ＦＰプログラム代数則検証に関する基礎

布川 博士 ・ 富樫 敦 ・ 野口 正一
東北大学電気通信研究所

１．序章

バッカスによって提案された関数型言語ＦＰは，関数型言語の特長としての記述の簡潔さ，読みやすさ，参照透過性をもつ以外，にプログラムに関しての証明，検証，プログラムの等価変換のためのスキーマである代数則を持っているのが大きな特長である．したがってＦＰは単なる言語にとどまらず，プログラムの代数を持つ関数型プログラミングシステムである．

しかしながら，この代数を用いてプログラム方程式をヒューリスティックに解いたり有用な代数則を証明した例は見受けられるが[williams 81][Tang 84][Song 84],自動的に検証しようとする試みはほとんどない．そこで我々はＦＰの代数則を検証するシステム作成を考えているが,まずここではそのための基礎をあたえる．本稿のアプローチはＦＰを項書き換え系として記述すること，つぎにそれを用いて代数則か否かを判定するアルゴリズムを与えるというものである．

まずはじめに２章において基礎的な概念を与えた後に，３章において項書き換え系としてＦＰを記述しその性質について議論する．４章において検証系のアルゴリズムを与えその正しさを示す．

２．基礎的考察

以下ではＦＰを記述するための準備として多ソート代数，項書き換え系，代数的仕様記述についてまた代数則の議論のために終モデルとの関連について述べる．

２．１ 多ソート代数系

ソート(sort)の集合をＳ，Ｓ上の関数記号族を
$\Sigma=\{\Sigma_{\omega,s}\}_{s\in S^*}$, $s\in S$ とする．シグネチャとは２項組$<S,\Sigma>$のことである．またΣ代数とは$s\in S$に対する空でない集合A_sからなる集合族$S_A=\{A_s\}_{s\in S}$,と
$\alpha\in\Sigma$, $arity(\alpha)=s_1\ s_2\ \ldots\ s_n$, $sort(\alpha)=s$ (このとき $\alpha:s_1, s_2, \ldots, s_n\to s$ と書く)に対して関数
$f_A:A_{s1}\times A_{s2}\times\ldots\times A_{sn}\to A_s$ からなる集合族
$\Sigma_A=\{f_A\}f\in\Sigma$との２項組$<S_A,\Sigma_A>$である．さらに各ソートに対しては変数$X_s$が付随し，$X=\{X_s\}s\in S$.

２．２ 項書き換え系

[Toyama 82a]にしたがって項書き換え系について，項上の関係として定義する．ソートｓの項の集合$T[\Sigma,X]_s$及びsortは以下のように,帰納的に定義される．

１．$x_s\in X_s$はソートｓの項であり，$sort(x_s)=s$.
２．$\alpha\in\Sigma_{\lambda,s}$はソートｓの項であり，$sort(\alpha)=s$.
３．$\alpha\in\Sigma$, $arity(\alpha)=s_1\ s_2\ \ldots\ s_n$, $sort(\alpha)=s$.
$t_i\in T[\Sigma,X]_{s_i}$であるとき $\alpha\ t_1\ t_2,\ldots,t_n$ はソートｓの項である．

以上を用いて項全体の集合は
$T[\Sigma,X]=\{T[\Sigma,X]_s\}_{s\in S}$と定義される．またコンテキストＣ[ｘ]とは変数を含む項のことである．

【定義 2.2.1】（書き換え規則）
$T[\Sigma,X]$上の書き換え規則とは$T[\Sigma,X]$上の関係$\triangleright$であり，

１．どんな書き換え規則 $r\triangleright s$ においてもｓ中の変数は必ずｒ中にあらわれる．
２．$\triangleright$は有限集合である．

を満たすものをいう．□

【定義 2.2.2】（項書き換え系 ＴＲＳ）
シグネチャ$<S,\Sigma>$上の項書き換え系（ＴＲＳ）とは３項組$<T[\Sigma,X],\to,\triangleright>$であり，関係$t\to t'$が

$t\to t'$ ⇐⇒ あるコンテキストＣ，代入θ，
書き換え規則$l\triangleright r$が存在し
$t=C[l\theta], t'=C[r\theta]$である．

で定まるものである．

【定義 2.2.3】（正規項，合流性，ネーター性）
$t\to^* t\downarrow$でありｔ↓→*ｔ'となるｔ'∈$T[\Sigma,X]$が存在しないときｔ↓はｔの正規項であるといわれる．また項上の関係→がどのようなｔ$_1$に対してもｔ$_1\to^*$ｔ$_2$，ｔ$_1\to^*$ｔ$_3$であれば，あるｔ$_4$が存在してｔ$_3\to^*$ｔ$_4$かつｔ$_2\to^*$ｔ$_4$となるとき→は合流的であるといわれ，$t_0\to t_1\to\ldots$となる無限列が存在しないとき→はネーター的であると言われる． □

２．３　代数的仕様

【定義　2.3.1】（仕様，部分仕様）

仕様とは３項組Ｓｐ＝＜Ｓ，Σ，Ｅ＞である．ここで＜Ｓ，Σ＞はシグネチャ，Ｅは等式の集合である．また

$S_0 \subset S$，$\Sigma_0 \subset \Sigma$，$E_0 \subset E$であるとき

$Sp_0 = <S_0, \Sigma_0, E_0>$をＳｐの部分仕様という．□

書き換え規則l▷rを等式l=rとみなすことよりＴＲＳから仕様が得られる．

次にＴＲＳから得られた仕様Ｓｐによって定まる同値関係$\equiv_E$，$\simeq_E$，$\sim_E$を定義する．

【定義　2.3.2】（$\equiv_E$，$\simeq_E$，$\sim_E$）

１．$t \equiv_E t' \Leftrightarrow \quad t \leftarrow^* \rightarrow t'$.

２．$t \simeq_E t' \Leftrightarrow$ すべての代入θに対して

$t\theta \leftarrow^* \rightarrow t'\theta$.

３．$t \sim_E t' \Leftrightarrow$ すべての代入θ，

すべてのコンテキストＣに対して，

$C[t\theta] \leftarrow^* \rightarrow C[t'\theta]$.

ここで，$\leftarrow^* \rightarrow$はＴＲＳにおけるリダクション関係→の反射，推移，対称的，閉包である．□

このとき明らかにつぎの関係が成立する．

【命題 2.3.4】

$$\equiv_E \subset \simeq_E \subset \sim_E \quad \square$$

仕様の完全性，無矛盾性を以下のように定義する．

【定義 2.3.5】（無矛盾，強完全）

仕様Ｓｐ，その部分仕様Sp_0において，Ｓｐが無矛盾であるとはどのような$T[\Sigma_0]$の要素ｔ，ｔ'においても

$t \equiv_{E_0} t' \Rightarrow t \equiv_E t'$

をみたすことをいい，強完全であるとはソートが$s_0 \in S_0$である項には必ず　$t \equiv_E t'$　となる　$t' \in T[\Sigma_0]$が存在することである．　□

２．４　終モデル

【定義 2.4.1】（モデル）

仕様の解釈とはΣ代数であり代数の要素が項として表現できるものである．モデルとは仕様の等式をすべてみたす解釈である．　□

【定理 2.4.2】[Wand 79]

仕様Ｓｐ＝＜Ｓ，Σ，Ｅ＞が無矛盾，強完全であればＳｐには終モデルＦｍ（Ｅ）が存在する．　□

以上より我々はつぎの定理を示す．

【定理 2.4.3】

Ｓｐ＝＜Ｓ，Σ，Ｅ＞を無矛盾かつ強完全とするとき

$t \sim_E t' \Leftrightarrow$ Ｆｍ（Ｅ）でｔ＝ｔ'が成立　□

証明は紙面の都合上文献［Togashi 85］の定理2.3を参照されたい．

３．項書き換え系としてのＦＰ

３－１．記述

項書き換え系としてＦＰを記述する．ＦＰは項書き換え系を用いて記述されるが，ＦＰの構造は[Backus81 a]によれば図３－１に示すようになっておりここでもそれにしたがって階層的に記述する．

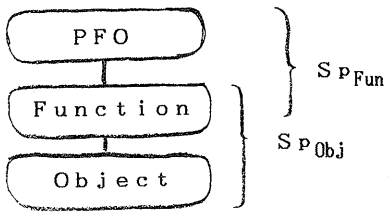

図３－１　ＦＰの階層的構造

この階層化により$Sp = Sp_{Obj} + Sp_{Fun}$，$Sp_0 = Sp_{Obj}$とし部分仕様Sp_0の存在が保証される．

またＦＰの持っているデータタイプのうちAtom,Boolに関してはすでに与えられているものとし，ここでは図3-2にＦＰの原子関数のうちでも特にシーケンスに関するものをSp_{Obj}として示す．またPFOに関して図3-3に示す．

３－２．記述されたＦＰの性質

前節で記述された項書き換え系としてのＦＰの性質について考察する．（以下Ｆｐとは項書き換え系として記述されたＦＰのことである）

【命題 3.2.1】

Ｆｐにおいて，ソートがObjであるすべての基礎項（変数を含まない項）はシーケンス構成子のみからなる項へ書き換えることができる．□

（略証）

ソートがObjである項はinj(a)，$\perp_{obj}$，nilを基に，◉，原子関数，PFOより帰納的に構成されることに着目し，その構造に関する帰納法によって示すことができる．■

```
SORT: Obj.
    CONSTRUCTORS:
        ⊥Obj : λ→Obj.
         nil : λ→Obj.
         inj : Atom→Obj.
           ● : Obj,Obj→Obj.
    EXTENDERS:
        proj : Obj→Atom.
          hd : Obj→Obj.
          tl : Obj→Obj.
       apndl : Obj→Obj.
       apndr : Obj→Obj.
       distl : Obj→Obj.
      distlr : Obj→Obj.
    RULES:
        proj(⊥obj) ▷ ⊥Atom.
          proj(nil) ▷ ⊥Atom.
         proj(x●y) ▷ ⊥Atom.
       proj(inj(x)) ▷ x.
          hd(⊥obj) ▷ ⊥Obj.
         hd(inj(x)) ▷ ⊥Obj.
           hd(nil) ▷ ⊥Obj.
          hd(x●y) ▷ x.
          tl(⊥obj) ▷ ⊥Obj.
         tl(inj(x)) ▷ ⊥Obj.
           tl(nil) ▷ ⊥Obj.
          tl(x●y) ▷ x.
  apndl(x●(nil●nil)) ▷ x●nil.
apndl(x●((y●z)●nil)) ▷ x●(y●apndl(z●nil)).
apndr(nil●(x●y)●(z●nil)) ▷ x●nil.
    apndr((x●y)●(z●nil)) ▷ x●(apndr(y●(z●nil))).
 distl(x●(nil●nil)) ▷ nil.
distl(x●((y●z)●nil) ▷ x●(y●(nil●(distl(x●(z●nil))))).
 distr(nil●(x●nil)) ▷ nil.
distr((x●y)●z●nil) ▷ x●((z●nil)●distr(y●z)).
END Obj.
```

図３－２　Sp_{Obj}の記述

```
SORT:Fun.
    CONSTRUCTORS:
        ⊥Fun : λ→Fun.
        Null : λ→Fun.
           ⊛ : Fun,Fun→Fun.
           ○ : Fun,Fun→Fun.
        ⇒ ; : Fun,Fun,Fun→Fun.
           α : Fun→Fun.
           ／ : Fun→Fun.
           ― : Obj→Fun.
    EXTENDERS:
        ： : Fun,Obj→Fun.
    RULES:
        ⊥fun：x ▷ ⊥obj.
        Null：x ▷ ⊥obj.
        (f○g)：x ▷ f：(g：x).
         αf：nil ▷ nil.
       αf：(x●y) ▷ (f：x)●(αf：y).
       (f⇒g;h)：x ▷cond(f：x,g：x,h：x).
         ／f ：nil ▷ nil.
      ／f ： (x●y) ▷ f：x●(／f●y).
        f⊛ g ：x ▷ f：x ⊛ (g ：y).
            ― ：x ▷ x.
END Fun.
```

図３－３　Sp_{Fun}の記述

【命題 3.2.2】

Ｆｐはネーター的かつ合流性をみたす．　□

（略証）

合流性に関しては書き換え規則に重なりがないことより明らか．また停止性については命題3.2.1と同様に示せる．

■

【命題 3.2.3】

Ｆｐはつぎの２つの性質をみたす

性質１

どんな項ｔ，ただし，$t \in T[\Sigma_0]$に対しても，あるｔ'$\in T[\Sigma_0{}^c]$が存在し$t \equiv_{E0} t'$となる．

性質２

どのようなｔ，$t' \in T[\Sigma_0{}^c]$に対しても$t \equiv_{E0} t'$ならばｔ＝ｔ'である．□

この性質はHuet,Hullotによって定められたprinceples of definitionと同様である．[Huet 82]

【命題 3.2.4】

Ｆｐは強完全かつ無矛盾である．したがって終モデルＦｍが存在する．□

３－３．ＦＰの代数則

つぎにＦＰにおける代数則とＦｐにおけるその定式化について述べる．

ＦＰにおいて，２つの関数形式をそれぞれＥ，Ｆとした時

にE＝Fが代数則であるとはE，Fに現れる関数変数にどのような関数を代入してもその計算結果が同じオブジェクトになることである．（すなわち全てのオブジェクトx，関数変数に具体的な関数を代入するすべてのσに対してσE：x＝σF：xとなること）Fpにおいては，以下のように定義する．

【定義 3.3.1】
sort(f)=sort(g)=Funである2つの項f，gにおいてf＝gが代数則であるとは，f～$_E$gであること□

この定義は以下で示されるようにFPの代数則と一致する．

【命題 3.3.2】
Cをsort(C［u］)=Funであり高々1つソートがFunである変数uを含んでいる項とする．このときf，gが変数を含んでいなければ，全てのオブジェクトxに対して
f：x≡$_E$g：xであれば
（C［f］）：x≡$_E$（C［g］）：x．　□

【定理 3.3.3】
sort(f)=sort(g)=Fun，f＝gにおいて次の3つは同値．
1．f＝gがFpの代数則（f～$_E$g）である．
2．すべての代入θについて，fθ：t≡$_E$gθ：t
3．fθ：t↓≡$_E$gθ：t↓　□

4．判定アルゴリズム．

4－1　アルゴリズムに対する基礎
2章において3つの同値関係≡$_E$，≃$_E$，～$_E$を述べ，3章でとくにFPの代数則のためには，第三の同値関係が重要であることを述べた．ここでしめすアルゴリズムは～$_E$に対してのものである．オリジナルのKnuth-Bendixアルゴリズムは，≡$_E$に対してのものであり，さらに，第二の同値関係にたいしては，Huet,Hullot[Huet 82]によってしめされている．本章で示されるアルゴリズムはこのHuetらのアプローチにそったものであり，FpがRとして与えられているとし，FPの代数則かどうかを判定する等式
t＝t'が入力されると定理4.2.2 によって示される結果にそった答えを得ることができる．まず本節では基本的結果として無矛盾性，強完全性に関して以下の命題を示す．

【命題　4.1.1】
E，E'を＜S，Σ＞上の等式の集合とする．E'がEを含めば，つまり，2つの間に≡$_E$⊂≡$_{E'}$の関係があれば以下の1，2が成立する．
　1．E'が無矛盾ならばEも無矛盾
　2．Eが強完全ならばE'も強完全　□

この命題と同様にして性質1，2に関して次の命題を得る

【命題　4.1.2】
E，E'を包含関係≡$_E$⊂≡$_{E'}$を満たす＜S，Σ＞上の等式の集合とするとき
　1．Eが性質1をみたす⇒E'も性質1を満たす．
　2．E'が性質2を満たす⇒Eも性質2を満たす．　□

【補題　4.1.3】
Eを強完全な等式の集合，E'はEを含む等式の集合とする．このとき，E'が無矛盾であるのは，Eが無矛盾かつすべてのE'の等式がEの終モデルFm（E）で成立するときでありかつその場合のみである．　□
（証明）　⇒）
　命題4.1.1より，E'がEの終モデルで成立することのみを示せば十分である．
　任意の等式M＝N∈E'に対してどのようなコンテキストC，代入θについても，C［Mθ］≡$_{E'}$C［Nθ］である．ここで，C［Mθ］，C［Nθ］はともにソートがs0∈S0の変数を含まない項（基礎項とよぶ）であり，Eは強完全であることからM_0，N_0∈T［Σ_0］が存在して
　　C［Mθ］≡$_E$$M_0$，　C［Nθ］≡$_E$$N_0$　．
　またE'の無矛盾性とM_0≡$_{E'}$$N_0$　であることより
M_0≡$_E$$N_0$となり，$E_0$⊂E⊂E'であることから，
M_0≡$_E$$N_0$となり
　　C［Mθ］≡$_E$C［Nθ］
を結論する．したがってFm（E）⊨M＝Nとなる．
⇐）M_0，N_0∈T［Σ_0］$_{s0∈S0}$として，M_0≡$_{E'}$$N_0$とする．≡$_{E'}$の定義から
　M_0≡$_{E'}$$N_0$
　　⇐⇒　M_0＝t_0←→t_1←→　．．．　←→t_n＝N_0
さらに書き換えの定義から
t_{i-1}＝C_i［$l_i\theta_i$］　t_i＝C_i［$r_i\theta_i$］　l_i＝r_i∈E'
　　（または，t_iとt_{i-1}を入れ換える）
と表記でき，各iについてl_i＝r_iはEの終モデルで成立

するから　$t_{i-1}=C_i[l_i\theta_i]\equiv_E C_i[r_i\theta_i]=t_i$
となる．ここで，仮定より，Eは無矛盾であるから各iについて　$t_i\equiv_{E0} t_{i-1}$となることより
$$M_0=t_0 \equiv_{E0} t_n=N_0$$
となり結局E'も無矛盾となる．　∎

【系 4.1.4】
強完全かつ無矛盾である等式の集合E，Eを含む等式の集合E'において，E'がEの終モデルで成立することとE'が無矛盾であることとは同値
$F_m(E)\models M=N, M=N\in E' \Leftrightarrow$　E'は無矛盾　□

すなわち，あらかじめ与えられている等式の集合Eが，強完全かつ無矛盾であれば，あらたに付け加えられた等式の集合Dが終モデル上で成立するかどうかの判定は，$E'=D\cup E$が無矛盾であるかどうかを調べることで行えることが示せた．とくに，$D=\{t=t'\}$とすれば，それはEの終モデルでt＝t'が成立するかを判定できることになる．

【命題　4.1.5】
M，Nのソートがともにs_0であるとき終モデル上で
M＝Nが成立するならば任意の基礎代入θについて
$M\theta\equiv_E N\theta$．またその逆も成立する．　□

これまでの議論から，アルゴリズム中で具体的に用いられているつぎの補題を示す．

【補題　4.1.6】
Eを性質1，性質2をみたす強完全かつ無矛盾な等式の集合とする．このとき，$M=c(M_1,\ldots,M_n)$
$N=c(N_1,\ldots,N_n)$　　$c\in\Sigma_0^c$
であるときMとNが終モデル上で等しいのは，各引数M_i，N_iそれぞれが終モデル上で等しいときでありかつその場合のみである．　□
（証明）
MとNが終モデル上で等しいとする．構成子をcとおくと$sort(M_i)=sort(N_i)=s_i$　$s_i\in S_0$であることより各$M_i=N_i$が終モデルで成立することを示すには命題4.1.5より任意の（変数を構成子のみからなる項で置き換える）代入θに関して$M_i\theta\equiv_E N_i\theta$をいえば十分である．そこで$\theta$をそのような代入とすると
$$M\theta=c(M_1\theta,\ldots,M_n\theta)$$
$$N\theta=c(N_1\theta,\ldots,N_n\theta)$$
であるから，各$M_i\theta$，$N_i\theta$はそのソートがS_0に属す項でありEが強完全かつ性質1をみたすことより
$$M_i\theta\equiv_E M_i' \qquad N_i\theta\equiv_E N_i'$$
となるM_i'，$N_i'\in T[\Sigma_0^c]$が存在する．ここでM＝Nが終モデル上で成立することより
$$c(M_1',\ldots,M_n')\equiv_E c(N_1',\ldots,N_n')$$
でありさらにEが性質1を満たしていることから
$$c(M_1',\ldots,M_n')=c(N_1',\ldots,N_n')$$
となる．したがって
$$M_i'=N_i' \qquad 1\leqq i\leqq n$$
ゆえに
$$M_i\theta\equiv_E N_i\theta \qquad 1\leqq i\leqq n$$
また逆に，どんなコンテキストC[x]にたいしても，
$$C[M\theta]=C[c(M_1\theta,\ldots,M_n\theta)]$$
$$\equiv_E C[c(N_1\theta,\ldots,N_n\theta)]$$
$$=C[N\theta]$$
よって，MとNは終モデル上で等しい．　∎

この補題によれば，$M_i=N_i$が終モデルええ成立することから，M，Nがこの条件をみたすとき，Eが無矛盾であるかどうかは，各引数すべてが終モデル上で成り立っているかを調べることに，帰着される．
これらの事実は判定アルゴリズムの，（a），（i）で用いられている．さらに，つぎの系が判定アルゴリズムに対する基礎となる．

【系 4.1.7】
Eを補題4.1.6と同様な等式の集合としE'をEを含む等式の集合とする．このとき2つの項
$M=c(M_1,\ldots,M_n)$，
$N=c(N_1,\ldots,N_n)$　　$c\in\Sigma_0^c$
についてM＝N∈E'とする．E'よりE''を
$$E''=E-\{M=N\}\cup\{Mi=Ni \mid 1\leqq i\leqq n\}$$
と定義したときE'が無矛盾であるための必要十分条件はE''が無矛盾であること，さらにE'が無矛盾であれば
1．$\equiv_E\subset\equiv_{E'}$
2．E'：強完全無矛盾かつ性質1，性質2を満足
3．EとE'の終モデルが一致する．
が成立する．　□

ここまではM，Nがともにおなじ構成子を根にもつときの性質について議論してきた．以下ではさらに，異なった構成子から始まるときを考える．これは直感的には，矛盾が生じるときである．まず，単純な場合としてつぎの補題をしめす．

【補題 4.1.8】
Eを性質1．2を満足する強完全な等式の集合とする．このときEが構成子c，c'により
$M=c(M_1,\ldots,M_n)$
$N=c'(N_1,\ldots,N_n)\quad c\neq c'\qquad c,c'\in\Sigma_0^c$
と表現されるM，Nに関して$M\equiv_E N$ならばEは無矛盾ではない．（この意味で矛盾である）　□
（証明）
無矛盾であるとして，背理法によって，性質2に矛盾することを示す．
$c(M_1,\ldots,M_n)\equiv_E c'(N_1,\ldots,N_n)$
とすると，任意の代入θに関して
$c(M_1\theta,\ldots,M_n\theta)\equiv_E c'(N_1\theta,\ldots,N_n\theta)$
であり，Eは強完全性を満足しているから各$M_i\theta$，$N_i\theta$には対応する項M_i'，N_i'（ともに$T[\Sigma_0]$の要素）があり　$M_i\theta\equiv_E M_i'$，　$N_i\theta\equiv_E N_i$
したがって$c(M_1',\ldots,M_n')\equiv_E c'(N_1',\ldots,N_n')$ Eは無矛盾と仮定したから
$c(M_1',\ldots,M_n')\equiv_{E0} c'(N_1',\ldots,N_n')$
性質1より書くM_i'，N_i'に対して，更に$T[\Sigma_0^c]$に属するM_i''，N_i''が存在して$M_i'\equiv_E M_i''$，$N_i'\equiv_E N_i''$
$c(M_1'',\ldots,M_n'')\equiv_E c'(N_1'',\ldots,N_n'')$
かつ
$c(M_1'',\ldots,M_n'')\neq c'(N_1'',\ldots,N_n'')$
ゆえに性質2に反し矛盾であることが示せた．■

以上の議論よりアルゴリズム（a）（ii）で用いられている条件についての系を得る．

【系 4.1.9】
Mが構成子を根にもつ項であるとしNと同じ合同関係に入るとする．すなわち$M=c(M_1,\ldots,M_n)$
$c\in\Sigma_0^c$，かつ$M\equiv_E N$のとき
1．$N=c'(N_1,\ldots,N_n)$
$c\neq c'\qquad c,c'\in\Sigma_0^c$
であれば，Eは，矛盾しており，また，
2．Nが変数のときもEは矛盾している．　□

アルゴリズムでは，この系の条件を満たすときには，矛盾であることをしめすdiproofを出力して止まる．

以上で図4-1に示したアルゴリズムについて基本的な方針をあたえた．本節で述べられた事柄がオリジナルのKnuth-Bendixアルゴリズムに付加される中心的部分であり，この部分によってKnuth-Bendixアルゴリズムが$\sim_E$を判定できるように改良される．

4－2　アルゴリズムの能力

本節では，前節で示された判定アルゴリズムから得られる結果について考察する．

アルゴリズムに与えられる規則Rはチャーチロッサーかつネーター的であるとする．また，書き換え系を等式とみなしたときには強完全かつ無矛盾であるとする．

このような初期条件のもとで本アルゴリズムは，つぎつぎと新しい等式と書き換え規則を作り上げてゆく．そこで，各スッテプにおいて，新に作り上げられる書き換え規則および等式をそれぞれ，E_{i+1}，R_{i+1}とし，ステップiのときと比較してみる．$L_i=R_i\cup E_i$とし$\mathfrak{C}_i$を$L_i=R_i\cup E_i$から作られる最小の合同関係とする．これに関しては2つの場合がある．
1つは，アルゴリズム（a），（i）で，
$$R_{i+1}=R_i$$
$$E_{i+1}=E_i\cup\{M_i=N_i\}-\{M=N\}$$
となるときで，このステップでは，$\mathfrak{C}_i\subset\mathfrak{C}_{i+1}$となる．
もう1つは他のステップにおいてで，$\mathfrak{C}_{i+1}=\mathfrak{C}_i$となることがわかる．
したがって，双方のステップにおいて命題4.1.1，命題4.1.2よりL_iが強完全で性質1を満たしていれば，L_{i+1}も同様にその2つを満たすことがわかる．

また，はじめに与えられた書き換え規則がチャーチロッサーかつネーター的であり，さらに，強完全かつ無矛盾であるとすると，L_0も強完全で性質1を満たしている．
このように本アルゴリズムは$L_{i+1}=L_i$，$L_i\subset L_{i+1}$とにわけて考えることができる．

【補題 4.2.1】
L_iが無矛盾であるときL_{i+1}も無矛盾でありそれらの終モデルは存在すれば一致する．すなわち，
$$F_{\mathfrak{C}}(L_i)=F_{\mathfrak{C}}(L_{i+1})$$　□
（証明）
$L_{i+1}=L_i$の場合にはあきらかである．

初期条件

調べる等式　$t = t$　　：判定する等式

整そ順序　$>$　　：停止性を満たすようにするもの

$E_0 := \{t = t'\}$

$R_0 := R$　　：ＦＰを記述している書き換え規則

$i := 0$

$P :=$ R中のルールの数　：書き換え規則に番号付を行う

Loop

while $E \neq \phi$ do

begin

等式の書き換え

Eから任意のM＝Nを選ぶ；　$E_{i+1} = E_i - \{M = N\}$

R_iを用いてM↓，N↓を求める

1. if M↓＝N↓then $E_{i+10} := E_i$；$R_{i+1} := R_i$；$i := i + 1$
2. else if $M\downarrow \in T[\Sigma, X]_{s0 \in S0}$
   (1) then if $N\downarrow \in T[\Sigma, X]_{s0 \in S0}$
   then 以下のことを行う
   (a) if $M\downarrow = C(M_1, \ldots, M_n)$，$C \in \Sigma_0{}^c$
   (i) then if $N\downarrow = C(N_1, \ldots, N_n)$
   then $E_{i+1} := E_i \cup \{M_m = N_m \mid 1 \leqq m \leqq n\}$；$R_{i+1} := R_i$；$i := i + 1$
   (ii) else if $N\downarrow = C'(N_1, \ldots, N_n)$，$C' \in \Sigma_0{}^c$，$C' \neq C$
   or　　then "disproof"
   N↓は変数
   (iii) else if $N\downarrow > M\downarrow$ then $l := N\downarrow$；　$r := M\downarrow$
   (iv) else　"Stop with Faulure"
   (b) else if $N\downarrow = C(M_1, \ldots, M_n)$，$C \in \Sigma_0{}^c$ then　M↓，N↓を入れ替えて(i)~(iv)を行う
   (c) else if　$N\downarrow > M\downarrow$ then　$l := N\downarrow$；　$r := M\downarrow$
   (d) else if　$M\downarrow > N\downarrow$ then　$l := M\downarrow$　　$r := N\downarrow$
   (e) else　"Stop with Faulure"
   (2) else "Stop with Faulure"
3. else if　$N\downarrow \in T[\Sigma, X]_{s0 \in S0}$ then　(1)～(2)と同様.
4. else if　$N\downarrow > M\downarrow$ then　$l := N\downarrow$；　$r := M\downarrow$
5. else if　$M\downarrow > N\downarrow$ then　$l := M\downarrow$；　$r := N\downarrow$
6. else "Stop with Faulure"

新しい書き換え規則の追加.

$K := \{k \mid l_k \triangleright r_k$，$l_k$が　$l \triangleright r$　によってl_k'に書き換えられる.$\}$

$E_{i+1} := E_i \cup \{l_k' = r_k \mid k \in K\}$

$p := p + 1$

$R_{i+1} := \{j \mid l_j \triangleright r_{k'}$　$j \notin K\} \cup \{p \mid l \triangleright r\}$

（この時，危険対を調べたマークは保存する．）

危険対を求める.

if　R_i中のすべての書き換え規則が，マークされている．　then "Stop with Success"

else　R_i中から，マークされていない最小のkを選び，マークする

$E_{i+1} := \{l_k \triangleright r_k$　との危険対全体$\}$

$R_{i+1} := R_i$

$i := i + 1$

Loop end

図４－１　判定アルゴリズム

また，$L_i \subset L_{i+1}$の場合，系4.1.7による． ■

この補題の特別な場合としてつぎの系を示すことができる．

【系 4.2.2】
$E \cup \{t = t'\}$ が無矛盾であればL_iも無矛盾でありそれらの終モデルは存在すれば一致する． □

以上の考察のもとに本アルゴリズムの能力を示すつぎの定理を証明できる．

【定理 4.2.3】
アルゴリズムに与えられた書き換え規則がチャーチロッサーかつネーター的で，等式とみなしたときには強完全かつ無矛盾で，性質1，2を満たしているとき，アルゴリズムに$t = t'$が入力されて
1．Success で停止すれば終モデル上で成立する．
2．Dissproof で停止すれば終モデル上では成立しない
3．終モデル上で成立しないときにはFailure または Disproofで停止する □

(証明)
1．ステップnで停止したとする．
このときに，得られた項書き換え系R_nは前節での議論からチャーチロッサーかつネーター的であり，
$E' = E \cup \{t = t'\}$ を含んでいる．さらに，仮定からR_nは等式とみなすと，強完全であり命題4.1.2 より性質1をみたす．また付け加えられる規則に注目すると性質2をも満足することがわかる．

以上の準備のもとに，R_nが無矛盾であることを背理法によって示す．
R_nが矛盾しているとすると，$T[\Sigma_0]$上の項M，Nにおいて，$M \equiv_E N$であるが$M \equiv_{E0} N$でないものがある．
しかしながら，Eは性質1を満たしているから，構成子のみからなる項M_0，N_0があり，$M_0 \neq_{E0} N_0$と$M_0 \equiv_E N_0$がともに成立する．しかしながらここで，M_0，N_0はともにR_nにおいて正規項となるから，$M_0 \equiv_E N$であれば，$M_0 = N_0$でなくてはならない．したがって$M_0 \equiv_{E0} N_0$．よって矛盾が導けた．結局E'は無矛盾であるから補題4.1.3より，$t = t'$が終モデルで成り立ち系4.1.4よりその終モデルは一致する．
2．L_nが矛盾していることでありこれはE'が矛盾していることである．補題4.1.4，系4.1.9より$t = t'$は終モデルで成立しない．
3．$t = t'$が，終モデル上で等しくないとする．
このとき補題4.2.1よりE'は矛盾していることになる．
ここで，M，Nを，$M \equiv_E N$であるが，$M \equiv_{E0} N$でない項とし，アルゴリズムが停止しない場合を考えてみる．これはアルゴリズムの性質より合流性をみたす書き換え規則を無限に作成しているときであり，MとNは同じ項へと書き換えができるはずである．しかしながらM，Nは構成子のみよりなるから，それらに対する書き換え規則は存在しない．したがってMとNは等しくなくてはならずこれは仮定に反する． ■

5．結論

FPの代数則を検証するためのシステムに対する基礎を与えた．現在本システムの実現方法について検討している．

参考文献

[Backus 78] Backus,J.H. : Can Programming be Liberated from the Von Neumann Style ?. A Functional Style and Its Algebra of Programs. Comm. ACM Vol.20 No.8 (1978) pp.613-641

[Backus 81a] Backus,J.H. : Function Level Programs as Mathmatical Objects. Conf. on Functional Programming language and Computer Architeture ACM (1981) pp.1-10

[Heut 81] Huet,G : A Complete Proofs of Correctness of Knuth-Bendix Completion Algorithm. JCSS Vol. 23 (1981) pp.11-21

[Heut 82] Huet,G. and Hullot,J.M. : Proofs by induction in Equational Theries with Constructors. JCSS Vol. 25 (1982) pp.239-265

[Muller 84a] Muller B : Busy and Lazy FP with Infinite Objects. Conf. on Lisp and Functional Programming ACM (1984) pp.282-292

[Muller 84b] Muller B : An Algebraic Semantics for Busy and Lazy Evaluation and It's application to Functional Langages. Lecture Notes in Computer Science ICALP 83 Automata,Language and Programming. Springer Verlag (1984) pp.513-526

[Tang 84] 唐,米田：関数方程式の展開について 理化学研究所シンポジウム関数的プログラミング (1984)

[Togashi 85] 富樫,布川,野口：A Proof Procedure for the Algebra of FP Programs.ICOT Tech. Memo. (to apper)

[Toyama 82a] 外山芳人：項書き換え系の直和について AL82-39 (1982)

[Toyama 82b] 外山芳人：ラベル付き項書き換えシステム AL82-40 (1982)

[Song 84] 孫,黄：準双線形方程式の展開算法 理化学研究所シンポジウム関数的プログラミング (1984)

[Williams 81] Williams,J.H. : On the Development of the Algebra of Functional Programming. Lecture Notes in Computer Science Vol.107 pp.733-757(1981)

[Wand 79] Wand M. : Final Algebra Semantics and Data Type Extensions JCSS Vol.19 No.1 (1979)