母関数を用いたプログラム変換

3 E − 5

川本史生
早稲田大学大学院 理工学研究科

辰己 丈夫
早稲田大学 理工学総合研究センター

二村 良彦
早稲田大学 理工学部

1. はじめに

効率にとらわれずに分かりやすいプログラムを作成した後に，系統的方法を用いて効率の良いものにするプログラム変換という方法論が，70年代から研究されるようになってきた[1, 10]．プログラム変換の過程で数式処理を必要とする場合が多々あるが，我々は数式処理システムを利用することで変換系を効率的に開発し，さらに，数論的関数に関して従来のプログラム変換よりも高度な変換を行なう研究を行なってきた[4, 5, 9, 11]．本稿は，母関数[3, 7]を利用したプログラム変換システムの REDUCE[8]と Allegro Common Lisp (ACL)による実現とその応用について報告するものである．

2. 母関数

従来のプログラム変換では不可能であるが，母関数を利用したプログラム変換で処理することができる数論的関数（再帰関数）の例を先ずのべる．

例 1：組合せ関数[2, 4]

(1.1) C(n, m)=0 if n<m or m≦0.

(1.2) C(n, m)=1 if n=m or m=0

(1.3) C(n, m)=C(n-1, m)+C(n-1, m-1)

これを，閉じた式 n(n-1)...(n-m+1)/(m!)に変換することは従来の方法では不可能であった．

例 2：文献[7]の 297 頁演習問題 17b[4]

(2.1) D(n, m)=0 if n<m or m<0.

(2.2) D(n, m)=1 if n=m or m=0.

(2.3) D(n, m)=(n-m)D(n-1, m)+D(n-1, m-1)

これを，閉じた式(n-m)(n-1)(n-2)...(m+1) に変換することは従来の方法では不可能であった．

例 3：二村関数

(3.1) F(n, m)=0 if n<m-1.

(3.2) F(n, 0)=F(m-1, m)=1.

(3.3) F(n, m)=F(n-1, m)+F(n-1, m-1)

関数 F は以下の例で述べる興味深い性質を持っており，発見者の名前を冠してここでは二村関数と呼ぶ．この式を，閉じた式(n+1)n...(n-m+2)/(m!)に変換することは従来の方法では不可能であった．以下では関数 F の第 2 引数mを定数として，F(n, m)を m について部分計算[6]した結果を $F_m(n)$ とおく．

例 4：数列 1+2+...+n の和 $F_2(n)$

(4.1) $F_2(n)=0$ if n<1

(4.2) $F_2(1)=1$

(4.3) $F_2(n)=F_2(n-1)+F1(n-1)=F_2(n-1)+n$

例 3 の結果と合わせて，F2(n)=n(n+1)/2 が得られる．

例 5：数列 $1^2+2^2+...+n^2$ の和 $F_3(n+1)+F_3(n)$

(5.1) $F_3(N)=0$ if n<2

(5.2) $F_3(2)=1$

(5.3) $F_3(n)=F_3(n-1)+F_2(n-1)$

$=F_3(n-1)+n(n-1)/2$ if 2<n

(5.3.1) $F_3(3)=4=2^2$

(5.3.2) $F_3(n)=F_3(n-2)+n(n-1)/2+(n-1)(n-2)/2$

$=F_3(n-2)+(n-1)^2$ if 3<n

(5.3.2.1) $F_3(n)=(n-1)^2+...+3^2+1^2$ if n is even

(5.3.2.2) $F_3(n)=(n-1)^2+...+4^2+2^2$ if n is odd

従って，$F_3(n+1)+F_3(n)=1^2+2^2+3^2+...+n^2$ となる．例 3 の結果と合わせて，F3(n+1)+F3(n)=(n+2)(n+1)n/6+(n+1)n(n-1)/6=n(n+1)(2n+1)/6

Program Transformation Using Generating Function: Fumio Kawamoto†, Takehiko Tatsumi†† and Yoshihiko Futamura†††
† Graduate school of Science and Engineering Waseda University, ††Advanced Research Center for Science and Engineering, Waseda University, †††School of Science and Engineering, Waseda University

が得られる.

以上で述べた 5 つの例は全て，母関数を利用することにより系統的に解決可能である.

3. 母関数を用いてのプログラム変換

母関数を用いてのプログラム変換は，外部の数式処理系を用いることで作成が容易になる. 今回作成した母関数処理系のメインプロセスは次のようになる.

(1) 与式を unfold した後母関数に直し，fold できるように変形する
(2) fold して得られた漸化式を解く
(3) 前のステップの解（閉じた式）をべき級数展開し，係数を取り出す

各ステップにおいて，REDUCE に式を渡して解かせるようにしている. ただし，簡単なパターンマッチングなど，LISP での実装が容易な物に関しては LISP 内で処理している.

例：c(n,m)の場合

母関数 $\sum_{m=0}^{\infty} c(n,m)Z^m$ を，infsum(c(n,m),m)で表す. 第 1 引数は Z の累乗に掛けられている式で，第 2 引数は Z の指数である.

最初のステップでは，LISP 上で unfold された母関数 infsum(c(n-1,m)+c(n-1,m-1),m)が REDUCE で定義した関数 ff_solve に渡される. ff_solve は与えられた式中にある全ての infsum の引数が(c(....,m),m)となるように式変形する関数であり，この場合は infsum(c(n - 1,m),m)*(z + 1)を出力する. LISP はパターンマッチングでこの出力を fold し，G(n)=G(n-1)*(z+1)を得る.

続いてこの漸化式を解くために，REDUCE で定義した関数 rr_solve に式を渡す. rr_solve は簡単な漸化式を解くための関数であり，結果 G(n)=(z+1)**n を得る.

最後のべき級数展開では，REDUCE の User Contributed Package である FPS を使用する. FPS に (z+1)^n を渡すと，((- 1) **k*pochhammer(- n,k)) / factorial(k)が出力される. (- 1) **k*pochhammer(- n,k) = factorial(n) / factorial(n-k)であることから，閉じた式 n(n-1)...(n-m+1)/(m!)が求められたことになる.

4. おわりに

今後の課題としては，いろいろな形の数式に母関数処理を行えるようにするための前処理の研究がある. また，多引数関数を処理するために，より多くの変換規則を実装し，母関数処理の第 2 段階においてさらに母関数を呼び出せるような，より汎用性のあるモデルで処理系を作成したい.

最後に，本稿の例題 3,4,5 を我々が発見するヒントを与えて下さったシンガポール国立大学W. N. Chin 博士に深謝いたします.

参考文献

[1] Burstall R.M., Darlington J.: A Transformation System for Developing Recursive Programs, *JACM, 24, 1*, pp 44-67, 1977
[2] Chin W. N.:A transformation method for dynamic-sized tabulation, *Acta Informatica*, 32, pp.93-115, March 1995
[3] C. L. Liu: *Introduction to Combinatorial Mathematics*, McGraw-Hill, 1968
[4] 二村良彦，矢農正紀：実際的プログラム変換の例，日本ソフトウェア科学会第 14 回大会，1997 年 9 月
[5] 二村，大谷，筧，坂本，小西：3E-07 ある種の木再帰プログラムからの再帰除去，情報処理学会第 56 回全国大会論文集 3E-07，1998 年 3 月.
[6] Futamura Y, Nogi K. and Takano A.: Essence of generalized partial computation, *Theoretical Computer Science 90*, pp 61-79, 1991
[7] Graham R. L., Knuth D. E. and Patashnik O.: *Concrete Mathematics*, Addison Wesley, Reading, Massachusetts, 1989
[8] Hearn A.C.: REDUCE - A Case Study in Algebra System Development, *Lecture Notes on Comp. Science*, No. 144, Springer-Verlag, Berlin, 1982
[9] 川本，中村，小西，湯浅，二村：プログラム変換と数式処理システムの融合，C11-3，日本ソフトウェア科学会第 14 回大会論文集，1997 年 9 月
[10] Pettorossi A. and Proietti M.: Rules and Strategies for Transforming Functional and Logic Programs, *ACM Computing Surveys, Vol.28, No.2*, pp360-414, 1996
[11] 坂本，川本，小西，二村：3E-06 線形再帰プログラムからの再帰除去法の実現，情報処理学会第 56 回全国大会論文集 3E-06，1998 年 3 月