

86-88

几种递归变换的改进模式

张为群 王旭仁

罗旭东

(西南师范大学 重庆 630715)(北京轻工业学院 北京 100037)

摘要 In this paper, several improved transformation patterns for recursive schemes are presented, their correctness is proved by the structural induction, and examples are given.

关键词 Automatic programming, Program transformation, Recursive transformation.

递归变换是程序自动设计中的一个研究课题。文[3,4]分别提出了几种递归变换,其可用性条件中都包括单位元。我们基于文[8]的思想,讨论了对文[3,4]中几种变换的改进。

一、对扩展 B 型 Cooper 变换的改进

1.1 扩展一阶 B 型 Cooper 变换

在文[4]中给出了如下递归变换模式。

输入模式:

$$f(x) \equiv \text{if } b(x) \text{ then } h(x) \text{ else } Q(R(x), F(g(x), f(k(x))))$$

输出模式:

$$f(x) \equiv G(e_q, e_r, x) \\ G(z, y, x) \equiv \text{if } b(x) \text{ then } Q(z, F(y, h(x))) \text{ else } \\ G(Q(z, F(y, r(x))), F(y, g(x)), k(x))$$

可用性条件:

1) Q, F 分别满足结合律: $Q(Q(x, y), z) = Q(x, F(y, z)); F(F(x, y), z) = F(x, F(y, z))$ 。

2) F 关于 Q 满足分配律: $F(x, Q(y, z)) = Q(F(x, y), F(x, z))$ 。

3) Q, F 分别具有左单位元 e_q, e_r : $Q(e_q, x) = x; F(e_r, x) = x$ 。

4) b, h, k, g, r 中均不含有 f 。

1.2 扩展一阶 B 型 Cooper 变换的改进型

一阶 B 型 Cooper 变换的可用性条件,即 Q, F 分别具有左单位元 e_q, e_r ,为的是保证在一开始 $b(x)$ 就为真时变换的正确性。因此只要将输出模式多加一支,分离另行处理,则条件 3) 实际上可略。为此提出一阶 B 型 Cooper 变换的改进型:

输入模式:

$$f(x) \equiv \text{if } b(x) \text{ then } h(x) \text{ else } Q(r(x), F(g(x), f(k(x))))$$

可用性条件:

1) Q, F 分别满足结合律: $Q(Q(x, y), z) = Q(x, Q(y, z)); F(F(x, y), z) = F(x, F(y, z))$ 。

2) F 关于 Q 满足分配律: $F(x, Q(y, z)) = Q(F(x, y), F(x, z))$ 。

• 86 •

 $(x, y), F(x, z))$

3) b, h, k, g, r 中均不含 f 。

输出模式:

$$f(x) \equiv \text{if } b(x) \text{ then } h(x) \text{ else } G(r(x), g(x), k(x)) \\ G(z, y, x) \equiv \text{if } b(x) \text{ then } Q(z, F(y, h(x))) \text{ else } \\ G(Q(z, F(y, r(x))), F(y, g(x)), k(x))$$

1.3 改进型的正确性

引理 1 $G(z, y, x) = Q(z, F(y, f(x)))$

证: 1) 当 $b(x)$ 为真时:

$$G(z, y, x) = Q(z, F(y, h(x))) \quad [\text{展开 } G] \\ = Q(z, F(y, f(x))) \quad [\text{卷叠 } f]$$

引理成立。

2) 当 $b(x)$ 为假时, 设 $\forall t < x$, 引理成立, 即:

$$G(z, y, t) = Q(z, F(y, f(t))), k(x) < x. \text{ 于是有: } \\ G(z, y, x) = G(Q(z, F(y, r(x))), F(y, g(x)), k(x)) \quad [\text{展开 } G]$$

$$= Q(Q(z, F(y, r(x))), F(F(y, g(x)), f(k(x)))) \quad [\text{归纳假设}]$$

$$= Q(Q(z, F(y, r(x))), F(y, F(g(x), f(k(x)))) \quad [F \text{ 的结合律}]$$

$$= Q(z, Q(F(y, r(x)), F(y, F(g(x), f(k(x)))) \quad [Q \text{ 的结合律}]$$

$$= Q(z, F(y, Q(r(x), F(g(x), f(k(x)))) \quad [F \text{ 关于 } Q \text{ 分配律}]$$

$$= Q(z, F(y, f(x))) \quad [f \text{ 卷叠}]$$

由结构归纳法, 可知引理成立。

定理 1 在改进的扩展一阶 B 型 Cooper 变换中, 输出模式与输入模式等价。

证: 由输出模式有:

$$f(x) \equiv \text{if } b(x) \text{ then } h(x) \text{ else } G(r(x), g(x), k(x))$$

由引理 1 知, $G(r(x), g(x), k(x)) = Q(r(x), F(g(x), f(k(x))))$, 故 $f(x) \equiv \text{if } b(x) \text{ then } h(x) \text{ else } Q(r(x), F(g(x), f(k(x))))$, 此即输入模式, 定理得证。□

1.4 应用示例

1) 我们在集合 $S = \{f: R^+ \rightarrow R^+ | f(x) \geq 0, x \in R^+, f \text{ 为单调递增函数}\}$ 上定义一个偏序关系 $\leq$ 如下:

对于 $f, g \in S, \forall x \in R$, 若有 $f(x) \leq g(x)$, 则 $f \leq g$.

2) 定义集合 S 上的乘 $\cdot$ 为:

$$\forall f, g \in S, f \cdot g(x) = f(g(x))$$

定义集合 S 上的加法 $\oplus$ 为:

$$\forall f, g \in S, (f \oplus g)(x) = \max(f(x), g(x))$$

定义 $P: S \rightarrow S$ 为: $\forall x \in R, \forall f \in S$,

$$P(f)(x) = \begin{cases} f(x) - 1, & \text{若 } f(x) \geq 1 \\ 0, & \text{若 } f(x) < 1 \end{cases}$$

显然 $P(f) \leq f$.

根据上述定义, 我们给出如下递归算法.

输入模式:

$$f(N) \equiv \text{if } B(N) \text{ then } e_0 \text{ else } L \oplus \text{arctg} \cdot f(P(N))$$

这里 $L \in S$, 定义为 $\forall x \in R, L(x) = x^2; e_0 \in S$, 定义为 $\forall x \in R, e_0(x) = 0; B(N)$ 定义为 $N = e_0, N \in S$.

注意: $x \sim N; b \sim B(N); h \sim e_0; r \sim L; g \sim \text{arctg}; k \sim P; F \sim 0; Q \sim \oplus; \forall x \in R, e_0 = 0; \forall x \in R, e_F(x) = e(x) = x_0$ 且易知它满足可用性条件:

$$1) (f \oplus g) \oplus h = f \oplus (g \oplus h)$$

$$(f \cdot g) \cdot h = f \cdot (g \cdot h)$$

$$2) h \cdot (f \oplus g)(x) = h \cdot \max(f(x), g(x)) = h(\max(f(x), g(x))) = \max(h(f(x), g(x))) = \max(h \cdot f(x), h \cdot g(x)) = h \cdot f(x) \oplus h \cdot g(x)$$

3) 显然.

于是, 由扩展一阶 B 型 Cooper 变换改进型的输出模式有:

$$f(N) \equiv \text{if } B(N) \text{ then } e_0 \text{ else } G(L, \text{arctg}, P(N))$$

$$G(z, y, N) \equiv \text{if } B(N) \text{ then } z \text{ else } G(z \oplus y \cdot L, y \cdot \text{arctg}, P(N))$$

实例 令 $c(x) = 4, x \in R^+$

$$f(c) = G(L, \text{arctg}, P(c))$$

$$= G(L \oplus \text{arctg} \cdot L, \text{arctg}^2, P^2(c))$$

$$= G(L \oplus \text{arctg} \cdot L \cdot \text{arctg}^2 \cdot L, \text{arctg}^3, P^3(c))$$

$$= G(L \oplus \text{arctg} \cdot L \oplus \text{arctg}^2 \cdot L \oplus \text{arctg}^3 \cdot L, \text{arctg}^4, P^4(c))$$

$$= L \oplus \text{arctg} \cdot L \oplus \text{arctg}^2 \cdot L \oplus \text{arctg}^3 \cdot L$$

这里函数的幂定义为: $f^n = \underbrace{f \cdot f \cdot \dots \cdot f}_n$.

二、对 A 型两步递归算法的等价变换的改进

2.1 算法的等价变换模式

文[3]给出了如下的二步递归算法:

输入模式:

$$f(x) \equiv \text{if } b_1(x) \text{ then } h_1(x) \text{ else}$$

$$\text{if } b_2(x) \text{ then } h_2(x) \text{ else}$$

$$F_1(F_2(f(k(x)), g_1(x)), F_2(f(k^2(x)), g_2(x)))$$

其中 $k^2(x) = k(k(x)), x, k: \text{TYPE1}; k(x) < x;$

$h_1, h_2, g_1, g_2, F_1, F_2, f: \text{TYPE2}; b_1, b_2: \text{Boolean}.$

输出模式:

$$f(x) \equiv \text{if } b_1(x) \text{ then } h_1(x) \text{ else}$$

$$\text{if } b_2(x) \text{ then } h_2(x) \text{ else } G(x, e_2, e_1)$$

$$G(x, y, z) \equiv \text{if } b_2(x) \text{ then } F_1(F_2(h_2(x), y), F_2(h_1(k(x)), z)) \text{ else}$$

$$G(k(x), F_1(F_2(g_1(x), y), z), F_2(g_2(x), y))$$

可用性条件为:

$$1) F_2 \text{ 满足结合律: } F_2(F_2(x, y), z) = F_2(x, F_2(y, z)).$$

$$2) F_1 \text{ 满足 } F_1(F_1(x, y), z) = F_1(F_1(x, z), y).$$

3) F_2 关于 F_1 具有左、右分配律:

$$F_2(F_1(x, y), z) = F_1(F_2(x, z), F_2(y, z));$$

$$F_2(x, F_1(y, z)) = F_1(F_2(x, y), F_2(x, z)).$$

4) F_1, F_2 分别具有右单位元 e_1, e_2 .

$$5) b_2(x) \equiv b_1(k(x)).$$

6) $b_1, b_2, g_1, g_2, h_1, h_2$ 中均不含 f .

2.2 算法的改进型

对于 A 型两步递归算法的等价变换的可用性条件, 即 F_1, F_2 分别具有右单位元 e_1, e_2 , 为的是保证在一开始 $b_1(x), b_2(x)$ 为真时变换的正确性. 因此, 只要将输出模式多加一分支, 分离另作处理, 则条件 4) 实际上可略去. 为此提出 A 型两步递归算法的等价变换改进型模式:

输入模式:

$$f(x) \equiv \text{if } b_1(x) \text{ then } h_1(x) \text{ else}$$

$$\text{if } b_2(x) \text{ then } h_2(x) \text{ else } F_1(F_2(f(k(x)), g_1(x)), F_2(f(k^2(x)), g_2(x)))$$

输出模式:

$$f(x) \equiv \text{if } b_1(x) \text{ then } h_1(x) \text{ else}$$

$$\text{if } b_2(x) \text{ then } h_2(x) \text{ else } G(k(x), g_1(x), g_2(x))$$

$$G(x, y, z) \equiv \text{if } b_2(x) \text{ then } F_1(F_2(h_2(x), y), F_2(h_1(k(x)), z)) \text{ else } G(k(x), F_1(F_2(g_1(x), y), z), F_2(g_2(x), y))$$

可用性条件为:

$$1) F_2 \text{ 满足结合律: } F_2(F_2(x, y), z) = F_2(x, F_2(y, z)).$$

$$2) F_1 \text{ 满足 } F_1(F_1(x, y), z) = F_1(F_1(x, z), y).$$

3) F_2 关于 F_1 具有左、右分配律:

$$F_2(F_1(x, y), z) = F_1(F_2(x, z), F_2(y, z));$$

$$F_2(x, F_1(y, z)) = F_1(F_2(x, y), F_2(x, z)).$$

4) $b_2(x) \equiv b_1(k(x)).$

5) $b_1, b_2, g_1, g_2, h_1, h_2$ 中均不含 f .

2.3 算法改进型的正确性

引理 2 $G(x, y, z) = F_1(F_2(f(x), y), F_2(f(k(x)), z))$

证, 略(证明方法同引理 1).

定理 2 在改进的 A 型两步递归算法的等价变换中,输出模式与输入模式等价。

证:略(证明方法同定理 1)。

4. 应用示例

Fibonacci 函数:

$f(x) \equiv \text{if } x=0 \text{ then } 0 \text{ else}$

$\text{if } x=1 \text{ then } 1 \text{ else } G(x, 1, 0)$

$G(x, y, z) \equiv \text{if } x=1 \text{ then } y \text{ else } f(x-1) + f(x-2)$

这时 $b_1(x) \equiv x=0, k_1(x) \equiv x-1, b_2(x) \equiv x=1, g_1(x) \equiv 1, h_1(x) \equiv 0, g_2(x) \equiv 1, h_2(x) \equiv 1, e_1 \equiv 0, F_1(x, y) \equiv x+y, e_2 \equiv 1, F_2(x, y) \equiv x * y$ 。容易验证它满足可用性条件(1—6)。于是有:

$f(x) \equiv \text{if } x=0 \text{ then } 0 \text{ else}$

$\text{if } x=1 \text{ then } 1 \text{ else } G(x-1, 1, 1)$

$G(x, y, z) \equiv \text{if } x=1 \text{ then } y \text{ else } G(x-1, y+z, y)$

实例

$f(6) = G(5, 1, 1) = G(4, 2, 1) = G(3, 3, 2) = G(2, 5, 3) = G(1, 8, 5) = 8$

三、对二阶 C 型 Cooper 变换的改进

3.1 二阶 C 型 Cooper 变换

文[2]中给出了 C 型 Cooper 变换如下:

输入模式:

$f(x) \equiv \text{if } b(x) \text{ then } h(x) \text{ else}$

$\text{if } b_1(x) \text{ then } F_1(f(k_1(x)), g_1(x)) \text{ else}$

$F_2(f(k_2(x)), g_2(x))$

输出模式:

$f(x) \equiv G(x, e)$

$G(x, y) \equiv \text{if } b(x) \text{ then } F_1(y, h(x)) \text{ else}$

$\text{if } b_1(x) \text{ then } G(k_1(x), F_1(y, g_1(x))) \text{ else}$

$G(k_2(x), F_2(y, g_2(x)))$

可用性条件:

1) F_1 满足: $F_1(F_1(x, y), z) = F_1(x, F_1(z, y))$ 。

2) F_1, F_2 满足: $F_1(F_2(x, y), z) = F_1(x, F_2(z, y))$

3) F_1 具有右单位元 e 。

4) $b, b_1, h, k_1, k_2, g_1, g_2$ 中均不含 f 。

3.2 二阶 C 型 Cooper 变换改进型

二阶 C 型 Cooper 变换可用性条件 3) 即 F_1 具有右单位元 e , 为的是保证在一开始 $b_1(x)$ 为真时变换的正确性。因此只要将该模式多加一个分支, 另作处理, 则可用性条件 3) 实际上可略去, 于是我们提出二阶 C 型 Cooper 变换的改进型。

输入模式:

$f(x) \equiv \text{if } b(x) \text{ then } h(x) \text{ else}$

$\text{if } b_1(x) \text{ then } F_1(f(k_1(x)), g_1(x)) \text{ else}$

$F_2(f(k_2(x)), g_2(x))$

输出模式:

$f(x) \equiv \text{if } b(x) \text{ then } h(x) \text{ else}$

$\text{if } b_1(x) \text{ then } G(k_1(x), g_1(x)) \text{ else}$

$G(k_2(x), g_2(x))$

$G(x, y) \equiv \text{if } b(x) \text{ then } F_1(h(x), y) \text{ else}$

$\text{if } b_1(x) \text{ then } G(k_1(x), F_1(y, g_1(x)))$

$\text{else } G(k_2(x), F_2(y, g_2(x)))$

可用性条件:

1) F_1 满足: $F_1(F_1(x, y), z) = F_1(x, F_1(z, y))$ 。

2) F_1, F_2 满足: $F_1(F_2(x, y), z) = F_1(x, F_2(z, y))$ 。

3) $b, b_1, h, k_1, k_2, g_1, g_2$ 中均不含 f 。

3.3 改进型的正确性

引理 3 $G(x, y) = F_1(f(x), y)$

证:略(证明方法同引理 1)。

定理 3 在改进的二阶 C 型 Cooper 变换中,输出模式与输入模式等价。

证:略(证明方法同定理 1)。

3.4 应用示例

函数 $E_1(x_1, x_2) = x_1 * x_2$

(x_1 : 自然数, x_2 : 非负整数)

$E_1(x_1, x_2) \equiv \text{if } x_2=0 \text{ then } 1 \text{ else}$

$\text{if } x_2 > 0 \text{ then } E_1(x_1, x_2-1) * x_1$

$\text{else } E_1(x_1, x_2-1)/x_1$

$\Rightarrow E_1(x_1, x_2) \equiv \text{if } x_2=0 \text{ then } 1 \text{ else}$

$\text{if } x_2 > 0 \text{ then } G(x_1, x_2-1, x_1)$

$\text{else } G(x_1, x_2+1, 1/x_1)$

$G(x_1, x_2, y) \equiv \text{if } x_2=0 \text{ then } y \text{ else}$

$\text{if } x_2 > 0 \text{ then } G(x_1, x_2-1, y * x_1)$

$\text{else } G(x_1, x_2+1, y/x_1)$

$E_1(2, 4) = G(2, 3, 2) = G(2, 2, 4) = G(2, 1, 8) =$

$G(2, 0, 16) = 16$

参考文献

- [1] 仲萃豪等, 程序设计方法学, 北京科学技术出版社, 1985
- [2] 蔡经球、张克均, 扩展 Cooper 变换, 厦门大学学报(自然科学版) Vol. 31, No. 5, 1992
- [3] 蔡经球, 关于横向程序变换的若干研究, 软件学报, 1990, 4
- [4] 潘晓玲、罗旭东, 扩展一阶 B 型 Cooper 变换, 北京轻工业学院院报, Vol. 13, No. 1, 1995
- [5] 张为群等, 几种反演递归变换模式, 计算机研究与发展, Vol. 32, 增刊, 1995
- [6] Cooper D. C., The Equivalence of Certain Computations, Comp. J., 1966. 9
- [7] Burstall, R. M. et al., A Transformation System for Developing Recursive Programs, JACM, 1977. 1
- [8] 蔡经球、周 松, 递归算法的若干等价变换, 厦门大学学报, 1983. 2