TRANSFORM2(s_i)$

(8)[为排序需要计算 B 中所有元素 b 的根式度 $D(b)$. $u_1/v_1,\cdots,u_k/v_k$ 是 b 出现在 $s_1,s_2,\cdots,s_l$ 中的所有约化有理幂]

FOR $\forall b\in B$ *DO* $D(b)=LCM(v_1,v_2,\cdots,v_k)$.

(9)*FOR* $i=1$ *TO* l *DO* $s_i=TRANSFORM3(s_i)$.

(10)[决定规范形式]

FOR $i=1$ *TO* l *DO* ($s_i=RECSAL(s_i)$.

$s_i=TRANSFORM4(s_i)$). ■

这里,除 *RECSAL*,*NUMBER*(求集合元素的个数)和 *LCM*(求最小公倍数)外,其余的过程都是根式表达式系统的算法. *ABFP* 是求 P 的一个基,即度$\geqslant 0$的$(X_1,X_2,\cdots,X_n)$上的不可约多项式的集合,且满足:如果$b\in B$,则存在$p\in P$使$b\mid p$,且如果$p\in P$,则$p=\pm\prod b_i^{e_i}$,$b_i\in B$,$e_i\in N$. $ISENP(P)$是布尔函数,当 P 中存在一个非正的元素时 $ISENP(P)=true$. $TRANSFORM2(s_i)$是用如下规则变换 s_i:$(p/q)^r\to p^r/q^r$;$(\prod b_i^{e_i})^r\to\prod(b_i^{e_i*r})$;$(-\prod b_i^{e_i})^r\to(-1)^r*\prod(b_i^{e_i*r})\Rightarrow(b^w)*(b^{u/v})$,如果$r=w+u/v$,$w\in N$,$0\leqslant u/v<1$且 u、v 互质. $TRANSFORM3(s_i)$是用如下规则变换 s_i:$b_j^{(u/v)}\to(b_j^{(1/D(b_j))})^{(u/v)*D(b_j)}$;$b_j^{(1/D(b_j))}\to y_j$. $TRANSFORM4(s_i)$是用规则 $y_j\to b_j^{(1/D(b_j))}$变换 s_i. 例如:

$t_1=(((2*X-2)/(X^3-1))^{-7/3}+(2/(X+1))^{1/2})/(24*X+24)^{1/4}$

(2):$s_1=(((X^2+X+1)/2)^{7/3}+(2/(X+1))^{1/2})/(24*X+24)^{1/4}$

(3):$R=\{(X^2+X+1,2),(2,X+1),(2^3*3*(X+1),1)\}$

(4):$P=\{X^2+X+1,2,X+1,2^3*3*(X+1)\}$

(5):$B=\{2,3,X+1,X^2+X+1\}$

(7):$s_1=(((X^2+X+1)^2*(X^2+X+1)^{1/3})/(2^2*2^{1/3})+(2^{1/2}/(X+1)^{1/2}))/(2^{5/4}*3^{1/4}*(X+1)^{1/4})$

(8):$D(2)=12$

$D(3)=4$

$D(X+1)=4$

$D(X^2+X+1)=3$

(9):$s_1=(((X^2+X+1)^2*y_4)/(4*y_1^4)+y_1^6/y_3^2)/(y_1^9*y_2*y_3)$

(10):$s_1=((X^2+X+1)^2*y_3^2*y_4+4*y_1^{10})/4*y_1^{13}*y_2*y_3^3$

$=((X^4+2*X^3+3*X^2+2*X+1)/(48*X+48))*2^{11/12}*3^{3/4}*(X+1)^{3/4}*(X^2+X+1)^{1/3}+(1/(6*X+6))*2^{9/12}*3^{3/4}*(X+1)^{1/4}$.

§4. 一类超越表达式的化简器

超越表达式系统主要处理一类三角表达式,包含超越函数符号(如 *exp*,*log*,*sin* 等)的表达式称为超越表达式.由于超越表达式类上不存在规范化简器[2],因此,为找到较通用的方法,往往要在表达式类上加一定的限制.现在,关于超越表达式类上的化简方法已有很多,如 Johnson 的归约方法、基于代数扩张的结构理论法、规则系统法以及点计值方法等.CASS1 中,我们提出了标准三角表达式类上的三角化简器的实现方法.

定义 4.1: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 4.2: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 4.3: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 s 的公式大小,其中:$\alpha(x)=\begin{cases}0 & \text{当 } x=0,1 \text{ 时;}\\ 1 & \text{当 } x\geqslant 2 \text{ 时.}\end{cases}$

设 t_1、t_2 是两个标准三角表达式,称 t_1 是 t_2 的简化,如果 t_1 的公式大小严格小于 t_2 的公式大小并且 t_1 和 t_2 函数相等.三角表达式化简的理论很少,L. Hornfeldt[4]提出过一种和替换方法,即用如下的和替换公式:

sincos(x,p,q)+sincos(x,p−2,q+2)−sincos(x,p−2,q)→0,其中 sincos(x,p,q)$=(\sin x)^p*(\cos x)^q$.

例如:s =sincos(x,6,0)+3sincos(x,4,2)+3sincos(x,2,4)

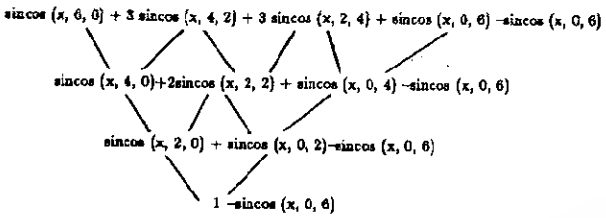
→sincos(x,4,0)+2sincos(x,4,2)+3sincos(x,2,4)

→sincos(x,4,0)+2sincos(x,2,2)+sincos(x,2,4)

→sincos(x,2,4)+sincos(x,2,2)+sincos(x,2,0).

令 s_1=sincos(x,2,4)+sincos(x,2,2)+sincos(x,2,0),则 s_1 的公式大小为 7+5+3=15.但显然 s_2=1−sincos(x,0,6)是 s 的简化,而 s_2 的公式大小为 7<15.

算法 TRSIMP 补充了和替换方法的不足,算法的详细描述在文[8]中.TRSIMP 通过找最大模式的对称项,用 TEFOAL(折叠算法)完成表达式的化简,上述表达式 s 的化简过程如下:



设 t 是一个标准三角表达式，如果做规则变换 sinx→X，cosx→Y，则可建立标准三角表达式与二变元多项式之间的一一对应.

定义 4.4：多项式 p 和 q 形成 E 的关键对，当且仅当存在多项式对 (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 3.1：S_E 是 $\rightarrow_E$ 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].

今设 $E=\{(X^2,1-Y^2)\}$，显然 S_E 是规范化简器. 例如：$t=\text{sincos}(x,4,2)+\text{sincos}(x,6,0)\rightarrow X^4 * Y^2+X^6\rightarrow_E(1-Y^2)^2 * Y^2+(1-Y^2)^3\rightarrow_E 1-2 * Y^2+Y^4\rightarrow 1-2\text{sincos}(x,0,2)+\text{sincos}(x,0,4)$. 我们已用算法 TTRANP 实现了上述过程. 从实用的角度，算法 TTRANP 不如算法 TRSIMP，因由 TRSIMP 知 $t\rightarrow(\sin x)^4$，因此我们提出了公式大小来衡量一个表达式的复杂程度.

参考文献

[1]B. F. Caviness, "Computer Algebra: Past and Future", Symblic Computation, (1986), 217—236.

[2]B. Buch berger. R. Loos, "Algebraic Simplification", Springer—Verlag(1982)4,11—143.

[3]Richard Parelle, "MACSYMA: Capabilities and Applications to Problems in Engineering and the Sciences", 18th, ACS, 1984.

[4]L. Hornfeldt, "A SUM—SUBSTITUTOR Used as TRIGONMETRIC SIMPLIFIER", EUROCAM'82.

[5]G. E. Collins, "ALDES and SAC—2 Now Available", SIGSAM Bull, 14/2,19 (1980).

[6]A. C. Heam, "REDUCE—A Case Study in Algebra System", EUROGAM'82.

[7]Yuefeng Li(李岳峰), Jianhua Chen(陈建华), "A Computer Aided Instruction System ASAC—2", BISYCP'89.

[8]李岳峰，"关于计算机代数系统建造和三角表达式化简的研究"，吉林大学研究生论文集刊，1988 年第四期.

[9]李岳峰，"CSCAS：规范化简的计算机代数系统"，第一届中国人工智能联合学术会议论文集，(1990).