Title	潜在帰納法と書換え帰納法の比較
Author(s)	小池，広高
Citation	
Issue Date	1999-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1240
Rights	
Description	Supervisor:外山 芳人，情報科学研究科，修士

修 士 論 文

潜在帰納法と書換え帰納法の比較

指導教官　外山芳人 教授

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

小池広高

1999 年 2 月 15 日

目 次

第 1 章

はじめに

1.1 研究の背景・目的

関数型言語、代数的仕様記述法など等式に論理的基礎をおく言語系の性質は多くの場合、等式論理の帰納的定理として取り扱うことができる。それゆえに等式論理における帰納的定理の自動証明法は、代数的仕様やプログラムの検証、また仕様とプログラムの等価性判定などを自動的に行なうためには不可欠である [1]。

等式論理における帰納的定理の自動証明手法は、明示帰納法 (explicit induction) と暗黙帰納法 (implicit induction) に大きくわけられる。明示帰納法とは帰納的図式を直接もちいて帰納的定理の証明を行なう手法である。Boyer と Moore によって研究された Nqthm[7] は、この手法による自動証明システムとして有名である。一方、暗黙帰納法とは、帰納的図式を直接もちいずに帰納的定理の証明を行なう手法で、Musser[21] により提案され Huet と Hullot[10] により拡張された潜在帰納法 (inductionless induction) や、Reddy[23] らによって提案された書換え帰納法 (rewriting induction) がある。また、潜在帰納法にもとづく自動証明システムとしては RRL[16]、書換え帰納法にもとづく自動証明システムとしては SPIKE[4, 5] が知られている。

明示帰納法による証明と暗黙帰納法による証明を簡単な例をもちいて説明する。自然数の集合 $\mathcal{N}$ 上の加算を次の等号論理 E によって定める。

$$E : \begin{cases} x + 0 = x \\ x + s(y) = s(x + y) \end{cases}$$

ここで自然数 0,1,2, ⋯ は 0,s(0),s(s(0)), ⋯ で表されているものとする。このとき任意の

自然数 n に対して、次の定理が成立することを証明する。

$$\forall n \in \mathcal{N}\ [0+n=n]$$

このような等式は、任意の項の集合上で成立する定理ではないが、基底項上では成立する定理である。基底項上の定理を証明するためには通常は適当なデータ構造上の帰納法が必要とされるので、このような定理は帰納的定理と呼ばれている。

明示帰納法では次の帰納的図式をもちいて証明を行なう。

$$P(0) \wedge \forall n \in \mathcal{N}[P(n) \Rightarrow P(s(n))] \Rightarrow \forall n \in \mathcal{N}.P(n)$$

実際に上記の図式をもちいて $P(n) := [0+n=n]$ の証明を行なう。
基礎段階: $0+0=0$
帰納段階: $0+n=n \Rightarrow 0+s(n)=s(n)$
上記のように各段階で成立することは明らか。よって帰納的図式より $\forall n \in \mathcal{N}.P(n)$ が得られ、与えられた等式は帰納的定理であることが示される。

一方、暗黙帰納法では公理 E に証明すべき等式を付け加えて等号論理 E ’$= E \cup \{0+n=n\}$ をつくる。つぎに、E と E’ によって定められる基底項の集合上の同値関係が等しいことを示す。すると、E’ のもとで成立している基底項上の等式は E のもとでも成立することになる。ここで、$0+n=n$ は E’ のもとで成立している基底項上の等式であるから、E のもとでも成立していることになる。つまり、帰納的図式を用いずに帰納的定理 $\forall n \in \mathcal{N}\ [0+n=n]$ が証明されたことになる。

本研究の目的は、暗黙帰納法による自動証明手法として重要な潜在帰納法と書換え帰納法を理論的に比較することである。ここでは、潜在帰納法と書換え帰納法を抽象的に形式化することで理論的に整理し、両者の本質的な差異が、合流性と退行性、および弱正規性と強正規性の差異にもとづくことを明らかにする。また、実際の定理証明システム [2, 4, 5, 8, 9, 10, 11, 12, 15, 21] で広くもちいられている反駁証明手法についても考察し、反駁証明手法を組み合わせると、潜在帰納法と書換え帰納法の能力が実質的には一致することを明らかにする。これまでは、このような観点から両者を比較した研究はなされておらず、ここで示された結果は今後両者を融合した新しい自動証明手法を確立する上で極めて有用であると考えられる。

1.2 構成

本論文の構成は次のとおりである。2 章では必要となる定義を与える。3 章では潜在帰納法、書換え帰納法と反駁証明法を抽象的な枠組で整理する。4 章では潜在帰納法と書換え帰納法の証明能力の比較を行なう。5 章では帰納的定理の自動証明システムに関する考察を行なう。

第 2 章

準備

本章では、抽象リダクションシステムと項書換えシステムに関する用語や概念を文献 [1] にもとづき定義する。

2.1 抽象リダクションシステム

抽象リダクションシステム $R = \langle A, \rightarrow \rangle$ は、対象集合 A と A 上の二項関係$\rightarrow$の対で定められる。$\rightarrow$をリダクション関係と呼ぶ。Rのリダクション系列とは$x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow \cdots$ のことである。A 上の同一関係を$\equiv$で表す。$\xrightarrow{*}$、$\xrightarrow{+}$をそれぞれ$\rightarrow$の反射推移閉包、推移閉包、$=$を$\rightarrow$から生成される同値関係とする。$x \rightarrow y$なる $y \in A$ が存在しないとき、$x \in A$ は正規形であるという。$x \xrightarrow{*} y$となる正規形 $y \in A$ が存在するとき、$x \in A$ は正規形 y をもつという。NFで正規形の集合を表す。

定義 2.1 $R = \langle A, \rightarrow \rangle$ におけるすべてのリダクション系列が有限であるとき、R は強正規性 *(Strongly Normalizing)* をもつといい、$\rightarrow$: SNと記す。すべての $a \in A$ が正規形をもつとき、R は弱正規性 *(Weakly Normalizing)* をもつといい$\rightarrow$: WNと記す。

定義 2.2　$R = \langle A, \rightarrow \rangle$ は次の条件を満たすとき、合流性 *(Church Rosser)* をもつといい $\rightarrow$: CR と記す。
$\forall x, y, z[x \xrightarrow{*} y \wedge x \xrightarrow{*} z \Rightarrow \exists w.\ y \xrightarrow{*} w \wedge z \xrightarrow{*} w]$　*(図 2.1)*

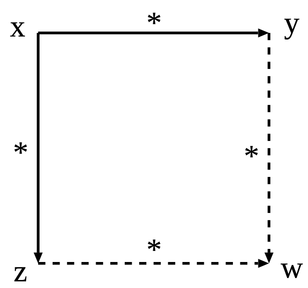

図 2.1: 合流性

定義 2.3　$R_2 = \langle A, \rightarrow_2 \rangle$ は次の条件を満たすとき、$R_1 = \langle A, \rightarrow_1 \rangle$ に退行 *(Retrogresse)* するといい $\rightarrow_2$ RET $\rightarrow_1$ と記す。
$\forall x, y[x \rightarrow_2 y \Rightarrow \exists z, w.\ x \rightarrow_1 z \wedge z \xrightarrow{*}_2 w \wedge y \xrightarrow{*}_2 w]$　*(図 2.2)*

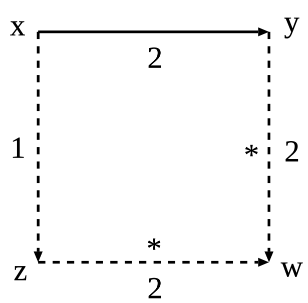

図 2.2: 退行性

命題 2.4　R が合流性をもつとき以下のことが成り立つ *[1]*。
(1)$\forall x, y \in A[x = y \Rightarrow \exists w \in A, x \xrightarrow{*} w \wedge y \xrightarrow{*} w]$、
(2)$\forall x, y \in NF[x = y \Rightarrow x \equiv y]$。

命題 2.5 *(ネータ帰納法)* $\rightarrow$が強正規性をもつとき$\xrightarrow{+}$は A 上の整礎な関係になるので、以下の性質が成り立つ [*1*]。

$$\forall x \in A[\forall y \in A[x \xrightarrow{+} y \Rightarrow P(y)] \Rightarrow P(x)] \ \Rightarrow \ \forall x \in A.P(x)$$

2.2 項書換えシステム

関数記号 f、g、h、$\cdots$の集合を F、変数記号 x、y、z、$\cdots$ の集合を Vとする。ここで、$F \cap V = \emptyset$ とする。

定義 2.6 項は写像 $arity$:$F \rightarrow \mathcal{N}$を用いて、次のように再帰的に定義される。
1. $x \in V$ は項である。
2. $f \in F$,$arity(f) = n$で、$t_1, \cdots, t_n$が項のとき、$f(t_1, \cdots, t_n)$ も項である。

FとVから生成される項の集合を$T(F,V)$ と記す。変数を含まない項を基底項と呼びその集合を$T(F)$ と記す。ここで、Fは少なくとも 1 つ $arity(f) = 0$ となる関数記号を含むとする。つまり、基底項の集合 $T(F)$ は空集合でないと仮定する。

代入θは変数から項への写像であり、$\theta(f(t_1, \cdots, t_n)) \equiv f(\theta(t_1, \cdots, t_n))$ のように項から項への写像に拡張される。項 t に代入θを適用した結果を $t\theta$で表す。特に $t\theta \in T(F)$ となるとき、代入θを基底代入と呼ぶ。以下ではθ_gで基底代入を表す。基底項の正規形全体からなる集合を NF^Gで表す。文脈 Cとは、$T(F \cup \{\square\}, V)$ の要素のことである (ただし $\square$ は Fに含まれない定数とする)。n 個の $\square$ の出現する文脈 Cを $C[, \cdots,]$ で表し、$C[, \cdots,]$ に出現する $\square$ を左から順に $t_1, \cdots, t_n$で置き換えて得られる項を $C[t_1, \cdots, t_n]$ で表す。特に $C[\ ]$ は $\square$ の出現が一つの文脈を表す。

定義 2.7 正整数の列の集合 N_+^*を用いて項に出現する関数記号の出現位置を以下のように定める。

1. 根記号 (最外の関数記号) の出現位置をε(空列を意味する) とする。
2. $C[f(t_1, \cdots, t_n)]$ における fの出現位置が u のとき各 t_iの根記号の出現位置を $u \cdot i$ とする。

項 t の部分項で出現位置 u を根とする部分項を $t|_u$で記す。

定義 2.8 規則は以下の変数条件を満たす項の対 (l, r) である。
1. lは変数でない。
2. rに出現する変数は必ず lに出現する。

以下では規則 (l, r) を $l \to r$で表す。項書換えシステム R は規則の有限集合によって定義される。

定義 2.9 項書換えシステム R における *2* 項関係$\to_R$を以下のように定義する。

$$t \to_R s \overset{\text{def}}{\Leftrightarrow} \exists\, l \to r \in R, \exists C[\ ],\ \exists \theta, t \equiv C[l\theta] \text{ かつ } s \equiv C[r\theta]$$

また、項 t の部分項 $l\theta$をリデックスと呼ぶ。$\to_R$の反射・推移・対象閉包を$\to_R$によって生成される合同関係と呼び$=_R$と書く。

例 2.10 項書換えシステム R を次のように定義する。

$$R : \begin{cases} x + 0 \to x \\ x + s(y) \to s(x + y) \end{cases}$$

$0, 1, 2, \cdots$は $0, s(0), s(s(0)), \cdots$ で表現されていることにすると、R は自然数上の加算に対応する項書換えシステムである。このとき、$s(0) + s(s(0))$ は、
$s(0) + s(s(0)) \to_R s(s(0) + s(0)) \to_R s(s(s(0) + 0)) \to_R s(s(s(0)))$
のように書き換えられる。この書換え列は、項 $s(0) + s(s(0))$ を評価した結果が正規形 $s(s(s(0)))$ となることを表し、$1 + 2$ を計算した結果が 3 であることを意味する。

定義 2.11 $l_1 \to r_1$と $l_2 \to r_2$を R の書換え規則とする。ここで、一般性を失うことなく *2* つの規則は変数を共有しないものと仮定する。ある文脈 $C[\ ]$ で $l_1 \equiv C[s]$ かつ $s \notin V$で、最汎単一化子θが存在して、$s\theta \equiv l_2\theta$であれば、*2* つの書換え規則 $l_1 \to r_1$と $l_2 \to r_2$は重なるという。このとき、$\langle C[r_1\theta], r_2\theta\rangle$ を危険対という。ただし、同じ書換え規則同士の重なりについて考えるときは、$C \not\equiv \Box$ とする。

例 2.12 書換え規則 $f(g(x)) \to a$ と $g(h(y)) \to y$は、文脈 $C[\ ] \equiv f(\Box)$ と代入$\theta = [x \leftarrow h(z), y \leftarrow z]$ を考えると明らかなように、危険対 $\langle C[y\theta], a\theta\rangle$ すなわち $\langle f(z), a\rangle$ をもつ。

定義 2.13 等式 $e = e'$が項書換えシステム R における帰納的定理であるとは、任意の基底代入θ_gについて $e\theta_g =_R e'\theta_g$となることである。

例 2.14 次の項書換えシステム R を考える。

$$R : \begin{cases} x + 0 \to x \\ x + s(y) \to s(x + y) \end{cases}$$

このとき、$0 + x =_R x$ は成立しない。しかし、$\forall\theta_g\ [(0 + x)\theta_g =_R x\theta_g]$ は成立するため $0 + x = x$ は項書換えシステム R における帰納的定理となる。

2.3　被覆集合

本節では、2 つの項書換えシステムにおける基底項の正規形集合の等価性を示す手段の一つである、被覆集合について説明する。

定義 2.15　項の有限集合 $\{t_i\}_i$が項書換えシステム R の被覆集合とは以下が成り立つことである。

$$\forall g \in T(F)\ \exists t_i\ \exists \theta_g\ [g \xrightarrow{*}_R t_i\theta_g]$$

また、被覆代入集合 $\{\sigma_i\}_i$とは、値域が被覆集合の元となる代入の集合である。

例 2.16　項書換えシステム R は弱正規性をもつとする。次の性質を満たす正規形の有限集合 $\{t'_i\}_i$ は必ず R の被復集合となる。

$$\forall g' \in NF_R^G \exists t'_i \exists \theta_g [g' \equiv t'_i\theta_g]$$

例 2.17　次の項書換えシステム R_{add}を考える。

$$R_{add} : \begin{cases} x + 0 \to x \\ x + s(y) \to s(x + y) \end{cases}$$

このとき、$\{z\}, \{0, s(z)\}, \{0, s(0), s(s(z))\}, \{0, s(z), z\}, \{z, x + y\}$ などは R_{add}の被覆集合となる。□

補題 2.18　項書換えシステム R_1の被覆代入集合 $\{\sigma_i\}_i$と $R_2 = R_1 \cup \{e \to e'\}$ を考える。また $H = \{e \to e'\}$ とする。このとき以下に示すことが成立する。

$$\forall \sigma_i . e\sigma_i \notin NF_{R_1} \Rightarrow NF_{R_1}^G = NF_{R_2}^G$$

証明　$NF_{R_2}^G \subseteq NF_{R_1}^G$は自明。$NF_{R_1}^G \subseteq NF_{R_2}^G$を示すために、$\forall s, t \in T(F) \exists u \in T(F) [s \to_{R_2} t \Rightarrow s \to_{R_1} u]$ が成立することを示す。$s \to_{R_1} t$ の場合は自明。$s \equiv C[e\theta_g] \to_H C[e'\theta_g] \equiv t$ の場合を考える。このとき、定義 2.15と $\forall \sigma_i . e\sigma_i \notin NF_{R_1}$より、$s \xrightarrow{*}_{R_1} C[e\sigma_i\theta'_g] \to_{R_1} u$ が成立する。□

例 2.19 項書換えシステム R_{add}と $R_2 = R_{add} \cup \{0 + x \to x\}$ を考える。このとき、被覆代入集合 $\{\sigma_i\}_i = \{\{x \leftarrow 0\}, \{x \leftarrow s(z)\}\}$ に対して、$\forall \sigma_i . (0 + x)\sigma_i \notin NF_{R_{add}}$は明らか。よって、$NF_{R_{add}}^G = NF_{R_2}^G$が示される。□

被覆集合は有限集合なので、例 2.19のように 2 つの項書換えシステム R_1と $R_2 = R_1 \cup \{e \to e'\}$ において、$NF^G_{R_1} = NF^G_{R_2}$の判定は容易に実行可能となる。また、特に補題 2.18 の $(\Leftarrow)$ の性質を合わせもつ代入集合、つまり項書換えシステム R_1と $R_2 = R_1 \cup \{e \to e'\}$ において、

$$\forall \sigma_i . e\sigma_i \notin NF_{R_1} \Leftrightarrow NF^G_{R_1} = NF^G_{R_2}$$

を満たす有限代入集合 $\{\sigma_i\}_i$は、テスト代入集合と呼ばれる[1] 。

被覆代入集合やテスト代入集合に関しては、様々な研究が行なわれている。詳細は、文献 [2, 3, 4, 5, 6, 8, 10, 11, 17, 15, 23, 24, 25, 26] 参照。

[1]例 2.17において、集合 $\{0, s(z)\}$、$\{0, s(0), s(s(z))\}$ の元を値域とする代入集合 $\{\sigma_i\}_i$は、R_{add}のテスト代入集合となる。

第 3 章

潜在帰納法と書換え帰納法

本章では適当な集合上で 2 つの抽象リダクションシステムの等価性を判定する手法を示す。また、項書換えシステムにおいて基底項全体からなる集合に着目することにより、等価性判定手法を帰納的定理の証明に応用する方法を説明する。

抽象リダクションシステム $R_1 = \langle A, \to_1 \rangle$ と $R_2 = \langle A, \to_2 \rangle$ を考える。それぞれのシステムによって定められる同値関係を$=_i$、正規形の集合を $NF_i (i = 1, 2)$ で表す。y が x の正規形のとき y を $x \downarrow$で表す。$\to_1 \subseteq \to_2$は $\forall x, y \in A[x \to_1 y \Rightarrow x \to_2 y]$ を意味する。A の部分集合 A' 上で R_1と R_2が等しい同値関係をもつとは $\forall x, y \in A'[x =_1 y \Leftrightarrow x =_2 y]$ が成立することである。これを$=_1 \ = \ =_2 \ in \ A'$と記す。

3.1 潜在帰納法

潜在帰納法は異なる 2 つのシステムの等価性を判定するために導入された手法である [10, 11, 12, 13, 15, 21, 25]。外山は潜在帰納法を、以下のような抽象リダクションシステムの枠組で定式化した。

命題 3.1 [*25*] 次の条件が成立するものと仮定する。

$$(1)\ \to_1 \subseteq \to_2 \quad (2)\ \to_1: WN \quad (3)\ \to_2: CR \quad (4)\ NF_1 = NF_2$$

このとき、$=_1 \ = \ =_2$ が成立する。

証明 $=_1 \subseteq =_2$は (1) より明らか。$=_2 \subseteq =_1$を示す。$x =_2 y$とすると、(2) より $\exists z \in NF_1[x \xrightarrow{*}_1 z]$ かつ $\exists w \in NF_1[y \xrightarrow{*}_1 w]$。(1) より $z =_2 w$となる。(4) より $z, w \in NF_2$となるので (3) と命題 2.4(2) より $z \equiv w$が成立。したがって $x =_1 y$。□

通常は適当なデータ構造上の帰納法が必要とされる帰納的定理の証明を、命題3.1をもちいると直接的な帰納法の適用なしで証明することが可能となる。以下では、命題3.1にもとづいて潜在帰納法の原理を説明する。

潜在帰納法の原理

項書換えシステム R_1のもとで等式 $e = e'$の証明を行なう。$R_2 = R_1 \cup \{e \to e'\}$ とおく。つまり、R_1の書換え規則に $e \to e'$を付け加えて R_2とする。このとき R_1、R_2が基底項上で

$$(1) \to_{R_1} \subseteq \to_{R_2} \quad (2) \to_{R_1}: WN \quad (3) \to_{R_2}: CR \quad (4) NF_{R_1} = NF_{R_2}$$

を満たすことを示す。これが成功すると命題3.1より$=_{R_1} \, = \, =_{R_2}$ $in\ T(F)$ が成立する。ここで R_2は書換え規則として $e \to e'$を含んでいる。よって明らかに、$\forall\theta_g.\ e\theta_g =_{R_2} e'\theta_g$である。したがって、$\forall\theta_g.\ e\theta_g =_{R_1} e'\theta_g$が得られ、$e = e'$が R_1における帰納的定理であることが証明される。

例 3.2　次の項書換えシステム R_1を考える。

$$R_1 : \begin{cases} x + 0 \to x \\ x + s(y) \to s(x + y) \end{cases}$$

$0 + z = z$が R_1における帰納的定理であることを示す。$R_2 = R_1 \cup \{0 + z \to z\}$ とおく。このとき、明らかに $\forall g \in T(F) \exists n \in \mathcal{N}[g \xrightarrow{*}_{R_1} s^n(0)]$ が成立する。よって R_1は弱正規性を満たす。危険対を調べることにより、R_2が合流性をもつことは容易に示すことができる[*1*]。また、被覆集合を利用すると、$NF^G_{R_1} = NF^G_{R_2}$が成立することが示せる(例2.19)。R_1と R_2は基底項上で、命題 *3.1*の条件 *(1)(2)(3)(4)* を満たすので$=_{R_1} \, = \, =_{R_2}$ *in T(F)*。よって $0 + z = z$は R_1の帰納的定理である。□

例3.2の場合は R_2が命題3.1 の条件を満たすので、直接帰納的定理であることを証明できた。しかし、$R_2 = R_1 \cup \{e \to e'\}$ が命題3.1の条件を満たしてないなら、R_2のかわりに基底項上で、$=_{R2} \, = \, =_{R3}$ かつ命題3.1の条件を満たす R_3を見つけても良い。このとき、$\forall\theta_g.\ e\theta_g =_{R2} e'\theta_g \Leftrightarrow \forall\theta_g.\ e\theta_g =_{R3} e'\theta_g \Leftrightarrow \forall\theta_g.\ e\theta_g =_{R1} e'\theta_g$ が成立することから $e = e'$が R_1の帰納的定理であることが示される。このような R_3を発見する手法の一つとしてKunuth-Bendix の完備化手続き [20] が考えられる。

例 3.3　次の項書換えシステム R_1を考える。

$$R_1 : \begin{cases} append(Nill, x) \to x \\ append(cons(x, y), z) \to cons(x, append(y, z)) \\ rev(Nill) \to Nill \\ rev(cons(x, y)) \to append(rev(y), cons(x, Nill)) \end{cases}$$

$rev(rev(x)) = x$ が R_1における帰納的定理であること示す。$R_2 = R_1 \cup \{rev(rev(x)) \to x\}$ を *Kunuth-Bendix* の完備化手続きによって完備化すると以下の R_3が得られる。

$$R_3 : \begin{cases} append(Nill, x) \to x \\ append(cons(x, y), z) \to cons(x, append(y, z)) \\ rev(Nill) \to Nill \\ rev(cons(x, y)) \to append(rev(y), cons(x, Nill)) \\ rev(rev(x)) \to x \\ rev(append(x, cons(y, Nill))) \to cons(y, rev(x)) \end{cases}$$

R_3は基底項上で、$=_{R_2} = =_{R_3}$かつ命題 *3.1*の *(1)(2)(3)(4)* を満たす。よって $rev(rev(x)) = x$ は R_1の帰納的定理である。□

3.2 書換え帰納法

書換え帰納法は Reddy により提案された帰納的定理の証明を行なう手法である [23]。潜在帰納法と書換え帰納法の証明能力を比較するため、3.1 節と同様に抽象リダクションシステムの枠組で書換え帰納法を考察する。

補題 3.4 次の条件が成立するものと仮定する。

$$(1) \ \to_1 \subseteq \to_2 \quad (2) \ \to_2: SN \quad (3) \ \to_2 RET \to_1$$

このとき、$=_1 = =_2$ が成立する。

証明 $=_1 \subseteq =_2$は (1) より明らか。$=_2 \subseteq =_1$を示すために、$\to_2$上のネータ帰納法で $x \xrightarrow{*}_2 y \Rightarrow x =_1 y$を示す。ここで (2) より$\to_2$は強正規性であることに注意する。$x \in NF_2$のときは$x \equiv y$となり成立。$x \to_2 z \xrightarrow{*}_2 y$について、(3) より $\exists x', z'[x \to_1 x' \wedge x' \xrightarrow{*}_2 z' \wedge z \xrightarrow{*}_2 z']$。(1) より $x \to_2 x'$。帰納法の仮定より $x' =_1 z'$、$z =_1 z'$、$z =_1 y$ (図 3.1)。よって $x =_1 y$。□

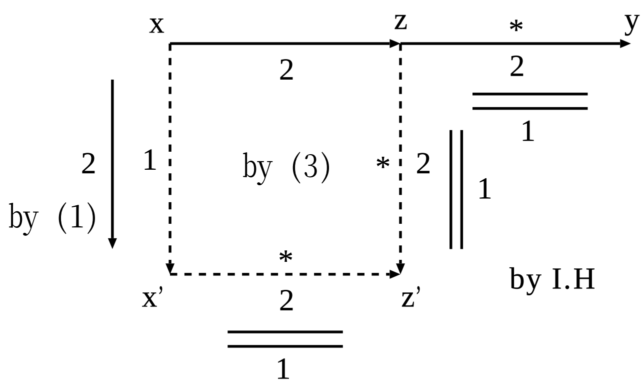

図 3.1: 補題 9 の証明

この補題で明らかなように、書換え帰納法で重要となるのは退行性と強正規性である。実際に、書換え帰納法をもちいて帰納的定理を証明するときには被覆集合を利用することが多い [4, 5, 19, 23]。以下では補題 3.4にもとづき書換え帰納法の原理を説明する。

書換え帰納法の原理

項書換えシステム Rのもとで等式 $e = e'$の証明を行なう。$H = \{e \to e'\}$ とおく。このとき R、$R \cup H$が基底項上で

$$(1)\ \to_R \subseteq \to_{R\cup H} \quad (2)\ \to_{R\cup H}: SN \quad (3)\ \to_{R\cup H}\ RET \to_R$$

を満たすことを示す。これが成功すると補題 3.4より$=_R\ =\ =_{R\cup H}\ in\ T(F)$ が成立する。ここで $R \cup H$は書換え規則として $e \to e'$を含んでいる。よって明らかに、$\forall \theta_g.\ e\theta_g =_{R\cup H} e'\theta_g$ である。したがって、$\forall \theta_g.\ e\theta_g =_R e'\theta_g$が得られ、$e = e'$が Rにおける帰納的定理であることが証明される。

このように書換え帰納法では基底項上で退行性が成立することを示す必要がある。しかし、これを直接示すのは困難である。基底項上で退行性が成立することを保証する条件として、e への被覆代入集合 $\{\sigma_i\}_i$をもちいて次の条件 $(3')$ を考える。

$$(3')\quad \forall \sigma_i \exists a \exists b.\ e\sigma_i \to_R a \to^*_{R\cup H} b \leftarrow^*_{R\cup H} e'\sigma_i$$

被覆集合は有限集合なので、$(3')$ は$\to_{R\cup H}: SN$ の場合は判定することが可能である。

補題 3.5 $\{\sigma_i\}_i$を項書換えシステム Rの被覆代入集合とし、$H = \{e \to e'\}$ とする。このとき $(3')$ が成立するならば、基底項上で $(3) \to_{R\cup H}\ RET \to_R$が成立する。

証明 次の条件 $(3'')$ を考える。

$$
\begin{aligned}
&(3'')\ \forall x, y[x \rightarrow_{R\cup H} y \Rightarrow \\
&\exists x', y', z, w.\ x \xrightarrow{*}_R x' \wedge y \xrightarrow{*}_R y' \wedge x' \rightarrow_R z \wedge y' \rightarrow^{*}_{R\cup H} w \wedge z \rightarrow^{*}_{R\cup H} w]
\end{aligned}
$$

$(3'') \Rightarrow (3)$ は明らか。よって $(3'')$ が基底項上で成立することを示す。$\forall s, t \in T(F)[s \rightarrow_{R\cup H} t]$ とする。このとき $\exists s', t', u, v \in T(F)[s \xrightarrow{*}_R s' \wedge t \xrightarrow{*}_R t' \wedge s' \rightarrow_R u \wedge t' \rightarrow^{*}_{R\cup H} v \wedge u \rightarrow^{*}_{R\cup H} v]$ が成立することを示す。$s \rightarrow_R t$ の場合は自明。$s \equiv C[e\theta_g] \rightarrow_H C[e'\theta_g] \equiv t$ の場合を考える。定義 2.15と $(3')$ より

$$
s \xrightarrow{*}_R C[e\sigma_i\theta'_g] \equiv s' \rightarrow_R u \rightarrow^{*}_{R\cup H} v \leftarrow^{*}_{R\cup H} t' \equiv C[e'\sigma_i\theta'_g] \xleftarrow{*}_R t
$$

が成立する (図 3.2)。□

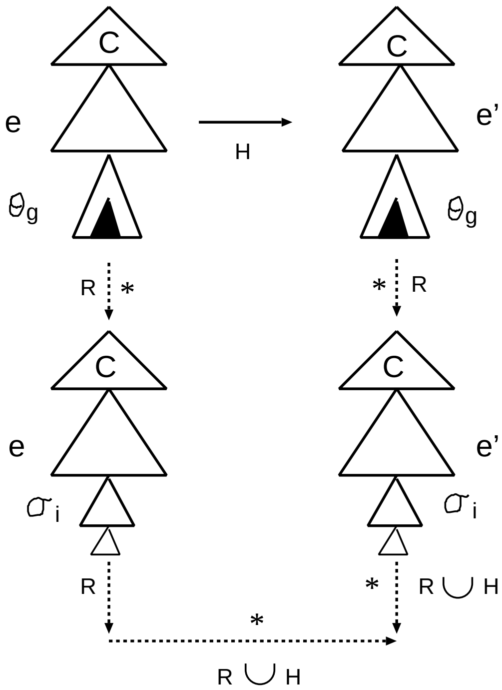

図 3.2: 補題 3.8 の証明

例 3.6 次の項書換えシステム Rを考える。

$$
R : \begin{cases} x + 0 \rightarrow x \\ x + s(y) \rightarrow s(x + y) \end{cases}
$$

$0+z=z$が Rの帰納的定理であることを示す。$H=\{0+z\to z\}$ とおく。このとき、$N=\{0,s(w)\}$ は R の被覆集合となる。これは明らかに、$\forall g\in T(F)\ \exists t\in N\ \exists\theta_g\ [g\xrightarrow{*}_R t\theta_g]$ が成立することより確かめられる。*(1),(2)* が成立することは明らか。次に (3′) が成立することを被覆集合を用いて示す。被覆代入集合 $S=\{\{z\leftarrow 0\},\{z\leftarrow s(w)\}\}$ に対し、$\forall\sigma\in S[0+z\sigma\to_R\cdot\xrightarrow{*}_{RUH}\cdot\xleftarrow{*}_{RUH} z\sigma]$ は以下のように成立する。

$0+0\to_R 0$ 、 $0+s(w)\to_R s(0+w)\to_H s(w)$。

Rと $R\cup H$は基底項上で、補題 *3.4*の条件 *(1)(2)(3)* を満たすので、$=_R\ =\ =_{R\cup H}\ in\ T(F)$。よって $0+z=z$は Rの帰納的定理である。□

例 3.7　次の項書換えシステム Rを考える。

$$R:\begin{cases} x+0\to x \\ x+s(y)\to s(x+y) \\ s(x)+y\to s(x+y) \\ double(0)\to 0 \\ double(s(x))\to s(s(double(x))) \end{cases}$$

$double(z)=z+z$が Rの帰納的定理であることを示す。$H=\{double(z)\to z+z\}$ とおく。このとき、$N=\{0,s(w)\}$ は R の被覆集合となる。*(1),(2)* が成立することは明らか。次に (3′) が成立することを被覆集合をもちいて示す。被覆代入集合 $S=\{\{z\leftarrow 0\},\{z\leftarrow s(w)\}\}$ に対し、$\forall\sigma\in S[(double(z))\sigma\to_R\cdot\xrightarrow{*}_{RUH}\cdot\xleftarrow{*}_{RUH}(z+z)\sigma]$ は以下のように成立する。

$double(0)\to_R 0\leftarrow_R 0+0$、

$double(s(w))\to_R s(s(double(w)))\to_H s(s(w+w))\leftarrow_R s(w+s(w))\leftarrow_R s(w)+s(w)$。

よって $double(z)=z+z$は Rの帰納的定理である。□

上記の例の場合は $R_2=R\cup H$が補題 3.4の条件を満たすので、直接帰納的定理であることを証明できた。しかし、一般には R_2が補題 3.4の条件を満たすとは限らない。このような場合は R_2のかわりに基底項上で、$=_{R2}\ =\ =_{R3}$ かつ補題 3.4の条件を満たす R_3を見つけても良い。このとき、$\forall\theta_g.\ e\theta_g=_{R2}e'\theta_g\Leftrightarrow\ \forall\theta_g.\ e\theta_g=_{R3}e'\theta_g\ \Leftrightarrow\ \forall\theta_g.\ e\theta_g=_{R1}e'\theta_g$ が成立することから $e=e'$が R_1の帰納的定理であることが示される。

被覆集合をもちいてこのような R_3を発見する手法の一つを簡単に説明する。$\{\sigma_i\}_i$を R_1の被覆代入集合とし、$R_2=R_1\cup\{e\to e'\}$ は強正規性をもつとする。ある s,t($s\not\equiv t$)$\in NF_{R_2}$ が存在して $e\sigma_j\to_{R_1}\cdot\xrightarrow{*}_{R_2} s$ かつ $e'\sigma_j\xrightarrow{*}_{R_2} t$ かつ $\forall\sigma_{i(\neq j)}\ [e\sigma_i\to_{R_1}\cdot\xrightarrow{*}_{R_2}\cdot\xleftarrow{*}_{R_2} e'\sigma_i]$ であったとする[2]。ここで s と t に向きづけを行ない、$R_3=R_2\cup\{s\to t\}$ とする。こ

---

[2] $\{\sigma_i\}_i$は R_1のテスト代入集合で、かつ基底項上で$\to_{R_1}$: CRとする。このとき、$\forall\sigma_i\exists s.e\sigma_i\to_{R_1}\cdot\xrightarrow{*}_{R_2} s$ が成立しないとき、つまり $\exists\sigma_i.e\sigma_i\in NF_{R_1}$の場合は、$NF^G_{R_1}\neq NF^G_{R_2}$が成立するため $e=e'$は帰納的定理でないことが証明される (補題 3.8)。

のとき $\forall\sigma_i\ [s\sigma_i \to_{R_1} \cdot \to^*_{R_3} \cdot \leftarrow^*_{R_3} t\sigma_i]$ が成立するなら基底項上で、$=_{R_2} = =_{R_3}$かつ (1)$\to_{R_1}\subseteq\to_{R_3}$,(2)$\to_{R_3}$: SN,(3) $\to_{R_3}$ RET $\to_{R_1}$は明らかに成立する (図 3.3)。よってこの場合は、$e = e'$が R_1の帰納的定理であることが示される。そうでない場合は、同様にこの手続きを繰り返す。このように上記の s,t が求まれば、直接 R_2が条件を満たさなくても帰納的定理であることが証明可能となる。

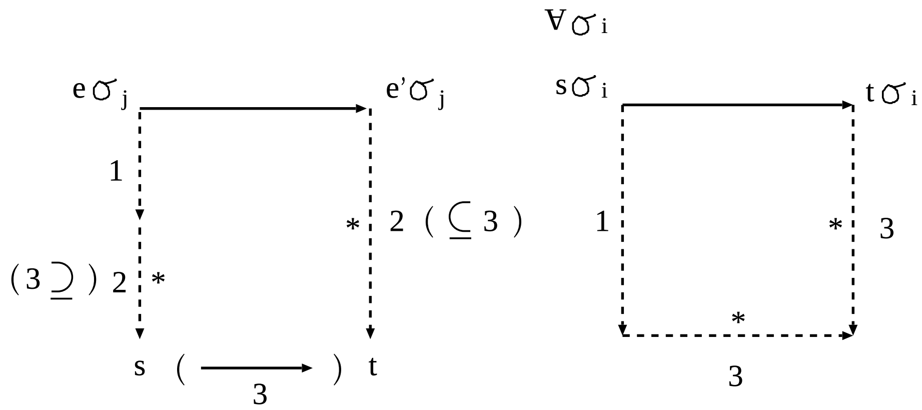

図 3.3: 基底項上で $\to_{R_3}$ RET $\to_{R_1}$

実際に、Reddy による証明手続き [23]、Bouhoula による SPIKE[4, 5] では、このような手法がもちいられている。

3.3 反駁証明法

反駁証明法とは等式 $e = e'$が項書換えシステム Rと帰納的に矛盾することを示すことによって非等式 $e \neq e'$の証明を行なう手法である [2, 3, 5, 8, 10, 11, 12, 15, 19, 21]。等式 $e = e'$が Rと帰納的に矛盾するとは $R' = R \cup \{e \to e'\}$ としたとき、$=_R \neq =_{R'}$ in T(F) が成立することである。ここでは、文献 [2, 10, 11, 21] によって提案された反駁証明法を抽象リダクションシステムの枠組で考察する。

補題 3.8 次の条件が成立するものと仮定する。

$$(1)\ \to_1 \subseteq \to_2 \quad (2)\ \to_2\!: SN \quad (3)\ \to_1\!: CR \quad (4) NF_1 \neq NF_2$$

このとき、$=_1 \neq =_2$ が成立する。

証明 (1) と (4) より $\exists x \in NF_1[x \to_2 y]$。(1) と (2) より $\exists z \in NF_1[y \xrightarrow{*}_1 z]$。(1) より $y \xrightarrow{*}_2 z$となり $x =_2 z$。ここで$=_1 = =_2$と仮定すると $x =_1 z$。(3) と命題 2.4(2) より $x \equiv z$ となる。$x \xrightarrow{+}_2 x$ より (2) に矛盾。□

補題 3.9 次の条件が成立するものと仮定する。

$$(1)\ \to_1 \subseteq \to_2 \quad (2)\ \to_1 : WN \quad (3)\ \to_1 : CR \quad (4)\ \exists x, y[x =_2 y \ \wedge \ \ x\downarrow_1 \not\equiv y\downarrow_1]$$

このとき、$=_1 \neq =_2$ が成立する。

証明 $=_1 = =_2$と仮定する。各 $x =_2 y$について、(2) より $\exists x'\ x\downarrow_1 \equiv x'$かつ $\exists y'\ y\downarrow_1 \equiv y'$。仮定より、$x' =_1 y'$。(3) と命題 2.4(2) より $x' \equiv y'$。すなわち、$x\downarrow_1 \equiv y\downarrow_1$。よって矛盾。□

一般に、反駁証明法をもちいる場合は、公理となる項書換えシステムが完備 (合流性＋強正規性) であることを仮定して、等式が帰納的に矛盾することを検出する [2, 3, 4, 5, 8, 9, 10, 11, 12, 15, 19, 21]。しかし、補題 3.9で示したように公理となるシステムの強正規性を仮定せずに反駁証明を行なうことも可能である。

補題 3.8、補題 3.9をもちいると、項書換えシステム R のもとで等式 $e = e'$ が帰納的定理でないことを証明することが可能となる。$R' = R \cup \{e \to e'\}$ として、基底項上で補題 3.8または補題 3.9の条件が成立することが示せたとする。このとき、$=_R \neq =_{R'}$ in T(F) が成立する。ここで $\forall \theta_g.\ e\theta_g =_R e'\theta_g$ が成立するなら、$=_R = =_{R'}$ in T(F) が成立することは明らか。よって、$\exists \theta_g.\ e\theta_g \neq_R e'\theta_g$ が得られ、$e = e'$が Rの帰納的定理でないことが証明される。

以下に、補題 3.8にもとづいた反駁証明法の例を示す。

例 3.10 次の項書換えシステム R_1を考える。

$$R_1 : \begin{cases} x + 0 \to x \\ x + s(y) \to s(x + y) \end{cases}$$

$s(z) = 0$ が R_1の帰納的定理でないことを示す。$R_2 = R_1 \cup \{s(z) \to 0\}$ とおく。このとき、R_1は合流性、R_2は強正規性をもつ。また、$NF^G_{R_1} = \{0, s(0), s(s(0)), \cdots\} \ \neq \ \{0\} = NF^G_{R_2}$ が成立する[3]。R_1と R_2は基底項上で補題 *3.8*の条件 (1)(2)(3)(4) を満たすので、$=_{R_1} \neq =_{R_2}$ *in T(F)*。よって $s(z) = 0$ は R_1の帰納的定理でないことが示された。□

[3] 自動証明手法では、$NF^G_{R_1} \neq NF^G_{R_2}$ の判定にテスト代入集合を利用することが多い [4, 5, 8, 10, 11, 12, 15]。

上記の例は R_2が補題 3.8の条件を満たすので、帰納的定理でないことが直接示された。しかし、$R_2 = R_1 \cup \{e \to e'\}$ が補題 3.8の条件を満たしていないなら、R_2のかわりに基底項上で$=_{R_2} \; = \; =_{R_3}$を満たし、かつ補題 3.8の条件を満たす R_3を見つけても良い。このとき、$=_{R_2} \; = \; =_{R_3} \; \neq \; =_{R_1} \; in \; T(F)$ が成立することから $e = e'$が R_1の帰納的定理でないことが示される。

例 3.11　次の項書換えシステム R_1を考える。

$$R_1 : \begin{cases} x + 0 \to x \\ x + s(y) \to s(x + y) \end{cases}$$

$0 + x = 0$ が R_1の帰納的定理でないことを示す。$R_2 = R_1 \cup \{0 + x \to 0\}$ とする。このとき $NF^G_{R_1} = NF^G_{R_2} = \{s^n(0)\}$ であるため補題 *3.8*は適用できない。ここで次の項書換えシステム R_3を考える。

$$R_3 : \begin{cases} x + 0 \to x \\ x + s(y) \to s(x + y) \\ 0 + x \to 0 \\ s(0) \to 0 \end{cases}$$

このとき $NF^G_{R_1} \neq NF^G_{R_3}$は明らか。$R_3$は基底項上で、$=_{R_2} \; = \; =_{R_3}$かつ補題 *3.8*を満たす。よって $0 + x = 0$ は R_1の帰納的定理でないことが示された。□

また、公理となる項書換えシステム R_1が基礎完備でない場合は、$=_{R_1} \; = \; =_{R'_1} \; in \; T(F)$ を満たし基礎完備な項書換えシステム R'_1を公理としてもちいても良い。

第 4 章

証明能力の比較

本章では、潜在帰納法と書換え帰納法の証明能力を比較する。3 章では、潜在帰納法と書換え帰納法を抽象リダクションシステムの枠組で形式化することにより、本質となる性質を明らかにした。

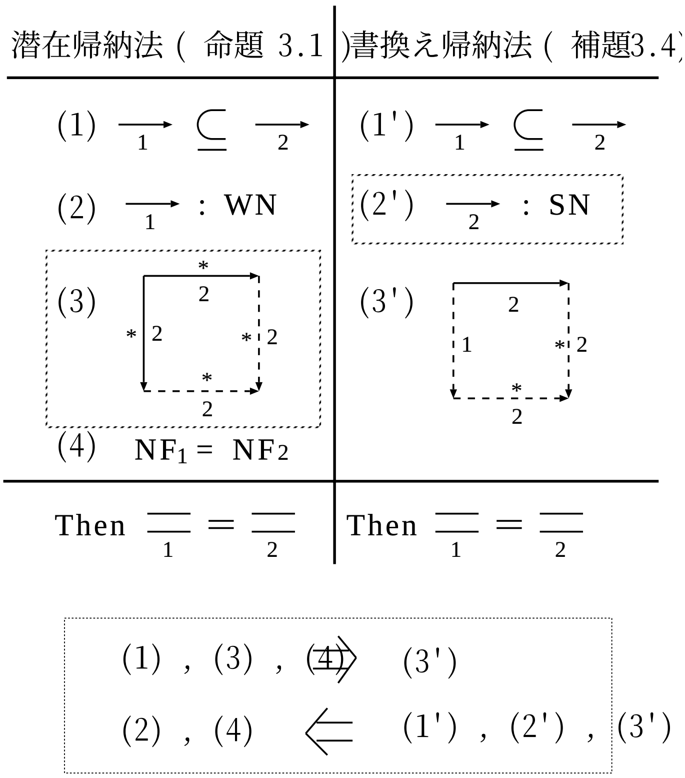

図 4.1: 命題 3.1 と補題 3.4 の比較

潜在帰納法では$\to_2$の合流性、書換え帰納法では $\to_2$の強正規性という独立した条件を要求するため、証明能力が異なることが分かる (図 4.1)。

実際に実現されている定理自動証明システムでは潜在帰納法と書換え帰納法にそれぞれ反駁証明法を組み合わせて用いることが多い [4, 5, 8, 10, 11, 12, 21]。以下に示す補題 4.1, 補題 4.2, 補題 4.3は、自動証明システムで多く用いられる手法を抽象リダクションシステムの枠組で定式化したものである。補題 4.1は命題 3.1と補題 3.8の組み合わせ、補題 4.2は補題 3.4と補題 3.8の組み合わせ、補題 4.3は補題 3.4と補題 3.8と補題 3.9の組合せである。

補題 4.1 次の条件が成立するものと仮定する。
(1) $\to_1 \subseteq \to_2$ *(2)* $\to_2$: *CR+SN* *(3)* $\to_1$: *CR*
このとき、$NF_1 = NF_2 \ \Rightarrow\ =_1 \,=\, =_2$ かつ $NF_1 \neq NF_2 \ \Rightarrow\ =_1 \,\neq\, =_2$ が成立する。

補題 4.2 次の条件が成立するものと仮定する。
(1) $\to_1 \subseteq \to_2$ *(2)* $\to_2$: *SN* *(3)* $\to_1$: *CR*
このとき、$\to_2 RET \to_1 \Rightarrow\ =_1 \,=\, =_2$ かつ $NF_1 \neq NF_2 \ \Rightarrow\ =_1 \,\neq\, =_2$ が成立する。

補題 4.3 次の条件が成立するものと仮定する。
(1)$\to_1 \subseteq \to_2$ *(2)*$\to_2$: *SN* *(3)*$\to_1$: *CR*
このとき、$\to_2 RET \to_1 \Rightarrow\ =_1 \,=\, =_2$ かつ

$(NF_1 \neq NF_2 \ \vee\ \exists x, y[x =_2 y \ \wedge\ \ x \downarrow_1 \not\equiv y \downarrow_1]) \ \Rightarrow\ =_1 \,\neq\, =_2$ が成立する。

例 4.4 補題 4.1にもとづいた帰納的定理の証明・反駁法について説明する。項書換えシステム R_1のもとで等式 $e = e'$について考える。$\{\sigma_i\}_i$を R_1のテスト代入集合とし、$R_2 = R_1 \cup \{e \to e'\}$ とおく。ここで、R_1と R_2は基底項上で補題 4.1の条件 *(1)(2)(3)* を満たすとする。このとき、

$$\forall \sigma_i.\ e\sigma_i \notin NF_{R_1} \Leftrightarrow NF^G_{R_1} = NF^G_{R_2} \Leftrightarrow \forall \theta_g . e\theta_g =_{R_1} e'\theta_g$$

が成立する。□

例 4.5 補題 4.3にもとづいた帰納的定理の証明・反駁法について説明する。項書換えシステム R_1のもとで等式 $s = t$ について考える。$R_2 = R_1 \cup \{s \to t\}$ とする。また、R_2は強正規性、R_1は合流性を満たすとする。$\exists \theta_g.\ s\theta_g \in NF_{R_1}$ の場合は、$s = t$ は R_1の帰納的定理でない。そうでない場合、すなわち $\forall \theta_g \exists s'[s\theta_g \to_{R_1} s']$ の場合を考える。このとき、$\exists u \in NF_{R_2}.\ s' \to^*_{R_2} u \leftarrow^*_{R_2} t\theta_g$ であることと、$s = t$ が R_1の帰納的定理であることは等価である。□

補題 4.1、補題 4.2、補題 4.3 で示したように反駁証明法を組み合わせてもちいるときは、命題 3.1、補題 3.4の条件に加えて$\rightarrow_1$の合流性、$\rightarrow_2$の強正規性が要求される。よって$\rightarrow_1$の合流性と$\rightarrow_2$の強正規性を仮定した場合、潜在帰納法 (命題 3.1) と書換え帰納法 (補題 3.4) の証明能力が一致したならば、2 つの手法にそれぞれ反駁証明法を組み合わせてもちいた手法の証明能力が一致することが示される。

補題 4.6　抽象リダクションシステム $R_1 = \langle A, \rightarrow_1 \rangle$ と $R_2 = \langle A, \rightarrow_2 \rangle$ を考える。$\rightarrow_1 \subseteq \rightarrow_2$ が成立していると仮定すると以下のことが成り立つ。

(a)$\rightarrow_2 RET \rightarrow_1 \Rightarrow\ NF_1 = NF_2$、
(b)$\rightarrow_2$: $CR \ \wedge\ NF_1 = NF_2 \ \Rightarrow\ \rightarrow_2 RET \rightarrow_1$、
(c)$\rightarrow_2 RET \rightarrow_1 \ \wedge\ \rightarrow_2$: $SN \wedge\ \rightarrow_1$: $CR \ \Rightarrow\ \rightarrow_2$: CR。

証明
(a) $NF_2 \subseteq NF_1$は自明。$NF_1 \subseteq NF_2$を示す。$x \notin NF_2$とする。すると $\exists y[x \rightarrow_2 y]$。このとき$\rightarrow_2 RET \rightarrow_1$より $\exists z[x \rightarrow_1 z]$。よって $x \notin NF_1$。
(b)$NF_1 = NF_2$より $\forall x, y \exists z[x \rightarrow_2 y \Rightarrow x \rightarrow_1 z]$。$\rightarrow_2$:CR と命題 2.4(1) より$\rightarrow_2 RET \rightarrow_1$。
(c)$\rightarrow_2$上のネータ帰納法で $\forall x, y, z[x \overset{*}{\rightarrow}_2 y \wedge x \overset{*}{\rightarrow}_2 z \Rightarrow \exists w\ y \overset{*}{\rightarrow}_2 w \wedge z \overset{*}{\rightarrow}_2 w]$ を示す。ここで$\rightarrow_2$は強正規性であることに注意する。$x \equiv y$あるいは$x \equiv z$のときは明らかに成り立つ。$x \rightarrow_2 y' \overset{*}{\rightarrow}_2 y$、$x \rightarrow_2 z' \overset{*}{\rightarrow}_2 z$とする。$\rightarrow_2 RET \rightarrow_1$より $\exists a, b[x \rightarrow_1 a \wedge a \overset{*}{\rightarrow}_2 b \wedge y' \overset{*}{\rightarrow}_2 b]$、$\exists a', b'[x \rightarrow_1 a' \wedge a' \overset{*}{\rightarrow}_2 b' \wedge z' \overset{*}{\rightarrow}_2 b']$ が成立。$\rightarrow_1$:CR より $\exists c[a \overset{*}{\rightarrow}_1 c \wedge a' \overset{*}{\rightarrow}_1 c]$。また、$\rightarrow_1 \subseteq \rightarrow_2$より、$x \rightarrow_2 a$、$x \rightarrow_2 a'$、$a \overset{*}{\rightarrow}_2 c$、$a' \overset{*}{\rightarrow}_2 c$、。帰納法の仮定を y', a, a', z'に繰り返し適用することにより $\exists w[y \overset{*}{\rightarrow}_2 w \wedge z \overset{*}{\rightarrow}_2 w]$。よって$\rightarrow_2$:CR。□

定理 4.7　抽象リダクションシステム $R_1 = \langle A, \rightarrow_1 \rangle$ と $R_2 = \langle A, \rightarrow_2 \rangle$ を考える。$\rightarrow_1$が合流性、$\rightarrow_2$が強正規性をもつと仮定したとき潜在帰納法 *(*命題 *3.1)* と書換え帰納法 *(*補題 *3.4)* の証明能力は一致する。

証明 潜在帰納法により$=_1 = =_2$ が証明できたとする。このとき、命題3.1の条件 (1)(2)(3)(4) が成立している。補題 3.4の条件 (1)(2) は明らかに成立。補題 4.6(b) より補題 3.4の条件 (3) は導出される。よって、書換え帰納法でも$=_1 = =_2$ は証明可能。逆に、書換え帰納法により$=_1 = =_2$ が証明できたとする。このとき、補題 3.4の条件 (1)(2)(3) が成立している。命題 3.1の条件 (1)(2) は明らかに成立。補題 4.6(a)(c) より命題 3.1の条件 (3)(4) は導出される。よって、潜在帰納法でも$=_1 = =_2$ は証明可能。□

潜在帰納法 (命題 3.1) と書換え帰納法 (補題 3.4) にそれぞれ、反駁証明法 (補題 3.8) を組み合わせた各証明法を図 4.2に示す。

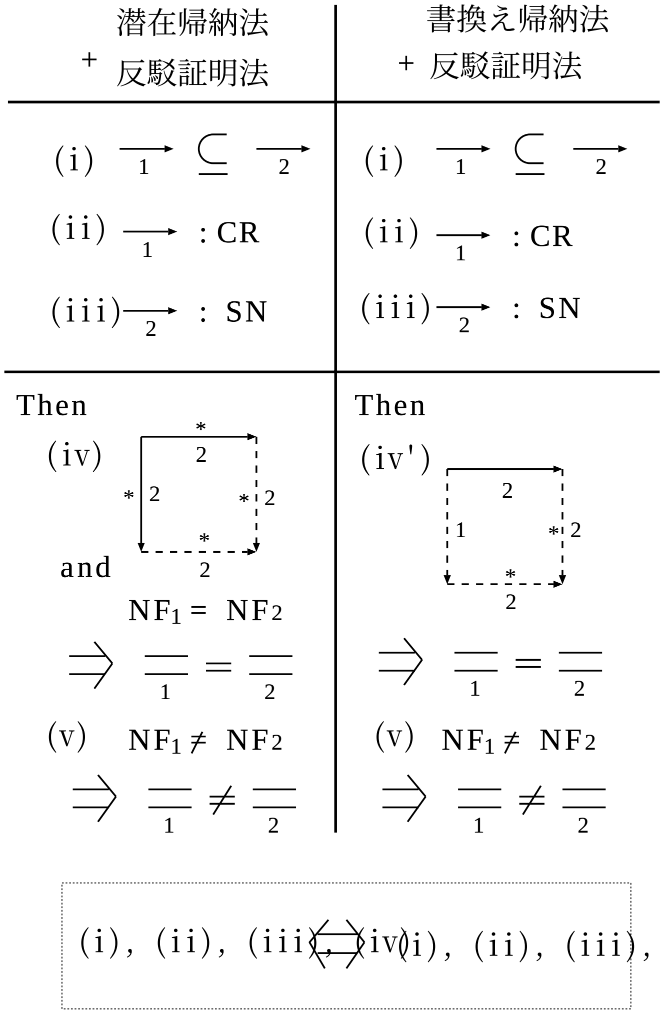

図 4.2: 証明法の比較

最後に、証明・反駁法の拡張可能性について考察する。上で述べたように反駁証明法と組み合わせてもちいたときは、一般に$\rightarrow_1$の合流性、$\rightarrow_2$の強正規性が要求される。しかし、命題 3.1と補題 3.9を組み合わせると強正規性を仮定せずに証明・反駁を行なえる。つまり、より弱い条件のもとで証明・反駁が可能となる。

補題 4.8 次の条件が成立すると仮定する。

(1)$\rightarrow_1 \subseteq \rightarrow_2$ *(2)*$\rightarrow_1$: $CR+WN$

このとき、$\rightarrow_2$: $CR \wedge NF_1 = NF_2 \Rightarrow =_1 = =_2$、$\exists x, y[x =_2 y \wedge \ x \downarrow_1 \not\equiv y \downarrow_1] \Rightarrow =_1 \neq =_2$ が成立する。□

補題 4.8で示した手法と補題 4.1, 補題 4.2, 補題 4.3の手法の証明能力を比較してみると、補題 4.1, 補題 4.2, 補題 4.3の手法より補題 4.8の手法の証明能力が高いことが分かる。

第 5 章

自動証明システムとの対応

3 章で述べたように、公理となるシステムのもとで等式 $e = e'$ が帰納的定理であることを命題 3.1や補題 3.4の枠組で証明する際にはそれぞれの条件判定が必要となる。3 章では、命題 3.1や補題 3.4の条件を満たす項書換えシステムを自動証明システムでどのように見つけるかについては言及してない。しかし、帰納的定理の自動証明システムでは、公理となる項書換えシステム R_1 と命題 3.1や補題 3.4の条件を満たす項書換えシステム R_2 を生成する手続きが重要となる。

これまで作られている自動証明システムでは、次のような手法が用いられている。まず、公理となるシステムを合同関係を保存したまま完備な項書換えシステム R_1 へと変形する。このような変形を行なう理由は命題 3.1や補題 3.4の条件、また反駁証明法の条件を判定するときに実装の上で適しているからである [4, 5, 8, 10, 11, 12, 15, 17, 22]。そして次に、$R_2 = R_1 \cup \{e \to e'\}$ とする。ここで R_2 が命題 3.1や補題 3.4の条件を満たしているなら、証明は完了である。しかしながら、直接 R_2 が条件を満たさないことは多い。よって、自動証明システムではこの R_2 をもとに命題 3.1や補題 3.4の条件を満たし合同関係を保存する R_3 の生成を行なう。

すなわち、潜在帰納法と書換え帰納法に基づいた手続きでは証明を行なうとき、それぞれ次の条件を満たす項書換えシステム R_3 の発見が重要となる。

潜在帰納法

(1) $\to_{R_1} \subseteq \to_{R_2}$, (2) $\to_{R_1}: WN$, (3) $\to_{R_3}: CR$, (4) $NF_{R_1} = NF_{R_3}$, (5) $=_{R_2} = =_{R_3}$
このとき、$=_{R_1} = =_{R_2} = =_{R_3}$ が成立する。

書換え帰納法

(1′) $\to_{R_1} \subseteq \to_{R_3}$, (2′) $\to_{R_3}$: SN, (3′) $\to_{R_3}$ RET $\to_{R_1}$, (4′) $=_{R_2} = =_{R_3}$
このとき、$=_{R_1} = =_{R_2} = =_{R_3}$ が成立する。

これら2つの比較をすると、退行性を満たすが合流性を満たすR_3への変換が困難な場合は、書換え帰納法に基づいた証明システムの方が有効で、停止性を満たすシステムへの変換が困難なときは、潜在帰納法に基づいた証明システムが有効であると考えられる。しかし、この差に関しては現在の実装技術に依存するため一般に比較するのは困難である。

本章では、実際に作成されている自動証明システムにおいてこのような変換がどのように実装されているか紹介し、考察を行なう。

5.1 帰納的定理を証明する自動証明システム

完備でない項書換えシステムを、合同関係を保存したまま完備な項書換えシステムへ変換する手続きとして Knuth-Bendix の完備化手続きがある [18]。この完備化手続きを利用した帰納的定理の自動証明システムとして Kapur による手続き [12]、Fribourg による手続き [8] が提案されている。

本節では、これらの手続きを紹介し、潜在帰納法と書換え帰納法に基づく自動証明システムでは、目標とする枠組が異なるため証明能力に差がでてくることを示す。また、これらの証明システムは反駁証明法を組み合わせたものであるため、この証明能力の差は変換システムの差であって帰納法の能力自体の差ではないことも示す。

最初に、Fribourg の手続きで用いられる概念に関する定義を行なう。

定義 5.1 [*8*] 項 t における出現位置 p が R 完全な出現 *(complete position)* とは以下が成り立つことである。

$$x\sigma_g \in NF_R^G \text{となる任意の基底代入} \sigma_g \text{に対し、} t\sigma_g|_p \text{はリデックス。}$$

また、上記の性質を満たす出現位置は帰納的に完全な出現位置 (inductively complete position) とも呼ばれている。

例 5.2 次の項書換えシステム R を考える。

$$R : \begin{cases} x + 0 \to x \\ x + s(y) \to s(x + y) \end{cases}$$

このとき項 $x + (y + z)$ について出現位置 $p = 2$ は R 完全な出現となる。□

以下に示すKapurによって提案された手続きは、命題3.1の条件を満たす項書換えシステムへの変換手続きであり、Fribourgによって提案された手続きは補題3.4の条件を満たす項書換えシステムへの変換手続きに対応している。これは我々の枠組において、Kapurの手続きは潜在帰納法に対応する手続き、Fribourgの手続きは書換え帰納法に対応する手続きと解釈される。また、これらの手続きは実際には反駁証明法と組み合わせて用いられている。

手続き 1. (Kapur の手続き)

Input: R_1:完備な項書換えシステム； $e = e'$:証明すべき等式； $>$: 停止性を保証する項上の順序

(1) $R_2 = R_1$; $E = \{e = e'\}$ とする。

(2) E が空なら終了。このとき $e = e'$ は R_1の帰納的定理であることが示される。E が空でないなら (2-1) から (2-4) を繰り返す。

(2-1) E から $s > t$ を満たす等式 $s = t$ をひとつ取り出し、$s \to t$ を R_2に付け加える。もしそのような等式がないなら終了。このとき証明は失敗。

(2-2) $NF^G_{R_1} \neq NF^G_{R_2}$ なら終了。このとき $e = e'$ は R_1の帰納的定理でないことが示される。

(2-3) $s \to t$ と R_2によって生じたすべての危険対を E に加える。

(2-4) R_2を用いて E の等式の両辺をすべて正規形にする。両辺が一致した等式は E から取り除く。

手続き 2. (Fribourg の手続き)

Input: R_1:完備な項書換えシステム； $e = e'$:証明すべき等式； $>$: 停止性を保証する項上の順序

(1) $R_2 = R_1$; $E = \{e = e'\}$ とする。

(2) E が空なら終了。このとき $e = e'$ は R_1の帰納的定理であることが示される。E が空でないなら (2-1) から (2-4) を繰り返す。

(2-1) E から $s > t$ を満たす等式 $s = t$ をひとつ取り出し、$s \to t$ を R_2に付け加える。もしそのような等式がないなら終了。このとき証明は失敗。

(2-2) $NF^G_{R_1} \neq NF^G_{R_2}$ なら終了。このとき $e = e'$ は R_1の帰納的定理でないことが示される。

(2-3) $s \to t$ と R_1の間で、項 s の R_1完全な出現位置[4] を適当に 1 箇所以上選び、そこで生成されるすべての危険対を E に加える。もしそのような出現位置がなければ終了。こ

[4] 項 s の出現位置 p が R_1完全であるかどうかは決定可能である [8]。

のとき証明は失敗。

(2-4) R_2を用いて E の等式の両辺をすべて正規形にする。両辺が一致した等式は E から取り除く。

Kapur の手続は Kunuth-Bendix の完備化手続きを、基底項上で正規形の集合の等価性判定が行なえるよう改良したものとなっている。これは命題 3.1で示したようにシステムの等価性を示すために、合流性と正規形の集合の等価性の判定が必要になるためである。また Fribourg の手続きでは、すべての危険対を付け加える必要のないように完備化手続きを改良している。元々の公理となるシステム R_1と、R_1完全な出現位置での危険対のみ付け加えれば良い理由は、補題 3.4で示したように書換え帰納法で本質となるのは合流性でなく退行性となるからである。実際には、以下の図 5.1を保証するように設計している。

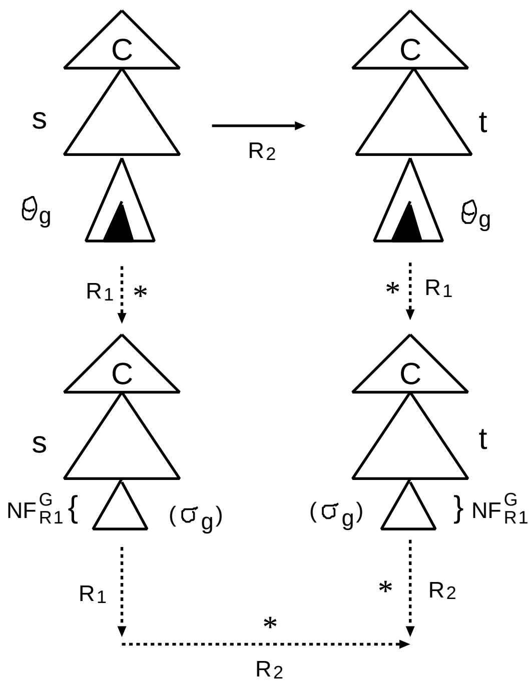

図 5.1: 手続き 2 の説明

この手法により等式の生成が制限でき手続き 1 の証明システムに比べ効率の良い自動証明が可能となる。線形戦略 (linear sterategy) と呼ばれるこの手法は Buchmaire により提案された帰納的定理の証明手続き [2]、Gramlich により提案された帰納的定理の証明手続き [9] でも用いられている。手続き 1,2 は入力されたシステムから結果として基礎完備な項書換えシステムをつくりだすため、帰納的完備化手続き (inductive completion procedure)

とも呼ばれる。また、手続き 2 は反駁証明と組み合わせてもちいるため入力となる項書換えシステムを完備としている。その結果として、証明が成功したとき基礎完備な項書換えシステムを作り出す。しかし、入力となる項書換えシステムに合流性を仮定しなくても証明は可能である。このとき結果としてつくられる項書換えシステムは基底項上で、退行性をもつが合流性をもつとは限らない。よって本研究では書換え帰納法に対応する手続きとして扱っている。

次に、一方の手続きで証明に成功し、他法の手続きでは証明に失敗する例を紹介する。

例 5.3 [8]　手続き *2* で証明可能で手続き *1* では証明できない例を示す。次の完備な項書換えシステム R_1 を考える。

$$R_1 : \begin{cases} 0 + x \to x \\ s(x) + y \to s(x+y) \end{cases}$$

このとき、$x + (y + z) = (x + y) + z$ の証明をを行なう

手続き *2* では、次のように証明が実行される。等式に向きづけを行ない $x+(y+z) \to (x+y)+z$ とする。このとき $p = \{\varepsilon, 2\}$ は項 $x+(y+z)$ の R_1 完全な出現位置となる。ここで、出現位置 ε を選び危険対の生成を行なうと、$(y+z) = (0+y)+z$ と $s(x+(y+z)) = (s(x)+y)+z$ の *2* つの等式が E に付け加えられる。これらの等式を $R_1 \cup \{x + (y + z) \to (x + y) + z\}$ をもちいて正規形にすると、両辺が一致するため削除される。この結果、次の R' が作られ証明は完了する。

$$R' : \begin{cases} 0 + x \to x \\ s(x) + y \to s(x+y) \\ x + (y + z) \to (x + y) + z \end{cases}$$

一方、手続き *1* では出現位置 *2* における危険対も生成し、その結果 $x + s^n(y + z) = (x + s^n(y)) + z$ となる等式が無限に生成されるため証明に失敗する。□

上記の例では、手続き 2 で証明に成功し手続き 1 では証明に失敗する例を紹介した。しかし、これは書換え帰納法で証明可能で潜在帰納法で証明できないということを意味しているわけではない。手続き 1 で証明に失敗した原因は、完備化手続きを利用し、合流性をもつシステムへと変形しているためである。実際、R_1 に証明すべき等式を加えた $R_2 = R_1 \cup \{x + (y + z) \to (x + y) + z\}$ は基底項上で命題 3.1の条件を満たすため、潜在帰納法により証明可能である。

また、出現位置 2 を選ぶと手続き 2 でも証明に失敗する。完全な出現位置が複数箇所存在する場合、出現位置の選択は証明を成功させる上でも、また効率良く手続きを実行するためにも重要となる。

逆に手続き 1 で証明可能で手続き 2 で証明に失敗する例も存在する。

例 5.4 [*20*]　手続き *1* で証明可能で手続き *2* では証明できない例を示す。次の完備なシステム R_1を考える。

$$R_1 : \begin{cases} (1) \quad app(Nill, x) \to x \\ (2) \quad app(cons(x, y), z) \to cons(x, app(y, z)) \\ (3) \quad rev(Nill) \to Nill \\ (4) \quad rev(cons(x, y)) \to app(rev(y), cons(x, Nill)) \end{cases}$$

このとき、$rev(rev(x)) = x$ の証明を行なう。

手続き *1* では、次のように証明が実行される。(5) $rev(rev(x)) \to x$ とする。(4)(5) の危険対から次の規則が作られる。(6) $rev(app(rev(y), cons(x, Nill))) \to cons(x, y)$。*(5)(6)* の危険対から次の規則が作られる。(7) $rev(app(y, cons(x, Nill))) \to cons(x, rev(y))$。*(6)* の規則は削除され結果として完備な次の R'が作られ証明は完了する。

$$R' : \begin{cases} app(Nill, x) \to x \\ app(cons(x, y), z) \to cons(x, app(y, z)) \\ rev(Nill) \to Nill \\ rev(cons(x, y)) \to app(rev(y), cons(x, Nill)) \\ rev(rev(x)) \to x \\ rev(app(y, cons(x, Nill))) \to cons(x, rev(y)) \end{cases}$$

一方、手続き *2* では証明の実行途中に次のような等式が無限に生成されるため証明に失敗する。

$$\begin{aligned} rev(app(rev(y), cons(x, Nill))) &= cons(x, y) \\ rev(app(app(rev(z), cons(y, Nill)), cons(x, Nill))) &= cons(x, cons(y, Nill)) \\ &\cdots \end{aligned}$$

このように手続き *2* では線形戦略を用いるため、(7) のような適当な補題が発見できず、証明に失敗する。□

上記の例では、手続き 1 で証明に成功し手続き 2 では証明に失敗する例について紹介した。しかし、これは潜在帰納法で証明可能で書換え帰納法で証明できないということを意味しているわけではなく、補題 3.4を満たすシステムの発見に失敗しているだけである。実際、例 5.4は次のようにすると補題 3.4に基づく証明が可能である。

$H = \{rev(rev(x)) \to x\}$ と置き、$R_2 = R_1 \cup H$とする。ここで$=_{R_2} = =_{R_3}$ in $T(F)$ を満たす、次の R_3を考える。

$$R_3 = R_1 \cup \{rev(rev(x)) \to x, rev(app(y, cons(x, Nil))) \to cons(x, rev(y))\}$$

R_3は、基底項上で補題 3.4の条件を満たす。基底項上で、$=_{R_2} = =_{R_3}$かつ補題 3.4の条件を満たす R_3が発見できたため、$rev(rev(x)) = x$ は R_1の帰納的定理であることが示された。

このように、潜在帰納法に対応した自動証明システムと書換え帰納法に対応した自動証明システムでは目標とする枠組が異なるため、証明能力に差が出てくることもある。しかし、ここで取り上げた例のように帰納法の証明能力自体に差があるわけではない。これは、定理 4.7からも分かる。すなわち、条件を満たす項書換えシステムへの変換手続きが実装において重要となる。

5.2 反駁証明法を組み合わせた自動証明システム

潜在帰納法と書換え帰納法は補題 4.1、補題 4.2、補題 4.3のように反駁証明法と相性良く組み合わせることが可能である。実際、前節で紹介した手続き 1,2 は証明と反駁証明を並行して実行する手続きである。本節ではこのような組み合わせに関する考察を行なう。

最初に手続き 1,2 において反駁証明法がどのように組み込まれているかを説明する。手続き 1,2 では完備な項書換えシステム R_1と証明すべき等式 $e = e'$を入力とし、$R_2 = R_1 \cup \{e \to e'\}$ を合同関係を保存したままそれぞれの目的を満たす R_3へと変形している。そして、手続きの途中で $NF^G_{R_1} \neq NF^G_{R_3}$ の検出が行なえた時点で手続きを終了し、$e = e'$ は帰納的定理でないことが示される[5] 。このとき付け加えられた規則に対応する等式を反証と呼ぶ。これが反駁証明となる。なぜならば R_3は基底項上で次の反駁証明法としての性質をもつからである。

反駁証明法

(1) $\to_{R_1} \subseteq \to_{R_3}$, (2) $\to_{R_3}$: SN, (3) $\to_{R_1}$: CR. (4) $NF_{R_1} \neq NF_{R_3}$, (5) $=_{R_2} = =_{R_3}$
このとき、$=_{R_1} \neq =_{R_2} = =_{R_3}$が成立する。

このように、手続き 1,2 は、Kunuth-Bendix の完備化手続きを利用し、危険対の生成により反証を検出しながら変換を行ない、反証が得られずに完備化が成功したとき結果とし

[5] 2 つの項書換えシステム R_1と R_2において $NF^G_{R_1} = NF^G_{R_2}$は決定可能である [15, 17, 22]。しかし、この計算には指数時間かかることが知られている [14, 17]。実際の自動証明システムでは効率良く実現するために、テスト集合をもちいるなど様々な手法がもちいられる。詳細は文献 [5, 8, 10, 11, 12, 15] 参照。

て帰納的定理であることが証明される。Kunuth-Bendix の完備化手続きを利用したこの種の証明法は無矛盾性による証明 (proof by consistency) とも呼ばれている。

帰納的定理でない等式の証明を実行したとき、命題 3.1や補題 3.4に基づく証明手続きのみでは無限ループにおちいって発散してしまう可能性もある。しかし、手続き 1,2 のように反駁証明法も並行して実行することで反証が得られれば、その時点で手続きを中止し帰納的定理でないことを示すことが可能となる。また、危険対の生成と反証の検出との関係については文献 [2, 3] で分かりやすく解説されている。ここで提案されている手続きは Fribourg の手続の拡張であり、等式の向きづけを必要としない反駁的に完全 (証明すべき等式が帰納的定理でないなら必ず反証が得られる) な手続きである。

このように、反駁証明法との組合せは自動証明システムにおいて重要となる。しかし、すでに述べたように反駁証明法を行なうとき、公理となる項書換えシステムには完備であることが要求される。よって自動証明システムにおいて、公理となるシステムを完備な項書換えシステムに変形することが困難な場合は、反駁証明は不向きである。

第 6 章

おわりに

6.1 まとめ

本研究の成果をまとめると以下のようになる。

1. 潜在帰納法と書換え帰納法を抽象リダクションシステムの枠組で考察し、両者の本質的な差異が合流性と退行性、および弱正規性と強正規性の差異にもとづくことを明らかにした。

2. 実際の自動証明システムで広くもちいられている反駁証明法についても同様に抽象リダクションシステムの枠組で考察し、潜在帰納法と書換え帰納法にそれぞれ反駁証明法を組み合わせてもちいたとき、実質的には証明能力が一致することを示した。

3. 実際に作成されている帰納的定理の自動証明システムをもちいて証明を実行した際、結果として作られる項書換えシステムが、実は我々の枠組に対応する性質をもつ項書換えシステムとなることを示した。また、ここで紹介した自動証明システムで証明に失敗する例を取り上げ、その原因が実装技術の問題であって帰納法の限界ではないことを示した。

6.2 今後の課題

今後の課題としては以下のことが考えられる。

1. 自動証明に適したより強力な証明・反駁手法の提案。

2. 適切な危険対を生成する手法。

5 章で述べたように自動証明システムでは適切な危険対の発見ができずに、発散してし

まうことがある。このような発散を防ぐ適切な危険対の発見手法は非常に重要となる。また、このような危険対の発見は自動証明システムの効率においても重要となる。

3. 2 つのシステムの等価性判定法の応用。

命題 3.1, 補題 3.4, 補題 3.8, 補題 3.9 の枠組をもちいると、ある集合上で条件部が成立することを示すことにより、その集合上での 2 つのシステムの等価性を示すことが可能となる。本研究では、項書換えシステムにおいて基底項全体からなる集合に着目することにより、帰納的定理の証明に応用した。適当な集合 (特定の構造をもつ項の集合や特定の関数記号からなる項の集合など) に着目することにより、帰納的定理以外の自動証明への応用について考察していくことは、今後の課題として興味深い。文献 [25] では、このような観点からの応用に関しても述べられている。

謝辞

本研究を行なうにあたり、また学生生活全般に渡り終始変わらぬ御指導を賜わりました外山芳人教授に心から感謝いたします。

適切な助言を頂いた鈴木太朗助手に感謝いたします。本論文の改善のために多くの助言をして頂いた草刈圭一朗氏に感謝いたします。

さらに学生生活を共にすごしてくださった外山研究室の皆様に心から御礼申し上げます。

また何年もの間、学生生活を支えてくれた家族に感謝します。最後に Yuu Yuu のスタッフの皆様には、やすらぎの場を提供して頂き、ここに感謝の意を表します。

参考文献

[1] F.Baader and T.Nipkow "Term Rewriting and All That", Cambridge University Press, 1998.

[2] L.Bachmair, "Proof by consistency in equational theories", In Proc 3rd IEEE Synp Logic in Computer Science 1988, pp.228-233, 1988.

[3] L.Bachmair, "Canonical equational proofs", Birkhauser Press, 1991.

[4] A.Bouhoula, E.Kounalis and M.Rusinowitch "Automated Mathematical induction", Journal of Logic and Computation 5 (5), pp.631-668, 1995.

[5] A.Bouhoula, "Automated theorem proving by test set induction", Journal of Symbolic Computation 23, pp.47-77, 1997.

[6] F.Bronsard, U.S.Reddy and R.W.Hasker "Induction using term order", Journal of Automated Reasoning 16, pp.3-37, 1996.

[7] R.S.Boyer and J.S.Moore "A Computational Logic", Academic Press, New York, 1979.

[8] L.Fribourg, "A strong restriction of the inductive completion procedure", Journal of Symbolic Computation 8, pp.253-276, 1989.

[9] B.Gramlich, "Completion based inductive theorem proving: An abstract framework and its applications", In Proc.ECAI-90, pp.314-319, 1990.

[10] G.Huet and J.M.Hullot, "Proof by induction in equational theories with constructors ", Journal of Computer and System Science 25, pp.239-266, 1982.

[11] J.P.Jouannand and E.Kounalis "Automatic proof by induction in theories without constructors", Information and Computation 82 (1), pp.1-33, 1989.

[12] D.Kapur, P.Narendran and H.Zhang "Proof by induction using test sets", Lecture Notes in Computer Science 230, pp.99-117, 1986.

[13] D.Kapur and D.R.Musser "Proof by Consistency", Artificial Intelligence 31, pp.125-157, 1987.

[14] D.Kapur, P.Narendran and H.Zhang "Complexity of sufficient completeness", Lecture Notes in Computer Science 241, pp.426-442, 1986.

[15] D.Kapur, P.Narendran and H.Zhang "Automating inductionless induction using test set", Journal of Symbolic Computation 11 , pp.83-111, 1991.

[16] D.Kapur and H.Zhang "A Rewrite Rule Laboratory", Lecture Notes in Computer Science 310, pp.768-769, 1988.

[17] D.Kapur, P.Narendran and H.Zhang "On sufficient-completeness and related properties of term rewriting system", Acta Information 24, pp.395-415, 1987.

[18] D.Knuth and P.Bendix "Simple word problems in universal algebras", In Computational Problems in Abstract Algebras, Oxford Pregamon Press, pp.263-297, 1970.

[19] E.Kounalis and M.Rusinowitch "Mechanizing inductive reasoning", In Proc of the American Association for Artificial Intelligence, pp.240-245, 1990.

[20] W.Kuchlin "Inductive completion by ground proof transformation", In Resolution of Equation in Algebraic Structures (Vol.2):Rewriting Techniques,Boston Academic Press, 1989.

[21] D.R.Musser "On proving induction properties of abstract data types", In Proc. 7th ACM Symp. Principles of Programming Languages, pp.154-162, 1980.

[22] D.Plaisted "Semantic confluence tests and completion methods", Information and Control 65, pp.192-235, 1985.

[23] U.S.Reddy, "Term rewriting induction", Lecture Notes in Computer Science 449, pp.162-177, 1990.

[24] J.J.Thiel, "Stop losing sleep over incomplete data type specifications", In Proc. 11th ACM Symp. Principles of Programing Languages, pp.76-82, 1984.

[25] Y.Toyama, “How to prove equivalence of term rewriting system without induction ”, Theoretical Computer Science 90, pp.369-390, 1991.

[26] H.Zhang, D.Kapur and M.S.Krishnamoorthy, “A mechanizable induction principle for equational specifications”, Lecture Notes in Computer Science 310, pp.162-181, 1988.

[27] 小池広高, 外山芳人, “潜在帰納法と書換え帰納法の比較”, 信学技報, COMP98-59, pp.65-72, 1998-11.

[28] 小池広高, 外山芳人, “潜在帰納法と書換え帰納法の比較”, 平成 10 年度電機関係学会北陸支部連合大会講演論文集, pp.260, 1998-10.