613

可換則に基づいたTRSの変換について

河口 信夫　　坂部 俊樹　　稲垣 康善 (名古屋大学)

1 はじめに

プログラム変換 [1] の重要性が認められて久しく，変換の正当性の研究 [2] などが進められている．しかし，一般のプログラム変換では変換と効率の関係が複雑であるため，実用的な変換方針は提案されていない．我々は変換手法を可換則に基づく変換に限定し，関数型の計算モデルTRSを用いて加算，乗算などの規則に対しプログラム変換実験を行なった．その結果，規則によっては10倍程度の効率の差が生じ，十分に変換の効果があることが確かめられた．本稿では可換則に基づく変換の手法，効率的なプログラムを得るための変換方針について述べる．

2 項書き換え系 (Term Rewriting Systems:TRS)

簡単にTRSについて述べる．TRSは左辺，右辺と呼ばれる2項の対の集合からなり，この対を規則と呼ぶ．項は加算無限個の変数記号 V，関数記号 F から帰納的に定義される集合 $T(F,V)$ の要素である．項の書き換えとは，規則の左辺の変数に適当な代入を施した項が，部分項に一致するとき，その部分を右辺に代入を施した項に書き換えることをいう．TRSはこの書き換えを，それ以上書き換えできない項 (正規形) まで繰り返す．

3 可換則に基づく変換

変換例として階乗の計算をとりあげる．最初に整数の加算 (add) の規則を示す．ただし，整数は0と1加算関数 s によって表されている．

$$\text{A1:}\quad add(0,y)\rightarrow y \quad , \quad add(s(x),y)\rightarrow s(add(x,y))$$

add の2引数には可換則が成り立つので，規則を可換則により変換すると，4種類になる．これらの規則をそれぞれA[1-4]と呼ぶ．A2,A4はそれぞれA1の右辺，左辺を交換したもの．A3はA2の左辺を交換したものである．これらの効率を考えると $add(n,m)$ に対し，A1は第1引数により再帰を行なうため $n+1$ 回の書き換え，同様にA3は第2引数により再帰を行なうので $m+1$ 回の書き換えを必要とする．一方,A2,A4は書き換えを行なう度に第1引数と第2引数を入れ換えるため，それぞれ $min(n+0.5,m+1)*2, min(n+1,m+0.5)*2$ 回の書き換えを必要とする．単一の書き換えに対し効率を考えれば,A1,A3が良く,A2,A4が悪くなるが，引数の差が大きい場合の平均で考えるとA2,A4も悪くはなく，add 単独ではどの規則が効率的かは判断することはできない．次に，乗算 ($mult$)，階乗 ($fact$) の規則を示す．

$$\text{M1:}\quad mult(0,y)\rightarrow 0 \quad , \quad mult(s(x),y)\rightarrow add(y,mult(x,y))$$
$$\text{F1:}\quad fact(0)\rightarrow s(0) \quad , \quad fact(s(x))\rightarrow mult(s(x),fact(x))$$

$mult$ では可換な部分が3箇所あるため可換則による変換により8種類の規則が得られる．これをM[1-8]と呼ぶ．$fact$ では右辺に $mult$ があるため,2種類の規則F1,F2が得られる．結局，階乗の規則は組み合せにより64種類となる．これらをすべて最外最左，最内最左戦略を用いて計算実験を行なった．結果の一部を以下に示す．

表1：規則の組み合せによる階乗の書き換え回数

戦略	組み合せ	fact(2)	fact(3)	fact(4)	fact(5)	fact(6)
最内最左	A1:M1:F1	14	26	62	194	928
	A1:M1:F2	12	24	62	232	1194
	A3:M1:F2	9	18	92	1532	44604
最外最左	A1:M1:F2	12	24	62	232	1194
	A1:M1:F1	20	74	330	1782	11426
	A2:M6:F2	34	138	594	3126	19838

4 変換の方針

実験の結果，$fact(5)$ では書き換え回数の最小値と最大値には7倍以上の差があり，十分変換の効果が確かめられた．また，規則によっては最外戦略でも十分な効率が得られることがわかる．この結果から，我々は「引数の大きさを比べ，一般的に小さくなる引数で再帰を行なうように変換する」という変換方針を提案する．ただし，項の大小関係は変換前に与えられているものとする．この方針に従いF1からの変換例を示す．F1の右辺には $mult(s(x),fact(x))$ があるが，$s(x)$ より $fact(x)$ が大きいと分かっているとすると，方針より $mult$ は第1引数で再帰を行なうように変換される．その結果のM1では，右辺に $add(y,mult(x,y))$ があり，第1引数の y が小さいと分かっているとすると，add は第1引数で再帰を行なうよう変換され，それはA1である．変換結果として規則はA1:M1:F1となり，書き換え回数の最も少ない規則となり，方針の正しさが確かめられた．

5 まとめ

可換則に基づく変換では，引数の大小関係より，再帰を行なう規則を選択することで，最内書き換えにおいて最も効率のよい規則を得ることができる．今後の課題として，項の大小関係の指定法や，戦略を考慮した変換法についての検討が挙げられる．

参考文献

[1] Burstall,R.M., Darlington,J. :A Transformation System for Developing Recursive Programs :J.ACM, Vol.24, No.1, pp44-67(1977).

[2] Toyama,Y :How to prove equivalence of term rewriting systems without induction: Theoritical Computer Science, No 90, pp369-390(1991).