# 被覆集合帰納法と代数的仕様の検証の機械的支援に関する考察

1M−3

酒井正彦　坂部俊樹　稲垣康善
名古屋大学工学部

## 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．

代数的仕様の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 [1]．この方法の問題点のひとつは，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ことである．このことは，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．このため，伝統的な帰納法の証明法も見直されてきており，構造帰納法の拡張である被覆集合帰納法が提案されている [2, 3]．

本論文では，文献 [3] で提案した被覆集合帰納法に基づく証明例を挙げ，その機械的支援に関する考察を行なう．

## 2 準備

$\Sigma = \langle S, F \rangle$ をシグニチャ，$V$ を変数の集合とする．項は $F$ の関数記号と $V$ の変数から構成される．変数を含まない項を基礎項と呼ぶ．

代数的仕様 $\mathcal{A}$ は $\langle \Sigma, V, E \rangle$ である．ここで，$E$ は同一ソートの項 $\xi, \eta$ からなる等式 $\xi \approx \eta$ の集合である．$E$ を項書換え系とみなしたとき，どの基礎項の正規形にも現れない関数記号の集合を $D$ とし，それ以外の $F$ の関数記号の集合を $C$ とする．変数を含まない項を基礎項，$F$ の関数記号を含まない項を構成子項と呼ぶ．$\xi, \eta$ を項とするとき，$\xi \approx \eta$ が $E$ から推論できるとき，$E \vdash \xi \approx \eta$ と書く．また，任意の基礎項の代入 $\theta$ に対して $\theta(\xi) \approx \theta(\eta)$ が $E$ から推論できるとき，等式 $\xi \approx \eta$ を $E$ の帰納的定理といい，$E \vdash_{\mathrm{ind}} \xi \approx \eta$ と書く．

## 3 被覆集合帰納法

まず，被覆集合帰納法を簡潔に述べる．詳しくは文献 [3] を参照されたい．任意の基礎項を表現するためにメタ変数 $p, q, r, \cdots$ を導入し，その集合を $MV$ とする．$(C \cup MV)$ 項の集合で，任意の構成子基礎項 $\xi$ に対してメタ変数への適当な項の代入によって $\xi$ に等しくなる要素が存在するとき，これを被覆集合と呼ぶ．$(F \cup MV)$ 項を引数とする二つの関数 SubT,SupT を定義する．(1) SubT($\xi$) は $\xi$ もしくは $\xi$ の部分項の集合を返す．(2) SupT($\xi$) は，$\xi$ 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．

関係 $\xi \succeq_{\mathrm{SuT}} \eta$ を，$\eta \in \mathrm{SupT}(\mathrm{SubT}(\xi))$ として定義すると，一変数の被覆集合帰納法は次のように定式化される．

$e$ 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．ソート $s$ の被覆集合 $M$ の任意の要素 $\xi$ に対して，

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

が成り立つならば，

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

## 4 証明例

図1の仕様に対して，帰納的定理 $P(x) : E \vdash_{\mathrm{ind}} \mathtt{Mod}(\mathtt{Suc}(\mathtt{Suc}(0)),x) \approx \mathtt{Mod}(\mathtt{Suc}(\mathtt{Suc}(0)), \mathtt{Mod}(\mathtt{Suc}(\mathtt{Suc}(0)),x))$ を証明する．

まず，文献 [4] に従って被覆集合を求めると，$M = \{0, \mathtt{Suc}(0), \mathtt{Suc}(\mathtt{Suc}(q))\}$ となる．$E \vdash_{\mathrm{ind}} P(x)$ を証明するためには (1) $E \vdash P(0)$, (2) $E \vdash P(\mathtt{Suc}(0))$, (3) $E \cup \{P(q), P(\mathtt{Suc}(q))\} \vdash P(\mathtt{Suc}(\mathtt{Suc}(q)))$ の三つを示す必要がある．$P(q)$, $P(\mathtt{Suc}(q))$ は帰納法の仮定と呼ばれる．

(1)〜(3) の証明は，$E$ を項書換え系とみなし

---

Cover Set Induction Principle and a Study about Mechanical Support for Verifying Algebraic Specification
Masahiko SAKAI, Toshiki SAKABE, Yasuyoshi INAGAKI
Nagoya University

て≈の両辺を書き換えることによって行なう．(1)の場合両辺は0になり，(2)の場合両辺はSuc(0)になるので，これらは示された．(3)の場合，左辺はNSg(NSg(Mod(Suc(Suc(0)), $q$)))，右辺はMod(Suc(Suc(0)), NSg(NSg(Mod(Suc(Suc(0)), $q$))))となる．左辺をさらに$P(q)$と$P$(Suc($q$))で書き換えると，NSg(NSg(Mod(Suc(Suc(0)), Mod(Suc(Suc(0)),$q$))))が得られる．このようにして得られた両辺を一般化して，$E \vdash_{\text{ind}}$ NSg(NSg(Mod(Suc(Suc(0)), $x$))) ≈ Mod(Suc(Suc(0)), NSg(NSg($x$)))という補題$Q(x)$を作成する．この補題が成り立てば，証明は完了する．

これと同様に，補題$Q(x)$についても証明を進めると，NSg(NSg(Mod(Suc(Suc(0)), NSg(NSg($q$)))))とSuc(0)が等しくなることが必要になる．そこで，新たにNSg(NSg(Mod(Suc(Suc(0)), NSg(NSg($x$)))))≈Suc(0)という補題$R(x)$を作成してこの証明を行なう．$R(x)$については補題の作成なしに証明される．

## 5 機械的支援に関する検討

被覆集合帰納法を用いた代数的仕様の検証の機械的支援系の実現には，いくつかの問題点がある．以下では，これらについて議論する．

先の例において，(1)〜(3)の証明は，基本的には，$E$を項書換え系とみなして行なので，無限の書換えが行なわれる可能性がある．これに対しては，直接実行可能な代数的仕様を検証の対象に考えているので，有限停止性を仕様に対する条件としても大きな問題は起こらない．

次に，帰納法の仮定の適用は，示すべき等式の左辺を書き換える場合には仮定の等式を右向きに使い，右辺を書き換える場合には仮定の等式に使うことにする．先の例の(3)では，$P$(Suc(Suc($q$)))の左辺を書き換える際には，$P(q)$や$P$(Suc($q$))を右向きの書き換え規則として用いる．また，反対に，$P$(Suc(Suc($q$)))の右辺を書き換える際には，$P(q)$や$P$(Suc($q$))を左向きの書き換え規則として用いる．これは，経験上この方向にしか仮定が用いられないことから定めた．その原因は，帰納法の仮定の形が示すべき等式の形に深く関わっているからである．この場合，無限の書換えを引き起こすことがある．例えば，先の例の(3)では，$P$(Suc(Suc($q$)))の左辺を書き換えて得られたNSg(NSg(Mod(Suc(Suc(0)), $q$)))に対して$P(q)$を右向きに一回だけ書き換えていた．しかし，この適用を機械的に行なうと，無限の書換えを引き起こす．したがって，帰納法の仮定による書き換えをするべきかどうかは，ユーザへの問い合わせが必要である．

補題の発見には，仕様からボトムアップ的に作る方法と，証明の過程でトップダウン的に作る方法がある．先の証明例では，後者の方法で補題の発見を行なった．この方法は，前者の方法よりも機械的な支援がしやすいと思われるが，例でも行なわれたように，一般化の技術が重要になる．一方，前者の方法は，人手で証明を行なう際に用いられる手法である．先の例において，人が$E \vdash_{\text{ind}}$ NSg(NSg($x$)) ≈ Sg($x$)と$E \vdash_{\text{ind}}$ Mod(Suc(Suc(0)), Sg($x$)) ≈ Sg($x$)という二つの補題を発見すれば，これらの二つの補題，ならびに，(3)の証明はいづれも機械的に行なうことができる．

## 6 まとめ

被覆集合帰納法による証明の例を挙げ，検証支援系の実現のための検討を行なった．今後，支援系の作成，評価により，効果的な証明支援方法を明らかにすることは今後の課題である．

## 参考文献

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

[2]Zhang H., Kapur K., Krishnamoorthy M. S., A Mechanizable Induction Principle for Equational Specification, LNCS 310, pp.162-181(1988,5).

[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，電子情報通信学会，技術報告COMP90-5, pp37-46(1990,5)

[4]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(1989,10).

```
S={nat},  C={0,Suc},
D={Add,Sg,NSg,Cycle,Mod}
E={Add(0,x) ≈ x;
   Add(S(x),y) ≈ S(Add(x,y));
   Sg(0) ≈ 0;
   Sg(S(x)) ≈ S(0);
   NSg(0) ≈ S(0);
   NSg(S(x)) ≈ 0;
   Cycle(0,x) ≈ S(0);
   Cycle(S(k),x) ≈
     Add(Add(Mod(k,x),Sg(Mod(k,x))),NSg(x));
   Mod(0,x) ≈ 0;
   Mod(S(0),x) ≈ 0;
   Mod(S(S(k)),0) ≈ 0;
   Mod(S(S(k)),S(x)) ≈
     Cycle(S(S(k)),Mod(S(S(k)),x)); }
```

図1 剰余の仕様