メッセージ交換を用いた項書き換え系のリデューサの作成

1E−3

矢野　博之†　布川　博士‡　野口　正一†
†東北大学応用情報学研究センター　‡東北大学電気通信研究所

1　はじめに

項書き換え系(以下TRS)のリダクションを実現する方法の1つに，メッセージ交換を用いる方法[1]がある．メッセージ交換によるリダクションは，TRS上の各関数記号fに対してproc(f)(プロセスという)を割り当て，リダクションの実行時には，それらの間で，定まったプロトコルを用いて通信を行うことによりリダクションを行う方法である．すなわち，この方法はTRSをプロセスの集合へ変換するコンパイル方式である．一般にコンパイル方式で最外戦略を実現する際には書き換えの制御を行うランタイムルーチンを付加する必要があり，書き換え規則の増加に対して柔軟に対応できない[2]．そのため，一般のコンパイラは最内戦略をとることが多い[3][4]．メッセージ交換による方式では，最外戦略の実現に関してこのルーチンを必要としないのが特長である．また，proc(f)は，それ自身がメッセージを受け取り，その内部構造(書き換え規則によって定まる)に従った動きをするため，proc(f)が受け取るメッセージの種類によりリダクション時の戦略を変えられる．さらに，メッセージの種類を増やすことにより，簡単に種々の制御機能を追加することができる．

2　処理系の実現

2.1　コンパイラの実現

TRSが与えられたときに，各関数記号fに対するproc(f)としてSchemeの関数を生成するものを作成した．現在Sun3上のMIT Schemeで稼働している．以下に実行例を示す(poは並列最外戦略，po-oneは並列最外戦略での1回の書き換え，piは並列最内戦略を表す)．

実行例

```
1 ]=> (send  '(po CRT (times (s zero) (s (s zero)))) fact)

;Value: (nf (s (s zero)))

1 ]=> (send  '(po-one CRT (times (s zero) (s (s zero)))) fact)

;Value: (fact (add (s (s zero)) (times zero (s (s zero)))))

1 ]=> (send  '(pi CRT (times (s zero) (s (s zero)))) fact)

;Value: (nf (s (s zero)))
```

Implementation of Reducer of Term Rewriting System based on Message Passing
Hiroyuki YANO, Hiroshi NUNOKAWA, Shoichi NOGUCHI
Research Center for Applied Information Sciences, Tohoku University
Research Institute of Electrical Communication, Tohoku University

2.2　メッセージ交換を用いたステッパ

ステッパは，項の書き換えたい部分項をその出現場所(stepperの受け取るメッセージ中のoc)を指定してユーザが望む戦略で1回の書き換えを行なうものである．一般にコンパイル方式でTRSのリデューサを作成した場合，ステッパの作成はコンパイルされた各関数に，与えられた戦略で1回の書き換えを行なう機能を付加しなければならなくなる．それに対して本方式では，各関数プロセスを変更することなしに，独立してステッパの実現が可能になる(図1)．proc(stepper)もMIT schemeで実現を行なった．

```
process stepper(sel, ret, term, oc)
case oc of
  (oc1,oc2, ···, ocn) : f(g1, ···, gi, ···, gn)   (where oc1 = i)
                        send (sel, h1, gi, (oc2, ···, ocn))
                          to proc (stepper)
                        send f(g1, ···, h1, ···, gn) to ret
                ( ) : f(g1, ···, gn)
                        send (sel, ret, g1, ···, gn) to proc (f)
```

図1.　ステッパのためのプロセスの内部表現

3　まとめ

本稿では，メッセージ交換に基づく項書き換え系のリダクション方式を用いたリデューサの作成を行なった．またステッパをプロセスと考えメッセージ交換により実行する方法を示し，その実現を行なった．この方法により，ステッパも分散処理システム上での実行が可能になる．

参考文献

[1]布川，野口：プロセス間でのメッセージ交換を用いた項書き換え系のリダクション，信学技法，SS89-28(1990), pp29-38
[2]布川，黒田，富樫，野口：項書き換え系の関数型言語への変換による実現，コンピュータソフトウェア Vol. 4 No. 4(1987), pp 5-15
[3]酒井，坂部，稲垣：抽象データ型直接実現システムCdimple，コンピュータソフトウェア Vol. 4 No. 4(1987), pp 16-27
[4]戸村，二木：項書き換えシステムからLispプログラムへの変換系，信学技法，SS86-9(1986), pp15-20
[5]MIT Scheme Reference, Massachusetts Institute of Technology (1989)