可換則に基づく項書換え系の自動変換システムとその評価

1G－1

河口信夫　坂部俊樹　稲垣康善

名古屋大学工学部

1　はじめに

可換則に基づく項書換え系の変換[4]は，可換な演算の引数を項の上の単純化順序を用いた評価基準により入れ換え，再帰の回数を減らすことで効率化を行なうもので，変換そのものが単純であるため自動化に有利である．しかし，実際に項書換え系の自動変換を行なうためには，以下のような解決すべき点がある．

1. 変換の正当性
2. 可換則が成り立つ関数の同定
3. 項の上の単純化順序

本稿ではまず，可換則に基づく変換の正当性を示し，2. については被覆集合帰納法[3]を用いることにより可換則な関数を同定する手法，3. については再帰経路順序[2]を用いた停止性判定システムを利用して順序を定める手法を提案する．また，我々はすでにGUIを持つ項書換え系の変換支援環境：TERSE[5]を実現しているが，その上にこれらの手法による可換則に基づく項書換え系の自動変換システムを実装し，その評価を行なった．

2　項書換え系

項書換え系について簡単に説明する．詳しくは[1]等を参照されたい．

可算無限個の変数記号の集合V，アリティ，ソートを持つ関数記号の集合Fの上に定義される項集合を$T(F,V)$と書く．関数記号は構成子関数記号の集合Cと被定義関数記号の集合Dに区別される．また，関数記号fに対し$arity(f)$, $sort(f)$はそれぞれアリティ，ソートを表す．特に$arity(f)=\varepsilon$のときfを定数記号という．TRS (Term Rewriting Systems)は左辺，右辺と呼ばれる2項の対$\alpha_i \to \beta_i$の集合からなり，この対を書換え規則と呼ぶ．

Automating Commutative Law Based Transformation of TRS

Nobuo KAWAGUCHI,Toshiki SAKABE,
Yasuyoshi INAGAKI (Nagoya University)

3　可換則変換

TRS Rと2引数関数記号fに対し，

$$\forall t,u \in T(F), f(t,u) \stackrel{R}{=} f(u,t)$$

であるときfはRの下で可換であるという．ただし$\stackrel{R}{=}$はTRS Rによる書換えの関係から定まる等価関係である．あるTRS Rの下で可換な関数の規則における引数を入れ換えることを引数交換という．例えば加算を計算するTRS A1は可換なaddに対する引数交換の適用により，TRS A2に変換される．

$$\text{A1}: \begin{cases} add(0,y) & \to & y \\ add(s(x),y) & \to & s(add(x,y)) \end{cases}$$

$$\text{A2}: \begin{cases} add(0,y) & \to & y \\ add(s(x),y) & \to & s(add(y,x)) \end{cases}$$

引数交換の繰返しによるTRSの変換を可換則変換という．可換則変換を行なうことによりTRSの効率（ここでは書換え回数をいう）が向上する場合があることを我々はすでに[4]で示しており，効率化のための可換則変換アルゴリズムを提案している．しかし自動変換を行なうためには，このアルゴリズムの入力であるTRS上の可換な関数記号，項の上の単純化順序を定めることが必要である．以下では変換の正当性と共にそれらの手法を示す．

ここでTRS RからR'への変換が正当であるとは以下の条件が成り立つことである．

$$\forall x,y \in T(F), x \stackrel{R}{=} y \iff x \stackrel{R'}{=} y$$

【定理 3.1】引数交換の正当性
関数記号fはTRS Rの下で可換であり，Rに対してfについての引数交換を行ないTRS R'が得られているとする．このときfがR'の下でも可換ならばこの引数交換は正当である．□

【証明】左辺に対して引数交換を適用した場合を考える．$t \stackrel{R}{\to} t''$とすると，t'が存在し$t' \stackrel{R'}{\to} t''$かつ$t' \stackrel{R'}{=} t''$となる．従って$t \stackrel{R'}{=} t''$である．このことから$t \stackrel{R}{=} u \iff t \stackrel{R'}{=} u$である．すなわち，引数交換は正当であることが導かれる．右辺の引数交換についても同様に証明される．□

〖系 3.1〗 可換則変換の正当性
正当な引数交換の繰返しによる可換則変換は正当である．□

3.1 可換な関数の同定

可換則は帰納的定理であるため，その証明には被覆集合帰納法[3]を用いる．一般に被覆集合は各ソートで無限個存在し，自動的に生成することは難しい．そこで，ソートごとにいくつかの被覆集合を前もって与えることとし，関数記号ごとに被覆集合帰納法を用いて可換則が成立することを確かめる．例えば関数記号 f が可換であるかどうかチェックする場合，まず f のアリティを調べる．$arity(f) = s \times s$ である時，ソート s に対する被覆集合 M により，新たな被覆集合 $M' = \{M \times M\}$ を作成し，これにより $f(x,y) = f(y,x)$ に対する被覆集合帰納法を行なう．すなわち M' の各々の要素 $< m_i, m_j >$ について，$R \cup \{f(p,q) \rightarrow f(q,p) | p \in Sup(m_i), q \in Sup(m_j))\}$ を用いて $f(m_i,m_j) = f(m_j,m_i)$ が成立するかどうかをメタ変数を用いた書換えにより確かめる．仮定としての等式を書換え規則にするときに，書換えの方向を R と重ならないように定めることで，書換えが停止しなくなることを避けることができる．$Sup(m)$ とは直観的には m 中のメタ変数を用いてそのメタ変数へのパス上でメタ変数と同じソートの関数記号とを入れ換えたものの集合を指す．

3.2 項の上の単純化順序

単純化順序としては停止性判定のための順序として良く用いられる再帰経路順序 $\succ_{rpo}$ を採用した．これは関数記号の間に半順序を必要とするので，次のような停止性を判定するための手続きを用いて順序づけを行ない，項の間の大小関係として用いる．

順序付けの手続き：入力として TRS R，空集合で初期化された関数記号上の半順序 O_f を用いる．t に現れる関数記号から１つを選んだものを $Fun(t)$ とする．

1. 規則 $\alpha \rightarrow \beta \in R$ に対して $(Fun(\alpha) > Fun(\beta))$ により O_f を拡張する．
2. 規則 $\alpha \rightarrow \beta$ が O_f により $\alpha \succ_{rpo} \beta$ ならば $R = R - \{\alpha \rightarrow \beta\}$．
3. $R = \emptyset$ ならば O_f で成功．
4. これ以上 O_f の拡張ができないならば失敗，できるならば 1 へ．

順序づけの制御は関数 Fun により行なわれ，出現による重みづけなどを用いて探索の効率化を計る必要がある．また，O_f の拡張を失敗した際にバックトラックを行なう必要もある．

4 自動変換システム

以上の手法を ML を用いて計算機上に実現した．[4] の例題で用いている階乗を計算する TRS は引数交換により 64 通りの TRS に変換できるが，これらすべてについて自動変換システムにより変換を行なった．その結果，64 通り全ての TRS が 4 通りの最適な効率をもつ TRS に変換された．また，変換の過程は TERSE のメインウィンドウ上で確認できる．

今まで人手で行なってきた変換がすべて自動的に行なうことが可能となり，その結果を目で確認することができるため変換を行なうことが非常に容易である．しかし，現在はまだ被覆集合を前もって与えるなどの処理を行なう必要があることや，結合則を用いた規則などの再帰経路順序では順序がつかない TRS が存在することもあり，まだ多くの改良の余地がある．

5 まとめ

項書換え系の可換則変換について，変換の正当性，可換な関数の同定法，項の間の単純化順序の定め方について議論を行なった．また，これらの手法を計算機上に自動変換システムとして実現しその評価を行なった．現在の可換な関数の同定法や停止性判定システムはまだ完全なものではないので，より一般の TRS について自動変換が行なえるよう，システムを拡張することが今後の課題である．

参考文献

[1] Huet, G.: “Confluent Reductions:Abstract Properties and Applications to Term Rewriting Systems”, J.ACM, Vol.27, No.4, pp797-821 (1980).

[2] Dershowitz, N.: “Termination of Rewriting” , J. Symbolic Computation, Vol.3, pp69-116 (1989).

[3] 酒井, 坂部, 稲垣:” 代数的仕様の検証のための被覆集合帰納法” , COMP90-5(1990).

[4] 河口, 坂部, 稲垣: “可換則に基づく項書換え系の変換と必須呼びによる効率の評価” , COMP93-64 (1993).

[5] 河口, 坂部, 稲垣: “グラフィカルユーザーインタフェースを持つ項書換え系の解析・変換支援環境” , SS93-44(1994).