5F−3

等式プログラミングにおけるリフレクション*

佐藤 崇昭　栗原 正仁　大内 東†

北海道大学工学部‡

1 はじめに

リフレクション(*reflection*)とは,「自分自身」について感知したり,変更したりする機能である.これまでに,リフレクション機能をプログラミング言語にもたせる研究は,関数プログラミングや論理プログラミングなど様々な枠組みで行われてきた.

本稿では,等式プログラミングの枠組みにおけるリフレクション機能を考察し,リフレクション機能をもつ等式プログラミング・システム(Reflective Equational Programming System,以下ではREPSと略記)を与える.

等式プログラミングとは,項書き換えシステムを計算モデルとし,等式を左辺から右辺への書き換え規則として用いる.したがって,等式プログラムは,書き換え規則の集合である.

REPSでは,計算状態を書き換え規則の集合とリデックスおよびそのまわりの文脈によって,モデル化している.これらの計算状態を表すメタ・レベルのオブジェクトはリーフィケーション(*reification*)によって,項へと変換されて計算が実行される.また,リフレクション(*reflection*)によって,任意の書き換え規則の集合や文脈を指定して,REPSの計算状態を変更することができる.

2 REPSにおけるリフレクション

2.1 REPSの項

関数シンボルの集合$\mathcal{F}$,変数シンボルの集合$\mathcal{V}$上でREPSにおける項を定義する.$\mathcal{F},\mathcal{V}$に対して,項の集合$\mathcal{T}(\mathcal{F},\mathcal{V})$を次の条件を満たす最小集合であるとする.

- $x \in \mathcal{V}$ならば,
$$x \in \mathcal{T}(\mathcal{F},\mathcal{V})$$
- $t_1, t_2, \ldots, t_n \in \mathcal{T}(\mathcal{F},\mathcal{V}), f \in \mathcal{F}$ならば,
$$f(t_1, t_2, \ldots, t_n) \in \mathcal{T}(\mathcal{F},\mathcal{V})$$
- $t_1, t_2, \ldots, t_n \in \mathcal{T}(\mathcal{F},\mathcal{V}), f \in \mathcal{F}$ならば,
$$f(t_1, t_2, \ldots, t_n).R.C \in \mathcal{T}(\mathcal{F},\mathcal{V})$$

ただし,R, Cはそれぞれ(リーファイされた書き換え規則の集合,文脈を表す)変数シンボルである.なお,第3の形の項は,書き換え規則の左辺としてのみ現れうる.

空の場所を表す特別な記号□を1つ含む特別な項を文脈という.項$\mathrm{embed}(C,t)$は,リーファイされた文脈C中の□に対応する部分を項tで置き換えた結果を表し,これを$C[t]$と略記する.

*Reflection in Equational Programming
†Taka-aki SATOH, Masahito KURIHARA and Azuma OHUCHI
‡Faculty of Engineering, Hokkaido University

2.2 Reification

リーフィケーションは,計算状態を表すメタ・レベルのオブジェクトを項へと変換するプロセスである.REPSでは計算状態を書き換え規則の集合$\mathcal{R}$と文脈Cでモデル化している.リーフィケーションは,リーファイア(*reifier*)によって実行される.REPSにおいて,ある書き換え規則の左辺が$f(t_1,\ldots,t_n).R.C$の形をしているとき,関数シンボルfはリーファイアである.

任意のREPSに対して,その書き換え規則の集合を$\mathcal{R}$とする.現在,書き換えの対象となっている項をt_0とする.ある書き換え規則
$$(f(t_1,\ldots,t_n).R.C \to t) \in \mathcal{R}$$
が存在し,t_0のリデックスとして選ばれた部分項が,$f(t_1,\ldots,t_n)$に照合するとき,この書き換え規則がt_0に適用される.このとき,リーフィケーションが起きる.すなわち,$\mathcal{R}$と文脈Cは,項へと変換されそれぞれ,変数R,変数Cへと代入され,t_0全体が対応する右辺に書き換えられる.

リーファイされた書き換え規則の集合Rに対しては,書き換え規則の追加・削除などのオペレーションが可能である.また,リーファイされた文脈Cは,項となっているので,書き換え規則による変更が可能である.

2.3 Reflection

特別な関数シンボルreduceによって構成される項$\mathrm{reduce}(r,t)$が書き換えられるときに必ず,リフレクションが起きる.このとき,reduceの第1引数で指定した書き換え規則の集合をもつREPSが新たに生成され,このREPSのもとで,項tを簡約する.項$\mathrm{reduce}(r,t)$は,最終的にこの簡約の結果得られる正規形に置き換わる.

3 REPSの操作的意味論

REPSの操作的意味論として,簡約→を定義する.まず,代入を定義する.

3.1 代入

x_iがすべて異なるような項t_iと変数x_iの対の有限集合
$$\theta = \{t_1/x_1, \cdots, t_n/x_n\}$$
を代入であるという.このとき,項tに対して代入θを作用させた結果$t\theta$を次のように定義する.

- $t \in \mathcal{V}$ならば,
$$t\theta = \begin{cases} t_i & \text{if } t \equiv x_i \\ t & \text{otherwise} \end{cases}$$
- $t \equiv f(s_1, \cdots, s_m)$ならば,
$$f(s_1, \cdots, s_m)\theta = f(s_1\theta, \cdots, s_m\theta)$$

• $t \equiv g(u_1,\cdots,u_l).R.C$ ならば,

$$(g(u_1,\cdots,u_l).R.C)\theta = g(u_1\theta,\cdots,u_l\theta).R\theta.C\theta$$

したがって，任意の代入θは，$\mathcal{T}(\mathcal{F},\mathcal{V}) \mapsto \mathcal{T}(\mathcal{F},\mathcal{V})$ なる写像である.

3.2 簡約の定義

項tとtを簡約する書き換え規則の集合$\mathcal{R}$の対$\{t\}_\mathcal{R}$を閉包項という．閉包項$\{t\}_\mathcal{R}$は，その部分項に閉包項$\{s_i\}_{\mathcal{R}_i}$を持つことができる．したがって，部分項が閉包項であれば，その部分項は$\mathcal{R}_i$で簡約されるが，部分項が項であれば，$\mathcal{R}$で簡約される．一般に項tと閉包項$\{t\}_\mathcal{R}$において，$t \neq \{t\}_\mathcal{R}$であるが，特に$t \in NF(\mathcal{R})$のとき，$t \equiv \{t\}_\mathcal{R}$である．ただし，$NF(\mathcal{R})$は，書き換え規則の集合$\mathcal{R}$における正規形の集合を表す.

ここで，閉包項の集合$\mathcal{FT}(\mathcal{T}(\mathcal{F},\mathcal{V}),\mathcal{R})$上の二項関係$\Rightarrow$を定義する．書き換え規則の集合$\mathcal{R}$において閉包項$s$を簡約するとき，$\Rightarrow$は次のような二項関係であるとする.

3.2.1 書き換え規則に対応する簡約

ある書き換え規則$(l \to r) \in \mathcal{R}$，およびパターンマッチングによって得られる代入$\theta$に対して,

• $l \equiv f(\cdots).R.C$ならば (Reification),

$$\{C[f(\cdots)\theta]\}_\mathcal{R} \Rightarrow \{r\tau\}_\mathcal{R}$$

ただし，τは

$$\tau = \theta \bullet \{rules^\wedge(\mathcal{R})/R, context^\wedge(C)/C\}$$

なる代入の合成であり，$rules^\wedge, context^\wedge$はそれぞれ，書き換え規則の集合，文脈を項へと変換する関数である.

$$rules^\wedge \ : \ \mathcal{R} \mapsto \mathcal{T}(\mathcal{F},\mathcal{V})$$
$$context^\wedge \ : \ C \mapsto \mathcal{T}(\mathcal{F},\mathcal{V})$$

• otherwise

$$\{C[l\theta]\}_\mathcal{R} \Rightarrow \{C[r\theta]\}_\mathcal{R}$$

適用された書き換え規則の左辺がリーファイアであれば (reification)，文脈が放棄されるが，それ以外の場合では，文脈が保持される.

3.2.2 関数シンボル reduce による簡約

特別な項 reduce(r,t) をリデックスにもつ項ならば (Reflection),

$$\{C[\text{reduce}(r,t)]\}_\mathcal{R} \Rightarrow \{C[\{t\}_{rules^\vee(r)}]\}_\mathcal{R}$$

特に，$s \equiv$ reduce(r,t) のとき,

$$\{\text{reduce}(r,t)\}_\mathcal{R} \Rightarrow \{t\}_{rules^\vee(r)}$$

とする.

ここで，$rules^\vee$は，項で表された書き換え規則の集合をメタ・レベルのオブジェクトである書き換え規則の集合へと変換する関数である.

$$rules^\vee \ : \ \mathcal{T}(\mathcal{F},\mathcal{V}) \mapsto \mathcal{R}$$

よって，$\mathcal{R}_1 = \{f(a) \to b\}, a \notin NF(\mathcal{R}_2)$ のとき，閉包項$\{f(\{a\}_{\mathcal{R}_2})\}_{\mathcal{R}_1} \not\Rightarrow \{b\}_{\mathcal{R}_1}$である.

このようにして定義される閉包項上の二項関係$\Rightarrow$において，項だけを抽出すると，REPS における項の集合$\mathcal{T}(\mathcal{F},\mathcal{V})$上の二項関係$\to$を得る．すなわち,

$$\{s\}_\mathcal{R} \Rightarrow \{t\}_{\mathcal{R}'} \text{ならば}, \ \ s \to t$$

である.

4 REPS の応用例

4.1 エラー処理

ある数を 0 で割るような項を書き換えようとしたとき，次のような書き換え規則を記述しておけば，すぐにエラーとして処理することができる.

```
x - 0 = x
0 - y = 0
s(x) - s(y) = x - y
div(0,s(y)) = 0
div(s(x),s(y)) = s(div(x - y,s(y)))
div(x,0).R.C = error
```

項の深い位置でエラーが発生した場合，引き算−でエラー処理が記述されていなくても，ただちにトップレベルで error に書き換えられる.

$$\texttt{s(0)} - \texttt{s(div(s(0),0))} \Rightarrow \texttt{error}$$

4.2 完備化

任意の計算結果が必ず止まり結果が得られることを停止性といい，計算結果が (求まれば) 一つに定まることを合流性という．そして，この二つの性質を持つとき，完備であるという．等式 (書き換え規則) の集合を完備にする手続きは，Knuth-Bendix 完備化手続き (以下，KB) と呼ばれる.

KB は書き換え規則をデータとして扱うので，リフレクション機能のない等式プログラミング・システムでは，完備化する書き換え規則の集合を簡約するためのインタプリタ・プログラムも記述する必要がある．このため，完備化する書き換え規則をインタプリタ・プログラムを通じてリダクション・マシンが計算するので，計算の効率が良くない．しかし，リフレクション機能を用いると，リーフィケーションによって，完備化する書き換え規則の集合を書き換え規則で扱うことができる．したがって，KB プログラムを記述している書き換え規則の集合$\mathcal{R}_{KB}$と完備化する書き換え規則の集合$\mathcal{R}_R$をもつ REPS プログラム$\mathcal{R}_{KB} \cup \mathcal{R}_R$を記述することによって，リダクション・マシンが効率良く完備化を実行することができる.

5 おわりに

本稿では，等式プログラミングにリフレクション機能を付加することを試みた．現在のところ，REPS の実現にむけて研究中である．また，REPS の応用例として，書き換え規則を動的に変える動的項書き換え計算 [1] やメンバシップ条件のようなメタ記述のある項書き換え計算 [2] に対しての応用にむけても研究を行う予定である.

参考文献

[1] 馮 速，坂部 俊樹，稲垣 康善，動的項書き換え計算モデルとその応用，電子情報通信学会研究報告，COMP 91-47, 31-40, 1991.

[2] 山田 順之介，停止性をもつメンバシップ条件付き TRS の合流性について，電子情報通信学会論文誌，D-I, Vol. J74-D-I No.9 666-674, 1991