ソフトウェア基礎論　35－5
(1990. 5. 24)

# 代数的仕様の検証のための被覆集合帰納法

酒井正彦　　坂部俊樹　　稲垣康善
名古屋大学　工学部

あらまし　代数的仕様の検証は，典型的には，等式で表現される性質がその仕様の意味する代数の上で成立することを証明することである．仕様の意味する代数が仕様を満たす代数のクラスの始代数であるときは，この検証は，仕様を等式理論とみなし，性質を表す等式がその理論の帰納的定理であることを証明することに対応する．帰納的定理の証明法にはKnuth-Bendixアルゴリズムに基づくInduction-less Inductionや構造帰納法によるものがある．

本稿では，基礎項集合の帰納的定義に基づいた単純な構造帰納法の拡張である被覆集合帰納法を提案し，その正当性を証明する．同様な拡張された構造帰納法として，Zhang, Kapur, Krishnamoorthy[11]が提案した方法があるが，それに比べると単純な帰納法であるが，能力は互いに含まれないという関係にある．

# Cover Set Induction Principle for Algebraic Specifications

Masahiko SAKAI,　Toshiki SAKABE,　Yasuyoshi INAGAKI
Fuculty of Engineering,　Nagoya University

Abstract　Typical verification of an algebraic specification is to prove that the specified algebra satisfies an equation which represents some property of the algebra. If the specified algebra is defined as the initial algebra of the class of all algebras satisfying the specification, the verification corresponds to proving that the equation is an inductive theorem of the specification. Inductive theorems of an equational theory are usually proved by induction-less induction based on Knuth-Bendix algorithm or structural induction.

In this paper, an induction principle, called the cover set induction, is proposed. It is an extension of the usual induction on the structure of the set of ground terms. Compared to the extended structural induction proposed by Zhang, Kapur and Krishnamoorthy[11], our induction is simpler, while its proving power is not less than that of Zhang, et.al.

## 1 はじめに

等式を用いて代数的に記述された抽象データ型の仕様が目的とする性質を満たすことの検証は，仕様に基づいて作成される，あるいは，自動生成されるソフトウエアの信頼性の向上のために重要である．

代数的仕様の検証は，典型的には，等式で表現される性質がその仕様の意味する代数の上で成立することを証明することである．仕様の意味する代数が仕様を満たす代数のクラスの始代数であるときは，この検証は，仕様を等式理論とみなし，性質を表す等式がその理論の帰納的定理であることを証明することに対応する．

帰納的定理の機械的証明方法としては，Induction-less Induction と呼ばれる手法が研究されてきている [8, 5, 6, 9, 10, 7]．これらの方法では，いずれも，Knuth-Bendix アルゴリズムを利用しており，次のような問題点がある．

(1) 手続きが無限ループに陥り易い．

(2) 項の上の順序の与え方によって，証明に失敗したり停止しなかったりする．

(3) 否定的な結果が導かれた場合，すなわち，等式が帰納的定理ではないことが証明されたとき，仕様のどの部分にその原因があるのかが実行結果から分りづらい．

(1) と (2) は，Induction-less Induction の手法で証明に失敗したり停止しなかった場合に，項の上の順序の与え方や等式の適用の戦略を変更して証明を繰り返し試みる必要性があることを表している．また，(3) は，証明の過程が直観的には理解しづらいためであり，仕様が満たすべき性質を満たすように，仕様を変更するための指針が得にくいことを意味している．すなわち，仕様の検証を通じた仕様の誤りの修正には，あまり適していない．

これに対して，伝統的な帰納法の証明法は，(2) については問題がなく，また，(1),(3) に対しても有利であると考えられる．実際に，被覆集合 (cover set) に基づく帰納法が Zhang らにより提案されている [11]．

著者らも，Zhang らとほぼ同時期から，代数的仕様の帰納的定理の帰納法に基づく証明を行う際にこれを支援する手続き，ならびに，その手続きの正当性について研究を行ってきた [12]．

本論文では，代数的仕様の帰納的定理の証明に有効となる被覆集合帰納法を提案し，その能力について他の方法と比較検討する．

以下，2. では準備として記法や基本的な定義を与え，3. では代数的仕様の帰納的定理とその構造帰納法に基づく証明を形式化する．4. では被覆集合帰納法とその拡張を提案する．さらに，5. では Zhang らの方法との比較を行う．

## 2 準備

シグニチャ $\Sigma$ は，ソートの集合 $S$ と関数記号の集合 $F$，および，二つの関数 $\mathrm{arity} : F \to S^*$, $\mathrm{sort} : F \to S$ からなる 4 項組である．ここで，$S^*$ は $S$ の元から生成されるすべての有限系列の集合である．また，$f \in F$ に対して，$\mathrm{arity}(f) = w$, $\mathrm{sort}(f) = s$ のとき，$f$ をアリティ $w$, ソート $s$ の関数記号であるという．特に，$\mathrm{arity}(f) = \epsilon$ (空系列) のとき，$f$ は定数記号と呼ばれる．

変数の集合を $V$，変数のソートを表す関数を $\mathrm{sort}_V : V \to S$ とするとき，次の条件 (1),(2) を満たす最小の集合を $\mathcal{T}[F \cup V]$ とする．

(1) $x \in V$ ならば，

$$x \in \mathcal{T}[F \cup V], \text{ かつ,}$$
$$\mathrm{sort}_{\mathcal{T}[F\cup V]}(x) = \mathrm{sort}_V(x)$$

(2) $f \in F$, $\mathrm{arity}(f) = s_1 \cdots s_n$, $t_i \in \mathcal{T}[F \cup V]$, $\mathrm{sort}(t_i) = s_i\,(i = 1, \cdots, n)$ ならば，

$$f(t_1, \cdots, t_n) \in \mathcal{T}[F \cup V], \text{ かつ,}$$
$$\mathrm{sort}_{\mathcal{T}[F\cup V]}(f(t_1, \cdots, t_n)) = \mathrm{sort}(f).$$

$\mathcal{T}[F \cup V]$ の要素を $F \cup V$ 項，あるいは，単に項という．特に $V = \phi$ のときには，$F$ 項，もしくは，基礎項という．以下では，$\mathrm{sort}_V$, $\mathrm{sort}_{\mathcal{T}[F\cup V]}$ を単に sort と書く．また，基礎項 $\xi$ の大きさを，$\xi$ に含まれる関数記号の数，すなわち，$\xi$ を木として表現したときのノードの数とし，$|\xi|$ と表記する．

同一ソートの変数と項の対からなる有限集合 $\{\langle x_1, \xi_1\rangle, \cdots, \langle x_n, \xi_n\rangle\}$ を代入と呼ぶ．また，変数 $y$ への $\theta$ の代入 $y\theta$ は次のように定義される．

$$y\theta = \begin{cases} \xi_i & y = x_i \text{ のとき} \\ y & \text{otherwise} \end{cases}$$

$\xi_1, \cdots, \xi_n$ がすべて基礎項であるような代入は基礎代入と呼ばれる．以下では，すべての基礎代入の集合を $\Theta$ で表す．

同一ソートの項 $\xi, \eta \in \mathcal{T}[F \cup V]$ の間に等号記号 $\approx$ をはさんで並べた系列 $\xi \approx \eta$ を等式という．等式の集合は等

$$
\begin{aligned}
&S_{\mathtt{Mod}} = \{\ \mathtt{nat}\}\\
&F_{\mathtt{Mod}} = C_{\mathtt{Mod}} \cup D_{\mathtt{Mod}}\\
&\quad C_{\mathtt{Mod}} = \{\ 0 : \rightarrow \mathtt{nat},\\
&\qquad\qquad \mathtt{Suc} : \mathtt{nat} \rightarrow \mathtt{nat}\}\\
&\quad D_{\mathtt{Mod}} = \{\ \mathtt{Mod} : \mathtt{nat} \rightarrow \mathtt{nat}\}\\
&E_{\mathtt{Mod}} = \{\ \mathtt{Mod}(0) \approx 0,\quad \mathtt{Mod}(\mathtt{Suc}(0)) \approx \mathtt{Suc}(0),\\
&\qquad\qquad \mathtt{Mod}(\mathtt{Suc}(\mathtt{Suc}(x))) \approx \mathtt{Mod}(x)\}
\end{aligned}
$$

図 1: 自然数の 2 の剰余の仕様

式理論と呼ばれる．等式理論 $E$ から等式論理の推論規則 [14] を用いて $\xi \approx \eta$ が導出されるとき，$E \vdash \xi \approx \eta$ と書く．

## 3 帰納的定理と構造帰納法

シグニチャ $\Sigma$，変数集合 $V$，等式の有限集合 $E$ の 3 項組 $\langle \Sigma, V, E\rangle$ は代数的仕様と呼ばれる．3. では代数的仕様における帰納的定理を定義し，この証明に対する構造帰納法の定式化を行う．

### 3.1 帰納的定理

まず，帰納的定理を定義しよう．

**定義 3.1** $E$ を等式理論，$\xi$,$\eta$ を項とするとき，任意の基礎代入 $\theta \in \Theta$ に対して $E \vdash \xi\theta \approx \eta\theta$ ならば，$E \vdash_{\mathrm{ind}} \xi \approx \eta$ と書き，「$E$ の下で $\xi$ と $\eta$ が帰納的に等しい」，もしくは，「$\xi \approx \eta$ は $E$ の帰納的定理である」という． □

一般には，$E \vdash_{\mathrm{ind}} \xi \approx \eta$ であっても，$E \vdash \xi \approx \eta$ とはならないので，注意されたい．例えば，図1は，自然数を $0, \mathtt{Suc}(0), \mathtt{Suc}(\mathtt{Suc}(0)), \cdots$ と表現し，自然数の 2 の剰余を求める関数 Mod の仕様である．この仕様に対して，等式 $\mathtt{Mod}(\mathtt{Mod}(x)) \approx \mathtt{Mod}(x)$ は $E_{\mathtt{Mod}}$ の定理ではないが，帰納的定理である．すなわち，

$$E_{\mathtt{Mod}} \quad \forall \quad \mathtt{Mod}(\mathtt{Mod}(x)) \approx \mathtt{Mod}(x) \qquad (1)$$

$$E_{\mathtt{Mod}} \quad \vdash_{\mathrm{ind}} \quad \mathtt{Mod}(\mathtt{Mod}(x)) \approx \mathtt{Mod}(x). \qquad (2)$$

### 3.2 構成子に基づく方法

Huet ら [6] のように，演算記号が構成子 (constructor) と非構成子 (defined symbol) に分けられていると考える．すなわち，代数的仕様 $\langle \Sigma, V, E\rangle$ に対して，関数記号の集合 $F$ が $C$ と $D$ に分割されており，任意の基礎項 $\xi$ に対して，$E \vdash \xi \approx \eta$ を満たす構成子のみからなる基礎項 $\eta \in \mathcal{T}[C]$ が存在するとする[1]．このような $F$ の分割は必ず存在し (例えば，$D = \phi$)，また，決定可能な十分条件が知られている [6, 7]．以下では，仕様に対して，そのような分割が与えられているものとして議論を進める．

構成子のみから構成される基礎項を構成子基礎項と呼ぶ．構成子以外の関数記号を含む基礎項には，$E$ の下でそれと等しいことが推論できる構成子基礎項が存在することから，次の命題が成り立つ．

**命題 3.2** $e$ を等式とする．任意の構成子基礎項の代入 $\theta$ に対して $E \vdash e\theta$ であるとき，かつそのときに限り $E \vdash_{\mathrm{ind}} e$ である． □

命題 3.2より，$e$ の変数に任意の構成子基礎項を代入したものが $E$ から推論できれば，$e$ が $E$ の帰納的定理であることが示される．すなわち，次の定理が成り立つ．

**定理 3.3 (構成子に基づく証明法)** $e$ を等式とし，$x_1, \cdots, x_k$ を等式に出現する変数とする．このとき，$\mathrm{sort}(\xi_i) = \mathrm{sort}(x_i)$ を満たす任意の $\xi_i \in \mathcal{T}[C]$ に対して，

$$E \vdash e[x_1 \leftarrow \xi_1, \cdots, x_k \leftarrow \xi_k]$$

が成り立つならば，

$$E \vdash_{\mathrm{ind}} e. \qquad \square$$

### 3.3 構造帰納法

構成子からなる項は通常無限個存在するので，定理 3.3に基づく方法は非現実的である．そのため，しばしば，帰納法が用いられる．種々の帰納法のうちで，Burstall によって提案された構造帰納法 [1] は，複雑な構造を持つデータの上で適用できるという点で優れたものである．

まず，代数的仕様の帰納的定理の証明に対して構造帰納法を定式化するための準備としてメタ変数を導入する．メタ変数は，直観的には，任意の基礎項を表現するものである．

メタ変数の集合を $MV$，メタ変数のソートを定義する関数を $\mathrm{sort}_{MV} : MV \rightarrow S$ とする．(ソートを定める他の関数と同様に，この関数を sort と略記する．) メタ変数を含む等式理論は，同一ソートの項 $\xi$,$\eta \in \mathcal{T}[F \cup V \cup MV]$ からなる等式の集合である．等式論理の推論においては，メタ変数は定数記号と同等に扱われる．

[1] Huet らの条件とは異なり，基礎項 $\eta \in \mathcal{T}[C]$ の唯一性は要求しない．

メタ変数の概念の導入により，構造帰納法は次のように定式化できる．

**定理 3.4 (構造帰納法)** $e$を等式とし，$x$を等式に出現する変数でそのソートを$s$とする．このとき，任意の$f \in C$に対して，

$$E \cup \{e[x \leftarrow p_i] \mid \mathrm{sort}(p_i) = s,\ 1 \leq i \leq k\} \vdash e[x \leftarrow f(p_1, \cdots, p_k)]$$

ならば，

$$E \vdash_{\mathrm{ind}} e.$$

ここで，$p_1, \cdots, p_k$は，$\mathrm{sort}(p_1) \cdots \mathrm{sort}(p_k) = \mathrm{arity}(f)$を満たすメタ変数である．□

## 4 被覆集合帰納法とその拡張

### 4.1 被覆集合帰納法

代数的仕様記述法では，多ソート，かつ，複雑な構造のデータを扱うことができるため，構造帰納法では不十分である．例として，図1の自然数の2の剰余の仕様を考えよう．

この仕様において，等式$\mathtt{Mod}(\mathtt{Mod}(x)) \approx \mathtt{Mod}(x)$が$E_{\mathtt{Mod}}$の帰納的定理であること，すなわち，

$$E_{\mathtt{Mod}} \vdash_{\mathrm{ind}} \mathtt{Mod}(\mathtt{Mod}(x)) \approx \mathtt{Mod}(x) \tag{3}$$

を構造帰納法によって証明するためには，$p$をソートnatのメタ変数とすれば，次の二つを示す必要がある．

$$E_{\mathtt{Mod}} \vdash \mathtt{Mod}(\mathtt{Mod}(0)) \approx \mathtt{Mod}(0), \tag{4}$$

$$E_{\mathtt{Mod}} \cup \{\mathtt{Mod}(\mathtt{Mod}(p)) \approx \mathtt{Mod}(p)\} \vdash \mathtt{Mod}(\mathtt{Mod}(\mathtt{Suc}(p))) \approx \mathtt{Mod}(\mathtt{Suc}(p)) \tag{5}$$

しかしながら(5)は成り立たないため構造帰納法では証明できない．また，この例の他にも，仕様が相互再帰的に定義されている場合にはやはり構造帰納法では証明できない．

これらの欠点を改良するため，被覆集合の概念を導入することによって，構造帰納法を拡張する．そのために，まず，被覆集合と関数SupTを定義する．

**定義 4.1 (被覆集合)** ソート$s$の$(C \cup MV)$項の集合$M$は，次の性質を満たすときソート$s$の被覆集合と呼ばれる．

> ソート$s$の任意の$C$項$\eta$に対して，$\eta = \sigma(\xi)$を満たす項$\xi \in M$と代入$\sigma \in \Theta_{MV}$が存在する．

□

**定義 4.2** 関数$\mathrm{SupT} : \mathcal{T}[F \cup MV] \to 2^{\mathcal{T}[F \cup MV]}$は次のように定義される．

$$\begin{aligned}&\mathrm{SupT}(\xi)\\ &= \{\ \eta \in \mathcal{T}[F \cup MV] \mid \eta[p \leftarrow \eta'] = \xi,\\ &\quad \eta' \in \mathcal{T}[F \cup MV],\ p \in \mathrm{MV}(\eta'),\ \eta \neq \xi\ \}\end{aligned}$$ □

直観的には，SupTは，図2に示すように，$\xi$中のメタ変数$p$への根からのパス上の関数記号で$p$と同一ソートのものを$p$で置き換えて得られる項の集合を返す．

$(F \cup MV)$項$\xi, \eta$に対して，$\eta \in \mathrm{SupT}(\xi)$であることを，$(F \cup MV)$項上の関係として捉えて，$\xi \succ_{\mathrm{SupT}} \eta$，もしくは，$\eta \prec_{\mathrm{SupT}} \xi$と書くことにする．メタ変数への基礎項の代入の集合を$\Theta_{MV}$とする．

次の命題が成り立つので，$\succ_{\mathrm{SupT}}$は無限減少列の存在しない半順序であることがわかる．

**命題 4.3** 任意の$(F \cup MV)$項$\xi, \eta$，および，任意のメタ変数への基礎項の代入$\theta \in \Theta_{MV}$に対して，$\xi \succ_{\mathrm{SupT}} \eta$ならば，$|\xi\theta| > |\eta\theta|$である．

〈**証明**〉 $\xi \succ_{\mathrm{SupT}} \eta$ならば，$\eta \in \mathrm{SupT}(\xi)$が成り立つ．SupTの定義より，$\eta \in \mathrm{SupT}(\xi)$に対して$\eta[p \leftarrow \eta'] = \xi$を満たす$\eta' \in \mathcal{T}[C \cup MV]$と$p \in MV(\eta')$が存在する．$\eta \neq \xi$より，$p \in \eta$かつ$p \neq \eta'$である．いま，

$$\begin{aligned}|\xi\theta| &= |\eta[p \leftarrow \eta']\theta|\\ &= |\eta[p \leftarrow \eta'\theta]\theta|\\ &= |\eta\theta| + n(|\eta'\theta| - |p\theta|).\end{aligned}$$

ここで，$n > 0$は$\eta$中における$p$の出現の数である．ところが，$p \in MV(\eta'),\ p \neq \eta'$であることから，$|\eta'\theta| > |p\theta|$．ゆえに，$|\xi\theta| > |\eta\theta|$が成り立つ．□

これらの概念を用いると構造帰納法は次のように拡張できる．

**定理 4.4 (被覆集合帰納法)** $e$を等式，$x$を等式に出現する変数でそのソートを$s$とする．また，$M$をソート$s$の被覆集合とする．このとき，$M$の任意の要素$\xi$に対して，

$$E \cup \{e[x \leftarrow \eta] \mid \xi \succ_{\mathrm{SupT}} \eta\} \vdash e[x \leftarrow \xi] \tag{6}$$

が成り立つならば，

$$\forall \zeta \in \mathcal{T}[C],\ E \vdash e[x \leftarrow \zeta]. \tag{7}$$

すなわち，

$$E \vdash_{\mathrm{ind}} e \tag{8}$$

□

この定理を証明するために補題を用意する．

**補題 4.5** $\xi, \xi'$ を $(F \cup V \cup \mathrm{MV})$ 項とする．このとき，$E \vdash e$ ならば，任意の $\theta \in \Theta_{MV}$ に対して，$E\theta \vdash e\theta$ ．
〈**略証**〉推論において，等式中のメタ変数は定数と同様に扱われるため，$E \vdash e$ の証明から $E\theta \vdash e\theta$ の証明が直接的に構成できる． □

〈**定理 4.4の証明**〉 式(7)が成り立たないと仮定すると，$E \not\vdash e[x \to \zeta]$ となる構成子基礎項 $\zeta \in \mathcal{T}[C]$ が存在する．そのようなζの内で $|\zeta|$ が最小のものを $\zeta'$ と置く．被覆集合の定義より，$\zeta' = \xi\theta$ を満たす $\xi \in M$ と $\theta \in \Theta_{MV}$ が存在する．したがって，$E \not\vdash e[x \leftarrow \xi\theta]$ である．命題4.3より $\xi \succ_{\mathrm{SupT}} \eta$ をみたす，任意の $\eta$ に対して $|\xi\theta| \succ_{\mathrm{SupT}} |\eta\theta|$ であるから，$\zeta'$ の最小性より，

$$\forall \eta \prec_{\mathrm{SupT}} \xi, \quad E \vdash e[x \leftarrow \eta\theta] \tag{9}$$

が成り立つ．また，式(6)に対して補題4.5を用いると，

$$E \cup \{e[x \leftarrow \eta\theta] | \xi \succ_{\mathrm{SupT}} \eta\} \vdash e[x \leftarrow \xi\theta] \tag{10}$$

が得られる．式(9),(10)より，$E \vdash e[x \leftarrow \xi\theta]$ が導かれるため，(7)が成り立たないという仮定に矛盾し，式(7)が導かれる．これと，命題3.2から，$E \vdash_{\mathrm{ind}} e$ が得られる．
□

この帰納法を用いて，図1の仕様において帰納的定理3が成り立つことを証明しよう．まず，ソート nat の被覆集合 $M_{\mathtt{Mod}}$ を次のようにとる．

$$M_{\mathtt{Mod}} = \{0, \mathtt{Suc}(0)), \mathtt{Suc}(\mathtt{Suc}(p))\}$$

$M_{\mathtt{Mod}}$ の各々の要素に対して次のように証明できる．

(i) $0 \in M_{\mathtt{Mod}}$ に対して，

$$E_{\mathtt{Mod}} \vdash \mathtt{Mod}(\mathtt{Mod}(0)) \approx \mathtt{Mod}(0)$$

(ii) $\mathtt{Suc}(0) \in M_{\mathtt{Mod}}$ に対して，

$$E_{\mathtt{Mod}} \vdash \mathtt{Mod}(\mathtt{Mod}(\mathtt{Suc}(0))) \approx \mathtt{Mod}(\mathtt{Suc}(0))$$

(iii) $\mathtt{Suc}(\mathtt{Suc}(p)) \in M_{\mathtt{Mod}}$ に対して，

$$\begin{aligned} E_{\mathtt{Mod}} \cup \{&\mathtt{Mod}(\mathtt{Mod}(p)) \approx \mathtt{Mod}(p), \\ &\mathtt{Mod}(\mathtt{Mod}(\mathtt{Suc}(p))) \approx \mathtt{Mod}(\mathtt{Suc}(p))\} \\ \vdash\ &\mathtt{Mod}(\mathtt{Mod}(\mathtt{Suc}(\mathtt{Suc}(p)))) \approx \mathtt{Mod}(\mathtt{Suc}(\mathtt{Suc}(p))) \end{aligned}$$

以上で述べた拡張構造帰納法は，単一変数に対して帰納法を適用するものであるが，これをさらに多変数に対して適用する帰納法に拡張することができる．そのための準備として，被覆集合を拡張する．

**定義 4.6** $(C \cup MV)$ 項の $n$ 項組の集合 $M$ は，次の性質を満たすとき $s_1 \times \cdots \times s_n$ の被覆集合と呼ばれる．

> $\mathrm{sort}(\eta_i) = s_i$ を満たす任意の $\eta_1, \cdots, \eta_n$ に対して，$\eta_1 = \sigma(\xi_1), \cdots, \eta_n = \sigma(\xi_n)$ を満たす $\langle \xi_1, \cdots, \xi_n \rangle \in M$ とメタ変数への代入 $\sigma \in \Theta_{MV}$ が存在する．

□

以下では，辞書的順序などを用いて，$\succ_{\mathrm{SupT}}$ が $(F \cup MV)$ 項の n 項組上の半順序に拡張されているとして議論する．

**定理 4.7 (多変数に対する被覆集合帰納法)** $e$ を等式，$x_1, \cdots, x_n$ を等式に出現する変数，$M$ を $\mathrm{sort}(x_1) \times \cdots \times \mathrm{sort}(x_n)$ の被覆集合とする．このとき，$M$ の任意の要素 $\langle \xi_1, \cdots, \xi_n \rangle$ に対して，

$$\begin{aligned} E \cup \{&e[x_1 \leftarrow \xi'_1, \cdots, x_n \leftarrow \xi'_n] \\ &\quad | \langle \xi_1, \cdots, \xi_n \rangle \succ_{\mathrm{SupT}} \langle \xi'_1, \cdots, \xi'_n \rangle\} \\ &\vdash e[x_1 \leftarrow \xi_1, \cdots, x_n \leftarrow \xi_n] \end{aligned}$$

ならば，

$$\forall \zeta_1, \cdots, \zeta_n \in \mathcal{T}[C], \quad E \vdash e[x_1 \leftarrow \zeta_1, \cdots, x_n \leftarrow \zeta_n] \tag{11}$$

が成り立つ．従って，$E \vdash_{\mathrm{ind}} e$ である． □

この定理は，定理4.4と同様に証明される．

次に，多変数に対する被覆集合帰納法を用いた加算の可換性の証明例を挙げる．図3は，自然数上の加算の仕様である．

図3の仕様について，

$$E_{\mathtt{Add}} \vdash_{\mathrm{ind}} \mathtt{Add}(x, y) \approx \mathtt{Add}(y, x) \tag{12}$$

を証明しよう．ソート nat × nat の被覆集合 $M_{\mathtt{Add}}$ を次のようにとる．

$$\begin{aligned} M_{\mathtt{Add}} = \{&\langle 0, 0\rangle, \langle \mathtt{Suc}(p), 0\rangle, \\ &\langle 0, \mathtt{Suc}(q)\rangle, \langle \mathtt{Suc}(p), \mathtt{Suc}(q)\rangle\} \end{aligned}$$

$M_{\mathtt{Add}}$ の各々の要素に対して次のように証明できる．

(i) $\langle 0, 0\rangle \in M_{\mathtt{Add}}$ に対して，

$$E_{\mathtt{Add}} \vdash \mathtt{Add}(0, 0) \approx \mathtt{Add}(0, 0)$$

(ii) $\langle \mathtt{Suc}(p), 0\rangle \in M_{\mathtt{Add}}$ に対して，

$$\begin{aligned} E_{\mathtt{Add}} \cup \{&\mathtt{Add}(p, 0) \approx \mathtt{Add}(0, p)\} \\ \vdash\ &\mathtt{Add}(\mathtt{Suc}(p), 0) \approx \mathtt{Add}(0, \mathtt{Suc}(p)) \end{aligned}$$

(iii) $\langle 0, \mathtt{Suc}(q)\rangle \in M_{\mathtt{Add}}$ に対して，

$$E_{\mathtt{Add}} \cup \{\mathtt{Add}(0,q) \approx \mathtt{Add}(q,0)\} \vdash \mathtt{Add}(0,\mathtt{Suc}(q)) \approx \mathtt{Add}(\mathtt{Suc}(q),0)$$

(iv) $\langle \mathtt{Suc}(p), \mathtt{Suc}(q)\rangle \in M_{\mathtt{Add}}$ に対して，

$$\begin{aligned} E_{\mathtt{Add}} \cup \{&\mathtt{Add}(p,q) \approx \mathtt{Add}(q,p),\\ &\mathtt{Add}(\mathtt{Suc}(p),q) \approx \mathtt{Add}(q,\mathtt{Suc}(p)),\\ &\mathtt{Add}(p,\mathtt{Suc}(q)) \approx \mathtt{Add}(\mathtt{Suc}(q),p)\}\\ \vdash\ &\mathtt{Add}(\mathtt{Suc}(p),\mathtt{Suc}(q)) \approx \mathtt{Add}(\mathtt{Suc}(q),\mathtt{Suc}(p)) \end{aligned}$$

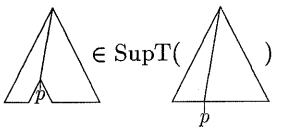


図 2: SupT の性質

### 4.2 被覆集合帰納法の改良

3.3 で述べた被覆集合帰納法は，代数的仕様の帰納的性質の証明に適しているが，まだ，能力に不十分な点がある．以下では，その点をを明らかにし，被覆集合帰納法の改良を行う．

図 4の仕様は，自然数の 2 次元ベクトルの和を定義している．この仕様に対して，

$$E_{\mathtt{AddV}} \vdash_{\mathrm{ind}} \mathtt{AddV}(x,y) \approx \mathtt{AddV}(y,x) \qquad (13)$$

は被覆集合帰納法では証明できない．ここでは，次に示すソート vec × vec の被覆集合を用いてその理由を示すが，他の被覆集合を用いてもやはり証明に失敗する．

$$\begin{aligned} M_{\mathtt{AddV}} = \{&\langle \mathtt{Vec}(0,0), \mathtt{Vec}(0,0)\rangle,\\ &\langle \mathtt{Vec}(\mathtt{Suc}(p),0), \mathtt{Vec}(0,0)\rangle,\\ &\langle \mathtt{Vec}(0,0), \mathtt{Vec}(\mathtt{Suc}(r),0)\rangle,\\ &\quad\vdots\\ &\langle \mathtt{Vec}(\mathtt{Suc}(p),\mathtt{Suc}(q)), \mathtt{Vec}(\mathtt{Suc}(r),\mathtt{Suc}(t))\rangle\} \end{aligned}$$

この中で，$\langle \mathtt{Vec}(0,0), \mathtt{Vec}(\mathtt{Suc}(r),0)\rangle \in M_{\mathtt{AddV}}$ に関しては，次式を示す必要があるが，これは成り立たない．

$$\begin{aligned} E_{\mathtt{AddV}} \cup\ &\{\mathtt{AddV}(\mathtt{Vec}(0,0), \mathtt{Vec}(r,0))\\ &\approx \mathtt{AddV}(\mathtt{Vec}(r,0), \mathtt{Vec}(0,0))\}\\ \vdash\ &\mathtt{AddV}(\mathtt{Vec}(0,0), \mathtt{Vec}(\mathtt{Suc}(r),0))\\ &\approx \mathtt{Add}(\mathtt{Vec}(\mathtt{Suc}(r),0), \mathtt{Vec}(0,0)) \end{aligned} \qquad (14)$$

これをもう少し詳しく観察しよう．等式 (14) の左辺

$$\mathtt{AddV}(\mathtt{Vec}(0,0), \mathtt{Vec}(\mathtt{Suc}(r),0))$$

と右辺

$$\mathtt{Add}(\mathtt{Vec}(\mathtt{Suc}(r),0), \mathtt{Vec}(0,0))$$

$$\begin{aligned} S_{\mathtt{Add}} &= \{\ \mathtt{nat}\}\\ F_{\mathtt{Add}} &= C_{\mathtt{Add}} \cup D_{\mathtt{Add}}\\ C_{\mathtt{Add}} &= \{\ 0 : \rightarrow \mathtt{nat},\\ &\qquad \mathtt{Suc} : \mathtt{nat} \rightarrow \mathtt{nat}\}\\ D_{\mathtt{Add}} &= \{\ \mathtt{Add} : \mathtt{nat}, \mathtt{nat} \rightarrow \mathtt{nat}\}\\ E_{\mathtt{Add}} &= \{\ \mathtt{Add}(x,0) \approx x,\\ &\qquad \mathtt{Add}(x, \mathtt{Suc}(y)) \approx \mathtt{Suc}(\mathtt{Add}(x,y))\} \end{aligned}$$

図 3: 自然数上の加算の仕様

$$\begin{aligned} S_{\mathtt{AddV}} &= \{\ \mathtt{nat}, \mathtt{vec}\}\\ F_{\mathtt{AddV}} &= C_{\mathtt{AddV}} \cup D_{\mathtt{AddV}}\\ C_{\mathtt{AddV}} &= \{\ 0 : \rightarrow \mathtt{nat},\\ &\qquad \mathtt{Suc} : \mathtt{nat} \rightarrow \mathtt{nat},\\ &\qquad \mathtt{Vec} : \mathtt{vec}, \mathtt{vec} \rightarrow \mathtt{vec}\}\\ D_{\mathtt{AddV}} &= \{\ \mathtt{Add} : \mathtt{nat}, \mathtt{nat} \rightarrow \mathtt{nat},\\ &\qquad \mathtt{AddV} : \mathtt{vec}, \mathtt{vec} \rightarrow \mathtt{vec}\}\\ E_{\mathtt{AddV}} &= \{\ \mathtt{Add}(x,0) \approx x,\\ &\qquad \mathtt{Add}(x, \mathtt{Suc}(y)) \approx \mathtt{Suc}(\mathtt{Add}(x,y)),\\ &\qquad \mathtt{AddV}(\mathtt{Vec}(x,y), \mathtt{Vec}(z,w))\\ &\qquad\quad \approx \mathtt{Vec}(\mathtt{Add}(x,z), \mathtt{Add}(y,w))\} \end{aligned}$$

図 4: 自然数の 2 次元ベクトル上の加算の仕様

は，それぞれ，

$$E_{\mathtt{AddV}} \vdash (\text{左辺}) \approx \mathtt{Vec}(\mathtt{Suc}(\mathtt{Add}(0,r)),0) \quad (15)$$

$$E_{\mathtt{AddV}} \vdash (\text{右辺}) \approx \mathtt{Vec}(\mathtt{Suc}(r),0) \quad (16)$$

である．$E_{\mathtt{AddV}}$ と帰納法の仮定である等式

$$\mathtt{AddV}(\mathtt{Vec}(0,0),\mathtt{Vec}(r,0)) \approx \mathtt{AddV}(\mathtt{Vec}(r,0),\mathtt{Vec}(0,0)) \quad (17)$$

を用いて $\mathtt{Vec}(\mathtt{Suc}(\mathtt{Add}(0,r)),0)$ と $\mathtt{Vec}(\mathtt{Suc}(r),0)$ が等しいことが示せればよいのであるが，$E_{\mathtt{AddV}}$ を用いて等式 (17) の両辺を推論しても，次の式しか導けない．

$$E_{\mathtt{AddV}} \vdash \mathtt{Vec}(\mathtt{Add}(0,r),0) \approx \mathtt{Vec}(r,0)$$

この等式は，$E_{\mathtt{AddV}}$ と等式 (17) から導かれたものであるから，これを式 (15) と式 (16) に適用してもよい．実際，式 (15) に適用してこれが式 (16) と等しいことが示されればよいのであるが，式 (15) の関数記号 Vec と Add の間に Suc が入っているためこれは不可能である．もし，等式 (4.2) の代わりに，次の等式

$$\mathtt{Add}(0,r) \approx r \quad (18)$$

のようにより強い等式を帰納法の仮定として用いてよいならば，これを式 (15) に適用して式 (16) と等しいことが示すことができ，証明が可能になる．

以下では，被覆集合帰納法を拡張して，等式 (18) のような帰納法の仮定が用いられるようにする．

$C_0 \subseteq C$ を次の条件を満たす関数記号 $f$ の集合とする．

> $(F \cup V)$ 等式の集合 $E$，および，$(F \cup V)$ 項 $\xi_1,\cdots,\xi_n,\eta_1,\cdots,\eta_n$ に対して，
>
> $$E \vdash f(\xi_1,\cdots,\xi_n) \approx f(\eta_1,\cdots,\eta_n)$$
> $$\textit{iff}$$
> $$E \vdash \xi_1 \approx \eta_1,\ \cdots,\ E \vdash \xi_n \approx \eta_n$$

$E$ を項書換え系として考えれば，左辺の最外に現れない関数記号の集合が $C_0$ に相当する．

次に，$F \cup V \cup MV$ 等式を引数として，$F \cup V \cup MV$ 等式の集合を返す関数 Decmp，および，$F \cup V$ 等式と $C \cup MV$ 項を引数として，$F \cup V \cup MV$ 等式の集合を返す関数 $\mathrm{Deriv_E}$ を次のように定義する．

$$\mathrm{Decmp}(\xi \approx \eta) = \begin{cases} \mathrm{Decmp}(\xi_1 \approx \eta_1) \cup \cdots \cup \mathrm{Decmp}(\xi_n \approx \eta_n) \\ \qquad \xi = f(\xi_1,\cdots,\xi_n), \eta = f(\eta_1,\cdots,\eta_n), \\ \qquad\qquad f \in C_0 \text{のとき} \\ \{\xi \approx \eta\}, \qquad \text{それ以外のとき} \end{cases}$$

$$\mathrm{Deriv_E}(\xi \approx \eta) = \{\xi' \approx \eta' \mid E \vdash \xi \approx \xi', E \vdash \eta \approx \eta'\}$$

直観的には，Decmp は与えられた等式の両辺の根から順に，等しい $C_0$ の関数記号を取り除いてできる等式の集合を返す．また，$\mathrm{Deriv_E}$ は等式の両辺を $E$ で推論して得られるすべての等式の集合を返す．

これらの二つの関数に関する三つの命題が成り立つ．

**命題 4.8** $(F \cup V)$ 等式集合 $E$，$(F \cup V)$ 等式 $e$ に対して，$E \vdash e$ ならば，任意の等式 $e' \in \mathrm{Decmp}(e)$ に対して，$E \vdash e'$ が成り立つ． □

**命題 4.9** $(F \cup V)$ 等式集合 $E$，$(F \cup V)$ 等式 $e$ に対して，$E \vdash e$ ならば，任意の等式 $e' \in \mathrm{Deriv_E}(e)$ に対して，$E \vdash e'$ が成り立つ． □

**命題 4.10** $e$ を $F \cup V \cup MV$ 等式，$\theta \in \Theta_{MV}$ をメタ変数への代入とする．このとき，

$$\mathrm{Decmp}(e\theta) \supseteq \{e'\theta \mid e' \in \mathrm{Decmp}(e)\} \quad (19)$$

□

Decmp はその定義域を $(F \cup V \cup MV)$ 等式の集合に自然に拡張できる．

Decmp と $\mathrm{Deriv_E}$ の概念を用いて被覆集合帰納法を拡張する．ただし，簡単のために一変数の帰納法についてのみ述べることにする．

**定理 4.11 (拡張された被覆集合帰納法)** $e$ を等式，$x$ を等式に出現する変数でそのソートを $s$ とする．また，$M$ をソート $s$ の被覆集合とする．このとき，任意の $\xi \in M$ に対して，

$$E \cup \bigcup_{\eta \succ_{\mathrm{SupT}} \xi} \mathrm{Decmp}(\mathrm{Deriv_E}(e[x \leftarrow \eta])) \vdash e[x \leftarrow \xi] \quad (20)$$

が成り立つならば，

$$\forall \zeta \in \mathcal{T}[C],\ E \vdash e[x \leftarrow \zeta]. \quad (21)$$

ゆえに，

$$E \vdash_{\mathrm{ind}} e.$$

〈**証明**〉 式 (21) が成り立たないと仮定する．$E \nvdash e[x \to \zeta]$ となる構成子基礎項 $\zeta \in \mathcal{T}[C]$ の内で $|\zeta|$ が最小のものを $\zeta'$ と置く．被覆集合の定義より，$\zeta' = \xi\theta$ を満たす $\xi \in M$ と $\theta \in \Theta_{MV}$ が存在する．したがって，$E \nvdash e[x \leftarrow \xi\theta]$ である．$\zeta'$ の最小性と命題 4.3から，

$$\forall \eta \succ_{\mathrm{SupT}} \xi,\ E \vdash e[x \leftarrow \eta\theta] \quad (22)$$

が成り立つ．これに補題4.9と補題4.8を順に用いると

$$\forall e' \in \mathrm{Decmp}(\mathrm{Deriv_E}(e[x \leftarrow \eta\theta])), \forall \eta \succ_{\mathrm{SupT}} \xi, \quad E \vdash e' \tag{23}$$

が成り立つ．

また，式(20)に対して補題4.5を用いると，

$$E \cup \{e'\theta \mid e' \in \mathrm{Decmp}(\mathrm{Deriv_E}(e[x \leftarrow \eta])), \xi \succ_{\mathrm{SupT}} \eta\} \vdash e[x \leftarrow \xi\theta] \tag{24}$$

が得られる．補題4.10および式(23),(24)より，$E \vdash e[x \leftarrow \xi\theta]$ が導かれるため，(21)が成り立たないという仮定に矛盾する． □

$\mathrm{Deriv_E}$ は一般的に無限集合である．したがって，拡張された被覆集合帰納法は，全く実用にならないと思われるかもしれない．しかしながら，$\mathrm{Deriv_E}$ はDecmpと共に用いられるので，$E$ を項書換え規則とみなしたとき停止性を満たす場合には，$\mathrm{Deriv_E}$ の代わりに $E$ の正規形を返す関数を用いるだけで十分である．したがって，そのような仕様に対しては，手続き化可能である．しかし，停止性を満たさない場合については，$\mathrm{Deriv_E}$ が表す無限集合に対して，どのような有限部分集合を求めれば有効であるかは今後の課題である．

## 5 Zhangらの証明法との比較

Zhangらは，著者らとほぼ同時期に被覆集合の概念に基づく帰納法を提案している．本節では，これとの比較を行う．

主な違いは，次の二つである．

(1) 被覆集合の定義が異なる．

(2) 帰納法の仮定の定め方が異なる．

証明能力については，互いに含まれない．すなわち，一方では証明できるのに他方では不可能であるような例が存在する．以下では，Zhangらの方法では証明できないが，著者らの方法では証明できるような帰納的定理の例を挙げる．

図5の仕様は，自然数の対に対して，その各々の要素の2の剰余を求める関数ModPの仕様である．等式 $e$ を $\mathtt{ModP(ModP}(x)) \approx \mathtt{ModP}(x)$ とするとき，$E_{\mathtt{ModP}} \vdash_{\mathrm{ind}} e$ は，著者らの方法では，被覆集合を次のようにとることにより証明できる．

$$M_{\mathtt{ModP}} = \{\ \mathtt{Pair}(0,0), \mathtt{Pair}(\mathtt{Suc}(0),0),$$
$$\mathtt{Pair}(\mathtt{Suc}(\mathtt{Suc}(p)),0), \mathtt{Pair}(0,\mathtt{Suc}(0)),$$
$$\mathtt{Pair}(\mathtt{Suc}(0),\mathtt{Suc}(0)),$$
$$\mathtt{Pair}(\mathtt{Suc}(\mathtt{Suc}(p)),\mathtt{Suc}(0)),$$
$$\mathtt{Pair}(0,\mathtt{Suc}(\mathtt{Suc}(q))),$$
$$\mathtt{Pair}(\mathtt{Suc}(0),\mathtt{Suc}(\mathtt{Suc}(q))),$$
$$\mathtt{Pair}(\mathtt{Suc}(\mathtt{Suc}(p)),\mathtt{Suc}(\mathtt{Suc}(q))),$$

$\mathtt{Pair}(\mathtt{Suc}(\mathtt{Suc}(p)),\mathtt{Suc}(\mathtt{Suc}(q))) \in M$ に対しては次のことを示せばよい．

$$E_{\mathtt{ModP}} \cup \{\ e[x \leftarrow \mathtt{Pair}(p,q)],$$
$$e[x \leftarrow \mathtt{Pair}(\mathtt{Suc}(p),q)],$$
$$e[x \leftarrow \mathtt{Pair}(\mathtt{Suc}(\mathtt{Suc}(p)),q)],$$
$$e[x \leftarrow \mathtt{Pair}(p,\mathtt{Suc}(q))],$$
$$e[x \leftarrow \mathtt{Pair}(\mathtt{Suc}(p),\mathtt{Suc}(q))],$$
$$e[x \leftarrow \mathtt{Pair}(\mathtt{Suc}(\mathtt{Suc}(p)),\mathtt{Suc}(q))],$$
$$e[x \leftarrow \mathtt{Pair}(p,\mathtt{Suc}(\mathtt{Suc}(q)))],$$
$$e[x \leftarrow \mathtt{Pair}(\mathtt{Suc}(p),\mathtt{Suc}(\mathtt{Suc}(q)))],\ \}$$
$$\vdash e[x \leftarrow \mathtt{Pair}(\mathtt{Suc}(\mathtt{Suc}(p)),\mathtt{Suc}(\mathtt{Suc}(q)))]$$

$e[x \leftarrow \mathtt{Pair}(\mathtt{Suc}(\mathtt{Suc}(p)),\mathtt{Suc}(\mathtt{Suc}(q)))]$ の左辺に対しては，

$$E_{\mathtt{ModP}} \vdash \mathtt{ModP(ModP(Pair(Suc(Suc}(p)),\mathtt{Suc(Suc}(q)))))$$
$$\approx \mathtt{Pair(Mod(Mod}(p)),\mathtt{Mod(Mod}(q)))$$

同様に右辺に対しては，

$$E_{\mathtt{ModP}} \vdash \mathtt{ModP(Pair(Suc(Suc}(p)),\mathtt{Suc(Suc}(q))))$$
$$\approx \mathtt{Pair(Mod}(p),\mathtt{Mod}(q))$$

である．$e[x \leftarrow \mathtt{Pair}(p,q)]$ は帰納法の仮定の一つである．この式の左辺

$$\mathtt{ModP(ModP(Pair}(p,q)))$$

$$S_{\mathtt{ModP}} = \{\ \mathtt{nat},\ \mathtt{pair}\}$$
$$F_{\mathtt{ModP}} = C_{\mathtt{ModP}} \cup D_{\mathtt{ModP}}$$
$$C_{\mathtt{ModP}} = \{\ \mathtt{0} : \rightarrow \mathtt{nat},$$
$$\mathtt{Suc} : \mathtt{nat} \rightarrow \mathtt{nat},$$
$$\mathtt{Pair} : \mathtt{nat}, \mathtt{nat} \rightarrow \mathtt{pair}\}$$
$$D_{\mathtt{ModP}} = \{\ \mathtt{Mod} : \mathtt{nat} \rightarrow \mathtt{nat},$$
$$\mathtt{ModP} : \mathtt{pair} \rightarrow \mathtt{pair}\}$$
$$E_{\mathtt{ModP}} = \{\ \mathtt{Mod(0)} \approx \mathtt{0},$$
$$\mathtt{Mod(Suc(0))} \approx \mathtt{Suc(0)},$$
$$\mathtt{Mod(Suc(Suc}(x))) \approx \mathtt{Mod}(x),$$
$$\mathtt{ModP(Pair}(x,y)) \approx \mathtt{Pair(Mod}(x),\mathtt{Mod}(y))\}$$

図5: 自然数の対の2の剰余の仕様

と右辺

$$\mathtt{ModP}(\mathtt{Pair}(p,q))$$

は，それぞれ，$E_{\mathtt{ModP}}$ の等式を用いて

$$\mathtt{Pair}(\mathtt{Mod}(\mathtt{Mod}(p)),\mathtt{Mod}(\mathtt{Mod}(q))),$$

および，

$$\mathtt{Pair}(\mathtt{Mod}(p),\mathtt{Mod}(q))$$

と等しいことがわかる．$M_{\mathtt{ModP}}$ の他の要素についても同様に証明できる．

この例の場合，Zhang らの方法では，帰納法の仮定が使えない．それは，$\mathtt{Pair}(\mathtt{Suc}(\mathtt{Suc}(p)),\mathtt{Suc}(\mathtt{Suc}(q)))$ の中にそれと同一のソートである pair をもつ部分項が存在しないためである．彼らの方法で証明するためには，補題

$$E_{\mathtt{ModP}} \vdash_{\mathrm{ind}} \mathtt{ModP}(\mathtt{ModP}(\mathtt{Pair}(y,z))) \approx \mathtt{ModP}(\mathtt{Pair}(y,z))$$

を用意する必要がある．

## 6 まとめ

代数的仕様の帰納的定理の証明に適した帰納法である被覆集合帰納法を提案し，その正当性を証明した．また，Zhang らの方法との比較を行い，証明能力が互いに含まれないことを示した．

仕様を項書換え系とみなしたとき有限停止性を満たすという条件の下で，本方法に基づいて帰納的定理の証明を支援する手続きについては文献 [12] に示している．実際にこの手続きを実現すること，それを用いていろいろな証明を行うこと，また，以上の条件を満たさない場合に対する手続き化については，今後の課題である．

**謝辞** 日頃ご指導下さる豊橋技術科学大学本多波雄学長，中京大学福村晃夫教授，また，御討論下さった中京大学外山勝彦助手，ならびに，平田富夫助教授，直井徹助手，外山勝彦助手をはじめとする関連研究室の皆様に感謝します．

## 参考文献

[1] Burstall R. M., Proving Properties of Programs by Structual Induction, The Computer Jurnal, Vol.12, No.1, pp.41-48(1969).

[2] Boyer R., Moore J.S., A Computational Logic,Academic Press(1979).

[3] Fribourg L., A Strong Restriction of the Inductive Completion Procedure, Proc. of 13th International Colloquium on Automata, Languages and Programming, LNCS226, pp.105-115(1986,7)

[4] Gaudel M. C., Specification of Compilers as Abstract Data Types Representations, Proc. of Workshop on Semantics-Directed Compilers in Aarhus, LNCS94, pp.140-164(1980).

[5] Goguen J. A., How to Prove Algebraic Inductive Hypotheses Without Induction, With Applications to the Correctness of Data Type Implementation, Proc. of the 5th Conference on Automated Deduction, Les Arcs(July, 1980).

[6] Huet G., Hullot J. M., Proofs by Induction in Equational Theories with Constructors, Rapports de Recherch, INRIA, No.28(1980).

[7] Lazrek A., Lescanne P., Thiel J. J., Proving Inductive Equalities, Algorithms and Implementation, Tech. Rep., NANCY, 86-R-087(1987).

[8] Musser D. L., On Proving Inductive Properties of Abstract Data Types, Proc. of the 7th Annual ACM Sympo. on Principles of Programming Languages, Las Vegas, pp.154-162(Jan, 1980).

[9] Paul E., Proof by Induction in Equational Theories with Relations between Constructors, Proc. 9th Colloquium on Trees in Algebra and Programming, Bordeaux, Cambridge U Press, pp.211-225(1984,3).

[10] Puel L., Proofs in the Final Algebra, Proc. 9th Colloquium on Trees in Algebra and Programming, Bordeaux, Cambridge U Press, pp.227-242 (1984,3).

[11] Zhang H., Kapur K., Krishnamoorthy M. S., A Mechanizable Induction Principle for Equational Specification, Proc. of 9th International Conf. on Automated Deduction at Argonne, Illinois, USA, LNCS 310, pp.162-181(1988,5).

[12] 酒井，坂部，稲垣，代数的仕様における帰納的性質の証明法，電子通信学会，技術報告 COMP88-86, pp.83-92(1989,1).

[13] 酒井，坂部，稲垣，代数的仕様の帰納的性質の証明における場合分けの制限について，電気関係学会東海支部連合大会, 531(1989).

[14] 稲垣，坂部，抽象データタイプの代数的仕様記述の基礎(1)-(4), 情報処理，**25**, 1,5,7,9, (1984)