社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
MSS2011-63 SS2011-48 (2012-1)

単純型付き項書換え系における書換え帰納法について

尾関　朗 †　草刈圭一朗 ††　坂田　翼 †
西田　直樹 ††　酒井　正彦 ††　坂部　俊樹 ††

†, †† 名古屋大学大学院情報科学研究科　〒464-8603 名古屋市千種区不老町
E-mail: †{ozeki,sakata}@trs.cm.is.nagoya-u.ac.jp, ††{kusakari,nishida,sakai,sakabe}@is.nagoya-u.ac.jp

あらまし　書換え帰納法は，与えられた等式が帰納的定理であるかどうかを推論規則を用いた導出により判定する原理であり，項書換え系上で提案され様々な拡張がなされてきた．本論文では、書換え帰納法を単純型付き項書換え系上へ拡張することにより，高階関数を含む等式に対しての書換え帰納法による帰納的定理の自動証明法を実現する．また，書換え帰納法の推論規則が満たすべき性質を定式化し，この定式化に基づいて各推論規則を適切に設計する．

キーワード　帰納的定理の自動証明，書換え帰納法，単純型付き項書換え系，高階関数

On Rewriting Induction for Simply-typed Term Rewriting Systems

Akira OZEKI†, Keiichirou KUSAKARI††, Tsubasa SAKATA†,
Naoki NISHIDA††, Masahiko SAKAI††, and Toshiki SAKABE††

†, ††Graduate School of Information Science,Nagoya University
Furocho, Chikusa-ku, Nagoya-shi, 464–8603 Japan
E-mail: †{ozeki,sakata}@trs.cm.is.nagoya-u.ac.jp, ††{kusakari,nishida,sakai,sakabe}@is.nagoya-u.ac.jp

Abstract　Rewriting induction is a principle of proving inductive theorems by means of derivations obtained by applying inference rules. The principle was first proposed on term rewriting systems, and was extended variously. In this paper, we propose the rewriting induction to simply-typed term rewriting systems, implementing an automated inductive theorem proving for higher-order equations. In order to design inference rules suitably, we also formulize properties which inference rules should satisfy.

Key words　automated inductive reasoning, rewriting induction, simply-typed term rewriting, higher-order function

1. 序　　論

帰納的定理の自動証明はプログラムの検証や等価性判定などを自動で行うために重要であり，関数型プログラムの計算モデルの 1 つである項書換え系 (TRS) 上で広く研究されてきた．Reddy により提案された書換え帰納法 [6] はその証明手法の 1 つであり，与えられた等式が帰納的定理であることを推論規則を用いた導出を行うことにより証明する．また，Bouhoula らにより反証機能が追加され，帰納的定理でないという判定を下すことも可能である [1]．しかし，TRS は高階関数を表現することができず，これまで高階関数を含む等式に対して書換え帰納法を利用することはできなかった．

本論文では，書換え帰納法を単純型付き項書換え系 (STRS) に対応するよう拡張する．STRS は TRS に単純型の概念を導入したもので，TRS では表現できなかった高階関数を扱うことが可能である [3]．代表的な高階関数である $foldl$ と $foldr$ は以下の STRS で表現できる．

$$\left\{\begin{array}{rcl} foldl[f,xs,nil] & \rightarrow & xs \\ foldl[f,xs,cons[x,ys]] & \rightarrow & foldl[f,f[xs,x],ys] \\ foldr[f,xs,nil] & \rightarrow & xs \\ foldr[f,xs,cons[x,ys]] & \rightarrow & f[x,foldr[f,xs,ys]] \end{array}\right.$$

書換え帰納法の基本原理に関して，書換え帰納法の本質が停止性と退行性であると主張されている [7]．これらの性質は確かに書換え帰納法の本質を捉えてはいるが，各個別の推論規則を適切に設計するには余りに抽象度が高過ぎる．

本論文では，書換え帰納法での証明が適切に行われるために推論規則が満たすべき性質を明らかにする．その上で，書換え帰納法で使用することができる推論規則を与える．

STRS における帰納的定理は TRS での帰納的定理の定義を

自然な形で拡張した原始帰納的定理に基づき与えられる [4]．本論文で与える STRS 上の書換え帰納法は原始帰納的定理か否かを証明する手法である．よって，帰納的定理の証明には直接適用可能であるが，反証を行うためにはいくつかの条件が成立する必要が出てくる．また書換え帰納法を繰り返し適用することにより原始帰納的定理でない帰納的定理を証明する手法も与える．

2. 準 備

本論文では，単純型付き項書換え系に関する用語は文献 [4] [5] に従う．

基底型の集合 $\mathcal{B}$ に対して**単純型**の集合は $\mathcal{S} ::= \mathcal{B} \mid (\mathcal{S}_1 \to \mathcal{S}_2) \mid (\mathcal{S}_1 \times \cdots \times \mathcal{S}_n)$ で定義される．$\alpha \to \beta$ の形の単純型を**関数型**，そうでない型を**非関数型**といい，非関数型の集合を $\mathcal{S}_{nfun}$ で表す．

関数記号の集合 Σ は**被定義記号**の集合 $\mathcal{D}$ と**構成子記号**の集合 $\mathcal{C}$ に分割される $(\Sigma = \mathcal{D} \uplus \mathcal{C})$ とする．変数記号の集合 $\mathcal{V}$ と Σ から生成される**型無し項**の全体からなる集合 $\mathcal{T}(\Sigma, \mathcal{V})$ を，$a \in \Sigma \cup \mathcal{V} \wedge t_1, \ldots, t_n \in \mathcal{T}(\Sigma, \mathcal{V}) \Rightarrow a[t_1, \ldots, t_n] \in \mathcal{T}(\Sigma, \mathcal{V})$ となる最小の集合で定義する．

型関数 τ は $\mathcal{V} \cup \Sigma$ から $\mathcal{S}$ への関数であり，項上に自然に拡張される．$\tau(t) = \alpha$ であるとき t^α で記す．**単純型付き項**全てからなる集合を $\mathcal{T}_\tau(\Sigma, \mathcal{V})$ で記す．変数が出現しない項を**閉じた項**という．また $\mathcal{T}_\tau(\mathcal{C}, \mathcal{V})$ 中の項を**構成子項**という．型付き項 t が非関数型の閉じた項であるとき t を**基底項**であるといい，基底項の集合を $\mathcal{T}_{nfun}(\Sigma)$ で表す．高階変数でない変数の集合を $\mathcal{V}_{nfun}$ で記す．

同じ型の項の対 (l, r) が $root(l) \in \mathcal{D}$ かつ $Var(l) \supseteq Var(r)$ であるとき**書換え規則**といい，$l \to r$ で記す．書換え規則の集合 R から生成される**書換え関係**を $\xrightarrow[R]{}$ で記す．**単純型付き項書換え系** (STRS) とは書換え規則の集合である．

STRS R の正規形の集合を $NF(R)$ で記す．R が合流性，弱正規性，強正規性を持つことをそれぞれ $CR(R), WN(R), SN(R)$ で記す．また，基底項上でこれらの性質が成立することをそれぞれ $GCR(R), GWN(R), GSN(R)$ で記す．

等式集合 E について $s \twoheadleftarrow\!\!\!\!\twoheadrightarrow_E t$ を $s \equiv C[s_1, \ldots, s_n] \wedge t \equiv C[t_1, \ldots, t_n] \wedge \forall i.s_i \xleftrightarrow[E]{} t_i$ で定義する．

等式 $s \simeq t$ について，$\tau(s) = \alpha_1 \to \cdots \to \alpha_n \to \alpha$ かつ $\alpha \in \mathcal{S}_{nfun}$ とする．このとき $s \simeq t$ の**完全拡張形** $(s \simeq t)^\uparrow$ を新しい変数 $z_1^{\alpha_1}, \ldots, z_n^{\alpha_n}$ を用いて $(s \simeq t)^\uparrow = s[z_1, \ldots, z_n] \simeq t[z_1, \ldots, z_n]$ で定義する．また，等式集合 E に対して $E^\uparrow$ を $E^\uparrow = \{(s \simeq t)^\uparrow \mid s \simeq t \in E\}$ で定義する．

各 $a^{\alpha_1 \to \cdots \to \alpha_n \to \alpha} \in \mathcal{C}$ で，$\forall i.\alpha_i \in \mathcal{S}_{nfun}$ となるとき，$\mathcal{C}$ は *1* **階の構造を持つ**という．

項 $t \in \mathcal{T}(\mathcal{C}, \mathcal{V}_{fun})$ が**擬似値**であるとは，項 t のどの変数の出現位置も葉の位置であることをいい，全ての擬値の集合を $PVal(\mathcal{C}, \mathcal{V}_{fun})$ で表す．任意の n 引数の $a \in \mathcal{D}$ と $t_1, \ldots, t_n \in \mathcal{T}(\mathcal{C})$ $(t_1, \ldots, t_n \in PVal(\mathcal{C}, \mathcal{V}_{fun}))$ に対して $a[t_1, \ldots, t_n]$ が簡約化可能であるとき，STRS R を**擬簡約化可能** (**強擬簡約化可能**) と呼び，$QR(R)$ $(SQR(R))$ で記す．

項の集合 $\mathcal{CS}$ が型 α の**被覆集合**であるとは，任意の $s \in \mathcal{T}_\alpha(\Sigma)$ に対して，$s \xrightarrow[R]{*} t\theta_c$ となる項 $t \in \mathcal{CS}$ と代入 θ_c が存在するような集合である．また，代入集合 $\mathcal{CSS}$ が変数集合 $X = \{x_1, \ldots, x_n\}$ 上の**被覆代入集合**とは，各変数 $x_1, \ldots, x_n$ に対してある被覆集合 $\mathcal{CS}_1, \ldots, \mathcal{CS}_n$ が存在して $\mathcal{CSS} = \{\{x_1 := c_1, \ldots, x_n := c_n\} \mid c_1 \in \mathcal{CS}_1, \ldots, c_n \in \mathcal{CS}_n\}$ となる代入集合である．

3. 帰納的定理

以下では文献 [4] に基づき帰納的定理に関する定義や定理を紹介する．TRS 上での帰納的定理を自然な形で STRS 上に拡張したものを原始帰納的定理と呼ぶ．原始帰納的定理は文脈に閉じていないため，原始帰納的定理の定義をそのまま帰納的定理の定義とするのは不適当である．STRS 上の帰納的定理は以下のように原始帰納的定理を用いて定義される．

定義 3.1 ([4]) 等式 $s \simeq t$ が STRS R の**原始帰納的定理**であるとは，任意の閉じた代入 θ_c と基底根文脈 $S_g[]$ に対し $S_g[s\theta_c] \xleftrightarrow[R]{*} S_g[t\theta_c]$ となることであり，等式 $s \simeq t$ が R の原始帰納的定理であることを $R \vdash_{\mathrm{pind}} s \simeq t$ と記す．また，等式集合 E について，E 中の全ての等式が R の原始帰納的定理であるとき，$R \vdash_{\mathrm{pind}} E$ と記す．

等式 $s \simeq t$ が STRS R の**帰納的定理**であるとは，ある n が存在して $R \vdash^n_{\mathrm{pind}} s \simeq t$ となることであり，等式 $s \simeq t$ が R の帰納的定理であることを $R \vdash_{\mathrm{ind}} s \simeq t$ と記す．ここで，$\vdash^n_{\mathrm{pind}}$ は以下のように帰納的に定義される．

- $R \vdash_{\mathrm{pind}} s \simeq t$ ならば $R \vdash^1_{\mathrm{pind}} s \simeq t$
- $R' = \{u = v \mid R \vdash^n_{\mathrm{pind}} u \simeq v\}$ に対し $R' \vdash_{\mathrm{pind}} s \simeq t$ ならば $R \vdash^{n+1}_{\mathrm{pind}} s \simeq t$

等式集合 E について，E 中の全ての等式が R の帰納的定理であるとき，$R \vdash_{\mathrm{ind}} E$ と記す．

一般には原始帰納的定理でない帰納的定理が存在する．しかし，STRS が適切な条件を満たしている場合には，原始帰納的定理と帰納的定理が一致する．

定理 3.2 ([4]) R を以下の全てを満たす STRS とする．

(a) $GSN(R)$ (b) $GCR(R)$ (c) $SQR(R)$
(d) $\mathcal{C}$ は 1 階の構造を持つ
(e) どの $l \to r \in R$ に対しても $l \notin \mathcal{T}_\tau(\mathcal{C}, \mathcal{V})$

このとき，任意の $s \simeq t$ で $R \vdash_{\mathrm{pind}} s \simeq t \iff R \vdash_{\mathrm{ind}} s \simeq t$.

例 3.3 $\mathcal{D} = \{foldl, foldr, app\}, \mathcal{C} = \{cons, nil, bar\}$ とし，以下の STRS R_1 を考える．

$$R_1 = \left\{ \begin{array}{rcl} foldl[f, xs, nil] & \to & xs \\ foldl[f, xs, cons[x, ys]] & \to & foldl[f, f[xs, x], ys] \\ foldr[f, xs, nil] & \to & xs \\ foldr[f, xs, cons[x, ys]] & \to & f[x, foldr[f, xs, ys]] \\ app[nil, ys] & \to & ys \\ app[cons[x, xs], ys] & \to & cons[x, app[xs, ys]] \end{array} \right.$$

$\tau(bar) = (List \to List) \to List$ とし，その他の関数記号および変数の型は通例に従うものとする．このとき，等式 $foldl[app, nil] \simeq foldr[app, nil]$ は原始帰納的定理であり，それゆえに帰納的定理でもある．一方，$bar[foldl[app, nil]] \simeq bar[foldr[app, nil]]$ は原始帰納的定理ではないが，帰納的定理ではある．

4. 書換え帰納法

本節では，STRS 上の書換え帰納法を与える．

書換え帰納法による帰納的定理の証明は，証明したい等式の集合 E に対して，対 $\langle E, \emptyset \rangle$ に各種推論規則を繰り返し適用して $\langle \emptyset, H \rangle$ の形へ導出することにより行う．例えば，我々が与える書換え帰納法を用いると，例 3.3 で与えた STRS R_1 において等式 $foldl[app, xs, nil] \simeq app[xs, nil]$ が帰納的定理であることを以下のような導出で示すことができる．

$$\langle \{foldl[app, xs, nil] \simeq app[xs, nil]\}, \emptyset \rangle$$
$$\vdash_{(Simplify)} \vdash_{(Expand)} \vdash_{(Simplify)} \vdash_{(Delete)}$$
$$\langle \emptyset, \{app[xs, nil] \to xs\} \rangle$$

また，我々が与える書換え帰納法では既存の TRS 上の書換え帰納法と同様に反証を行うこともできる．

4.1 書換え帰納法の基本原理

本節では書換え帰納法で利用可能な推論規則の持つべき性質を明らかにする．文献 [7] において書換え帰納法は抽象書換え系の枠組みで理論的に整理され，退行性と停止性が本質であると主張されている．

定義 4.1 ([7]) R_1, R_2 を抽象書換え系とする．このとき，任意の s, t について，$s \xrightarrow[R_2]{} t$ ならば $s \xrightarrow[R_1]{} u \xrightarrow[R_2]{*} v \xleftarrow[R_2]{*} t$ となる u, v が存在するとき，R_2 は R_1 に**退行する**という．R_2 が R_1 に退行していることを $RET(R_2, R_1)$ と記す．

定理 4.2 ([7]) R_1, R_2 を以下の性質全てを満たす抽象書換え系とする．このとき，$\xleftrightarrow[R_1]{*} = \xleftrightarrow[R_2]{*}$ が成立する．

$$(1) \xrightarrow[R_1]{} \subseteqq \xrightarrow[R_2]{+} \quad (2) SN(R_2) \quad (3) RET(R_2, R_1)$$

この定理を用いると，$\langle E, \emptyset \rangle \vdash^* \langle \emptyset, H \rangle$ という導出が存在した場合に，$SN(R \cup H)$ かつ $RET(R \cup H, R)$ かつ $\xleftrightarrow[E^{\uparrow}]{} \subseteqq \xleftrightarrow[R \cup H]{*}$ であれば，E が帰納的定理であると結論できる．

確かに，定理 4.2 は書換え帰納法の本質は捉えているが，証明の途中で導出された対が満たすべき状態が曖昧であるため，推論規則を適切に設計するためには不十分である．そこで，書換え帰納法で用いる推論規則を適切に設計するために，証明の途中で導出された対が満たすべき状態を考慮した上で各推論規則が満たすべき性質を洗い出す．具体的には，以下の 8 つの性質が，書換え帰納法による証明が正しく行われるために各推論規則が満たすべき性質であることを我々は明らかにする．

定義 4.3 R を STRS とする．等式集合の対から対への推論規則 $\langle E', H' \rangle \vdash \langle E'', H'' \rangle$ を *Proof* **規則**と呼ぶ．*Proof* 規則が**適正**であるとは以下が成立することである．

(RI1) $SN(R \cup H') \Rightarrow SN(R \cup H'')$

(RI2) $\xrightarrow[H']{} \subseteqq \xrightarrow[H'']{+}$

(RI3) $\mathcal{T}_{nfun}(\Sigma)$ 上で $\xrightarrow[H''-H']{} \subseteqq \xrightarrow[R]{} \xrightarrow[R \cup H'']{*} \nleftrightarrow_{E''^{\uparrow}} \xleftarrow[R \cup H'']{*}$

(RI4) H' が非関数型 $\Rightarrow H''$ が非関数型

(RI5) $\mathcal{T}_{nfun}(\Sigma)$ 上で $\nleftrightarrow_{E'^{\uparrow}} \subseteqq \xrightarrow[R \cup H'']{*} \nleftrightarrow_{E''^{\uparrow}} \xleftarrow[R \cup H'']{*}$

(RI6) $R \vdash_{\mathrm{pind}} E' \cup H'$ かつ H' が非関数型 $\Rightarrow R \vdash_{\mathrm{pind}} E'' \cup H''$

(RI7) どの等式集合 L に対しても $\langle E' \cup L, H' \rangle \vdash \langle E'' \cup L, H'' \rangle$

等式集合の対から定数 Disproof への推論規則 $\langle E', H' \rangle \vdash$ Disproof を *Disproof* **規則**と呼ぶ．*Disproof* 規則が適正であるとは，以下が成立することである．

(RI8) $R \nvdash_{\mathrm{pind}} E' \cup H'$

ここで，特別に図 1 で Postulate 規則を与えておく．これは，文献 [6] で提案された推論規則であり，補題を追加するために必要であるが，性質 (RI6) を満たさないため適正ではない．

(Postulate)

$$\frac{\langle E, H \rangle}{\langle E \cup E', H \rangle}$$

図 1 Postulate 規則

なお，定義 4.3 の性質に関して，以下のことがいえる．これは，後に記す健全性証明から見て取れる．

- 性質 (RI1)(RI2)(RI3)(RI4)(RI5) は帰納的定理の証明の際にしか用いていない．よって，帰納的定理の非証明のみを行う際には不必要．
- 性質 (RI6) は帰納的定理の非証明の際にしか用いていない．よって，帰納的定理の証明のみを行う際には不必要．
- 性質 (RI7) は Postulate 規則を用いない場合には不必要．

以下では，適正な推論規則と Postulate 規則のみを用いて導出を行った場合に，書換え帰納法による証明が健全であることを示す．

補題 4.4 適用した規則が全て適正な $\langle E, \emptyset \rangle \vdash^n \langle E_n, H_n \rangle$ の導出が存在するならば，以下が成立する．

(i) $R \vdash_{\mathrm{pind}} E_n \cup H_n \Rightarrow R \vdash_{\mathrm{pind}} E$

(ii) $\mathcal{T}_{nfun}(\Sigma)$ 上で $\xrightarrow[R \cup H_n]{} \subseteqq \xrightarrow[R]{} \xrightarrow[R \cup H_n]{*} \nleftrightarrow_{E_n^{\uparrow}} \xleftarrow[R \cup H_n]{*}$

証明. n に関する帰納法で容易に示せる． □

定理 4.5 R を停止性を持つ STRS とする．適用する全ての推論規則は Postulate 規則か適正な規則であるとする．

(i) $\langle E, \emptyset \rangle \vdash^* \langle \emptyset, H \rangle$ の導出が存在するとし，この間に Postulate 規則で追加した等式全てからなる集合を L とする．このとき $R \vdash_{\mathrm{pind}} E \cup L$．更に $R \vdash_{\mathrm{ind}} E \cup L$ も成立．

(ii) $\langle E, \emptyset \rangle \vdash^*$ Disproof の導出が存在するとし，この間に Postulate 規則で追加した等式全てからなる集合を L とする．このとき $R \nvdash_{\mathrm{pind}} E \cup L$．また定理 3.2 の条件を満たすならば $R \nvdash_{\mathrm{ind}} E \cup L$．

(Simplify)
$$\frac{\langle E \uplus \{s \simeq t\}, H\rangle}{\langle E \cup \{s' \simeq t\}, H\rangle} \quad s \xrightarrow[R \cup H]{*} s'$$

(Expand)
$$\frac{\langle E \uplus \{s \simeq t\}, H\rangle}{\langle E \cup Expd_p(s,t), H \cup \{s \to t\}^{\uparrow}\rangle}$$
p が s における
R 完全な出現かつ
$SN(R \cup H \cup \{s \to t\}^{\uparrow})$

(Delete Tautology)
$$\frac{\langle E \uplus \{s \simeq t\}, H\rangle}{\langle E, H\rangle}$$
$\exists H' \subseteq H$
$(R \cup H' \vdash_{\mathrm{pind}} s \simeq t$ かつ
$GCR(R \cup H'))$

(CaseSplit)
$$\frac{\langle E \uplus \{s \simeq t\}, H\rangle}{\langle E \cup \{s\sigma \simeq t\sigma \mid \sigma \in \mathcal{CSS}\}, H\rangle}$$
$\mathcal{CSS}$が
$X(\subseteq Var(s) \cup Var(t))$ 上の
空でない被覆代入集合

(Delete)
$$\frac{\langle E \uplus \{s \simeq s\}, H\rangle}{\langle E, H\rangle}$$

(C-Postulate)
$$\frac{\langle E, H\rangle}{\langle E \cup \{s \simeq t\}, H\rangle} \quad R \cup E^{\uparrow} \cup H \vdash_{\mathrm{pind}} s \simeq t$$

(S-Postulate)
$$\frac{\langle E, H\rangle}{\langle E, H \cup H'\rangle}$$
$\exists L. \langle L, \emptyset\rangle \vdash^* \langle \emptyset, H'\rangle$ かつ
$SN(R \cup H \cup H')$

(Subsume)
$$\frac{\langle E \uplus \{s \simeq t\}, H\rangle}{\langle E, H\rangle}$$
$\exists s' \simeq t' \in R \cup E \cup H.$
$\exists \sigma. s'\sigma \equiv s \wedge t'\sigma \equiv t$

(Extend)
$$\frac{\langle E \uplus \{s \simeq t\}, H\rangle}{\langle E \cup \{s[z] \simeq t[z]\}, H\rangle} \quad \tau(s) \in \mathcal{S}_{fun}\text{かつ } z \text{ は新しい変数}$$

図 2 *Proof* 規則

証明. (i) Postulate 規則以外の規則は全て適正であるので，性質 (RI7) より Postulate 規則を用いない導出 $\langle E \cup L, \emptyset\rangle \vdash^* \langle \emptyset, H'\rangle$ が存在する．このとき R の停止性と性質 (RI1) から $R \cup H'$ は停止性を持つ．性質 (RI2) と $R \cup H'$ の停止性と補題 4.4(ii) より $\xrightarrow[R]{}$ と $\xrightarrow[R \cup H']{}$ は定理 4.2 の条件を満たすので $\mathcal{T}_{nfun}(\Sigma)$ 上で $\xleftrightarrow[R]{*} = \xleftrightarrow[R \cup H']{*}$．よって $R \vdash_{\mathrm{pind}} H'$．よって，補題 4.4(i) より $R \vdash_{\mathrm{pind}} E \cup L$ を得る．また $\vdash_{\mathrm{ind}}$ の定義より $R \vdash_{\mathrm{ind}} E \cup L$ を得る．
(ii) 上と同様に Postulate 規則を用いない導出 $\langle E \cup L, \emptyset\rangle \vdash^* \langle E', H'\rangle \vdash$ Disproof が存在する．$R \vdash_{\mathrm{pind}} E \cup L$ を仮定すると性質 (RI6) より $R \vdash_{\mathrm{pind}} E' \cup H'$ となる．一方，性質 (RI8) より $R \nvdash_{\mathrm{pind}} E' \cup H'$ となるので矛盾．よって $R \nvdash_{\mathrm{pind}} E \cup L$．さらに，定理 3.2 の条件を満たすなら $R \nvdash_{\mathrm{ind}} E \cup L$ も成立． □

4.2 推論規則

本節では，STRS 上での書換え帰納法で使用可能な推論規則を与える．

4.2.1 *Proof* 規則

適正な *Proof* 規則を図 2 で与える．Simplify，Delete，Expand，Postulate，C-Postulate の各規則は文献 [6] で，Delete Tautology，CaseSplit，Subsume の各規則は文献 [2] で，S-Postlate 規則は文献 [8] で提案された推論規則を参考にしている．また，Extend 規則は文献 [5] の拡張形の定義を参考にした推論規則である．

なお，S-Postulate 規則中での導出 $\vdash^*$ は適正な規則か Postulate 規則のみを用いた導出とする．ここで，Expand 規則中で用いる R 完全の概念と集合 $Expd_p(s,t)$ の定義を次で与える．

定義 4.6 項 t 中の位置 p が R 完全な出現であるとは，任意の閉じた正規形代入 θ_v に対して $t|_p\theta_v$ が R で書換え可能であることである．$Expd_p(s,t)$ を $\{s[r]_p\sigma \simeq t\sigma \mid l \to r \in R,\ s|_p$ と l は単一化可能$, \sigma = \mathrm{mgu}(s|_p, l)\}$ で定義する．

定理 4.7 図 2 の推論規則は全て適正である．

さらに，$GCR(R)$ ならば図 3 の Decompose 規則も適正となる．なお，この規則は文献 [2] で提案された推論規則を参考にしている．

(Decompose)
$$\frac{\langle E \uplus \{f[s_1, \ldots, s_n] \simeq f[t_1, \ldots, t_n]\}, H\rangle}{\langle E \cup \{s_1 \simeq t_1, \ldots, s_n \simeq t_n\}, H\rangle} \quad \begin{array}{l} f \in \mathcal{C} \text{ かつ} \\ \tau(s_i) \in \mathcal{S}_{nfun} \end{array}$$

図 3 Decompose 規則

定理 4.8 $GCR(R)$ ならば，Decompose 規則は適正である．

Proof 規則の効率的な実現のため，以下では前記した推論規則の特別な場合や，適用条件の判定が容易になる場合などを記す．

a) Expand 規則について

擬簡約化可能な STRS R に対しては，より効率的に R 完全な出現を発見できる．具体的には，以下の容易に判定できる 3 条件が成立すれば項 t の位置 p が R 完全になっている．

(1)$\tau(t|_p) \in \mathcal{S}_{nfun}$ (2)$root(t|_p) \in \mathcal{D}$
(3)$t|_p$ の真部分項が全て構成子項

b) C-Postulate 規則について

C-Postulate 規則は Postulate 規則で追加する補題に $R \cup E^{\uparrow} \cup H \vdash_{\mathrm{pind}} s \simeq t$ という条件を加えることで定義されており，この条件のおかげで，C-Postulate 規則は適正な規則になっ

ている．$R \cup E^{\uparrow} \cup H$ における原始帰納的定理を与える方法の一つとして，危険対を用いることができる．$R \cup E^{\uparrow} \cup H$ の危険対は明らかに $R \cup E^{\uparrow} \cup H$ の原始帰納的定理であるので，C-Postulate 規則の特別な場合として図 4 の CP-Postulate 規則を与えることができる．

(CP-Postulate)

$$\frac{\langle E, H\rangle}{\langle E \cup \{s \simeq t\}, H\rangle} \quad s \simeq t \in CP(R \cup E^{\uparrow} \cup H)$$

図 4　CP-Postulate 規則

c）　Decompose 規則について

$\forall i.\tau(s_i) \in \mathcal{S}_{nfun}$ は性質 (RI5) を成立させるためだけに必要である．よって，反証のみを行う場合は除外してもよい適用条件となる．そのため，反証のみを行う際には図 5 の D-Decompose 規則を用いれば適用条件が少なく効率的に反証が行える．

(D-Decompose)

$$\frac{\langle E \uplus \{f[s_1,\ldots,s_n] \simeq f[t_1,\ldots,t_n]\}, H\rangle}{\langle E \cup \{s_1 \simeq t_1,\ldots,s_n \simeq t_n\}, H\rangle} \quad f \in \mathcal{C}$$

図 5　D-Decompose 規則

系 4.9　$GCR(R)$ であるとき，D-Decompose 規則は性質 (RI6) と (RI7) を満たす．

4.2.2　*Disproof* 規則

図 6 に *Disproof* 規則一覧を記す．*Disproof* 規則は全て文献 [2] で提案された推論規則を参考にしている．なお，Diprove4 規則の θ_c は閉じた代入とする．

(Disprove1)

$$\frac{\langle E \uplus \{s \simeq x\}, H\rangle}{\text{Disproof}} \quad x \notin Var(s) \text{ かつ } \tau(s) \in \mathcal{S}_{nfun}$$

(Disprove2)

$$\frac{\langle E \uplus \{f[s_1,\ldots,s_n] \simeq x\}, H\rangle}{\text{Disproof}} \quad \begin{array}{l} f \in \mathcal{C} \text{ かつ } x \in \mathcal{V}_{nfun} \text{ かつ} \\ \exists t \in GNF_{\tau(x)}(R). \\ root(t) \neq f \end{array}$$

(Disprove3)

$$\frac{\langle E \uplus \{f[s_1,\ldots,s_n] \simeq g[t_1,\ldots,t_m]\}, H\rangle}{\text{Disproof}} \quad \begin{array}{l} f, g \in \mathcal{C} \text{ かつ} \\ f \neq g \end{array}$$

(Disprove4)

$$\frac{\langle E \uplus \{s \simeq t\}, H\rangle}{\text{Disproof}} \quad \begin{array}{l} \exists \theta_c . s\theta_c\downarrow_R \neq t\theta_c\downarrow_R \text{ かつ} \\ \tau(s) \in \mathcal{S}_{nfun} \end{array}$$

図 6　*Disproof* 規則

定理 4.10　R を以下の 3 条件を満たす STRS とする．このとき，図 6 の 4 つの推論規則は全て適正である．

- $GWN(R)$ • $GCR(R)$ • $\forall \alpha \in \mathcal{S}_{nfun}.|GNF_\alpha(R)| \geqq 2$

なお，本定理で挙げた適用条件について以下のことが言える．

- $GWN(R)$ は Disprove4 規則を除く 3 つの *Disproof* 規則を用いるのに必要．
- $GCR(R)$ は全ての *Disproof* 規則を用いるのに必要．
- $\forall \alpha \in \mathcal{S}_{nfun}.|GNF_\alpha(R)| \geqq 2$ は Disprove1 規則を用いる際にのみ必要で，他の *Disproof* 規則には不必要．

4.3　書換え帰納法による証明例

高階関数を含む等式の書換え帰納法による証明例を記す．

例 4.11　例 3.3 の R_1 を考える．このとき，以下の等式 (1) が帰納的定理であることを示す．

$$foldl[app, nil] \simeq foldr[app, nil] \tag{1}$$

まず，以下の対を与える．

$$\langle\{foldl[app, nil] \simeq foldr[app, nil]\}, \emptyset\rangle$$

入力が関数型の等式であることから Extend 規則を適用して以下の対が得られる．

$$\langle\{foldl[app, nil, zs] \simeq foldr[app, nil, zs]\}, \emptyset\rangle$$

等式 $foldl[app, nil, zs] \simeq foldr[app, nil, zs]$ の左辺の位置 ε に Expand 規則を適用する．このとき，

$Expd_\varepsilon(foldl[app, nil, zs], foldr[app, nil, zs]) =$

$$\left\{\begin{array}{l} Nil \simeq foldr[app, nil, nil], \\ foldl[app, (app[nil, x]), xs] \\ \qquad \simeq foldr[app, nil, cons[x, xs]] \end{array}\right\}$$

であるので，以下の E_1 と H_1 の対 $\langle E_1, H_1\rangle$ が導出される．

$$\begin{array}{lcl} E_1 & = & Expd_\varepsilon(foldl[app, nil, zs], foldr[app, nil, zs]) \\ H_1 & = & \{foldl[app, nil, zs] \to foldr[app, nil, zs]\} \end{array}$$

この対に Simplify 規則を適用すると，

$$E_2 = \left\{\begin{array}{c} nil \simeq nil \\ foldl[app, x, xs] \simeq app[x, foldr[app, nil, xs]] \end{array}\right\}$$

と H_1 の対 $\langle E_2, H_1\rangle$ が導出される．Delete 規則により等式 $nil \simeq nil$ が消え，以下の対が得られる．

$$\langle\{foldl[app, x, xs] \simeq app[x, foldr[app, nil, xs]]\}, H_1\rangle$$

以後も同様に導出を行っていくが，この証明を成功させるためには，以下の等式を補題として追加する必要がある．

$$app[xs, app[ys, zs]] \simeq app[app[xs, ys], zs]$$

これは導出中に Postulate 規則を用いて追加することができる．補題の追加により，最終的に $\langle \emptyset, H\rangle$ という形の対を導出す

ることができ，等式 (1) が原始帰納的定理であると判定される．よって，等式 (1) が帰納的定理であることが示される．

例 4.12 例 4.11 と同様に例 3.3 の STRS R_1 を考える．ただし，$\mathcal{C} = \{cons, nil\}$ とする．このとき，以下の等式 (2) が帰納的定理でないことを示す．

$$foldl[app] \simeq foldr[app] \tag{2}$$

例 4.11 と同様に Extend，Expand，Simplify，Delete の各規則を用いて導出を行っていくと，

$$cons[x, app[xs, cons[y, ys]]] \simeq cons[y, app[ys, cons[x, xs]]]$$

を E' 中に含む対 $\langle E', H' \rangle$ を得る．この対には Decompose 規則が適用でき，

$$x \simeq y$$
$$app[xs, cons[y, ys]] \simeq app[ys, cons[x, xs]]$$

を E'' 中に含む対 $\langle E'', H' \rangle$ が導出される．このとき，等式 $x \simeq y$ は Disprove1 規則の適用条件を満たすため，定数 Disproof が導出される．よって，等式 (2) は原始帰納的定理でないと判定される．さらに R_1 は定理 3.2 の 5 つの条件を満たしているため，等式 (2) が帰納的定理でないことも示される．

この例では，例 3.3 と同様に $\mathcal{C}$ に関数記号 bar を含んでいたとしても，bar は R_1 にも求めたい等式中にも含まれない記号であることから，書換え帰納法の証明には直接影響しない．しかし，$\tau(bar) = (List \to List) \to List$ であることから定理 3.2 の条件 (d) を満たさないため等式 (2) が帰納的定理でないことは示されなくなる．

4.4 原始帰納的定理でない帰納的定理の証明

先に挙げた書換え帰納法による帰納的定理の証明は，本質的には原始帰納的定理を証明する手法である．そのため，帰納的定理であるにも関わらず原始帰納的定理でないものは証明できない．以下では，先に挙げた書換え帰納法を用いてより多くの帰納的定理を証明する手法を提案する．

定義 4.13 導出 $\langle E, H \rangle \vdash^* \langle E', H' \rangle$ について，この導出が STRS R の下での導出であることを $\langle E, H \rangle_R \vdash^* \langle E', H' \rangle_R$ で記す．文脈から R が明らかな場合は R を省略する．

証明中で用いる STRS R に $R \vdash^n_{\mathrm{pind}} E$ となる等式集合 E を追加することによって，$\vdash^{n+1}_{\mathrm{pind}}$ を求めることができ，原始帰納的定理でない帰納的定理を証明することが可能となる．

定理 4.14 $R \vdash^n_{\mathrm{pind}} E$ となる E が存在するとする．また，$R' \subseteq R \cup E$ とする．このとき，$SN(R')$ かつ任意の等式 E' について導出 $\langle E', \emptyset \rangle_{R'} \vdash^* \langle \emptyset, H' \rangle_{R'}$ が存在するならば，$R \vdash^{n+1}_{\mathrm{pind}} E'$．更に $R \vdash_{\mathrm{ind}} E'$ も成立．

例 4.15 例 3.3 の STRS R_1 を考える．このとき，

$$bar[foldl[app, nil]] \simeq bar[foldr[app, nil]] \tag{3}$$

が帰納的定理であることを示したい．しかし，等式 (3) は原始帰納的定理でないため，書換え帰納法を単純に行っただけでは帰納的定理であることは示せない．ここで次の等式を考える．

$$foldl[app, nil] \simeq foldr[app, nil]$$

この等式は例 4.11 で原始帰納的定理であることが示されている．$R'_1 = R_1 \cup \{foldl[app, nil]] \to foldr[app, nil]\}$ とすると $SN(R'_1)$ は容易に示される．さらに，書換え帰納法により

$$R'_1 \vdash_{\mathrm{pind}} bar[foldl[app, nil]] \simeq bar[foldr[app, nil]]$$

も容易に示される．定理 4.14 よりこれは

$$R_1 \vdash^2_{\mathrm{pind}} bar[foldl[app, nil]] \simeq bar[foldr[app, nil]]$$

であるため，等式 (3) が帰納的定理であることが示される．

5. 結 論

書換え帰納法の推論規則が持つべき性質を明らかにした上で，STRS 上での書換え帰納法を構築した．本研究の成果により，高階関数を含む等式に対しても書換え帰納法を用いた帰納的定理の証明が可能となった．

本研究では STRS における書換え帰納法の枠組みを定めたに過ぎない．よって，証明を効率的に行うための推論規則の適用戦略の考案が今後の課題である．また，証明の際には補題の追加が必要となる場合が多い．補題生成法に関しては TRS 上の書換え帰納法でも広く研究されてきたが，STRS 上でも補題生成法に関する議論が必要であり，今後の課題の 1 つとして挙げられる．

謝辞 本研究は，一部，文部科学省科学研究費#20300010, #20500008, #21700011 の助成を受けたものである．

文 献

[1] A. Bouhoula, E. Kounalis, and M. Rusinowitch: Automated Mathematical Induction, *Journal of Logic and Computation*,Vol. 5,No. 5, pp 631–668, 1995.
[2] A. Bouhoula: Automated Theorem Proving by Test Set Induction, *Journal of Symbolic Computation*, Vol. 23, No.1, pp.47–77, 1997.
[3] K. Kusakari: On Proving Termination of Term Rewriting System with Higher-Order Variables, IPSJ Transactions on Programming, Vol.42, No.SIG7(PRO 11), pp.35–45, 2001.
[4] K. Kusakari, M. Sakai, T. Sakabe: Primitive Inductive Theorems Bridge Implicit Induction Methods and Inductive Theorems in Higher-Order Rewriting, IEICE Transactions on Information and Systems, Vol.E88-D, No.12, pp.2715–2726, 2005.
[5] K. Kusakari, M. Sakai: Enhancing Dependency Pair Method using Strong Computability in Simply-Typed Term Rewriting, Applicable Algebra in Engineering, Communication and Computing, Vol.18, No.5, pp.407–431, 2007.
[6] U. Reddy: Term Rewriting Induction, LNCS. Vol.449, pp.162–177, 1990.
[7] 小池広高，外山芳人：潜在帰納法と書換え帰納法の比較，コンピュータソフトウェア，Vol.17，No.6，pp.1–12，2000.
[8] 中林直生，西田直樹，草刈圭一朗，坂部俊樹，酒井正彦：制約付き項書換え系の書換え帰納法における補題等式の自動生成法，コンピュータソフトウェア，Vol.28，No.1，pp.173–189，2011.