書き換え帰納法を利用した帰納的定理証明の補題生成法

加藤 裕人　青戸 等人

項書き換えシステムの帰納的定理の自動証明法においては，適当な補題の発見が証明成功の鍵となることが多い．このため，発散鑑定法や健全一般化法といった補題生成法が提案されている．しかし，これらは証明過程で局所的に適切な補題を生成する手法であり，証明過程に依存した限定的な補題しか生成されない．本報告では，特定の証明過程にのみ役立つ補題だけではなく，他の様々な自動証明手法において役立つような補題を発見するための補題生成法を提案する．提案手法のアイデアは，項書き換えシステムの自動証明法の一つである書き換え帰納法を柔軟に実行し，そこから得られる様々な証明発散過程から補題を抽出することである．補題生成手続きの実装および補題生成実験についても報告する．

1　はじめに

等式論理に基づく計算モデルとして項書き換えシステムがある．項書き換えシステムをプログラムとしてみなすときに成立する等式を帰納的定理とよぶ．帰納的定理の自動証明法として，Reddy により提案された書き換え帰納法が知られている [4]．

書き換え帰納法では，等式の集合と項書き換え規則の集合の対に関する導出を行いながら，証明を行う．証明過程において，推論規則の導出が無限に繰り返される場合には手続きは停止せず (発散)，証明に失敗する．このような場合に，適切な補題を一緒に証明することで，発散が抑制されて証明が成功することがある．

書き換え帰納法が発散して失敗する場合に，その発散系列から補題を発見する方法として発散鑑定法がある [6]．発散鑑定法は明示的帰納法のヒューリスティックであるリッピング法 [2] を書き換え帰納法の補題生成に応用したものである．しかし，どのような場合でも必ず補題が発見できるわけではない．また，書き換え帰納法や発散鑑定法の手続きは，規則を適用する等式の選び方や，適用する推論規則等に任意性があるため，非決定的である．このため，適切な補題を導出する方法は自明ではない．

一方，補題の発見は書き換え帰納法に限らず，様々な自動証明法において，証明成功の鍵となることが多い．このため，補題発見の成功率を高めることができれば，より強力な定理自動証明システムの実現につながると考えられる．定理自動証明システムに有用な補題を発見するシステムとして HipSpec が提案されている [3]．HipSpec では，より単純な等式から始まってランダムに等式を生成し，その等式が証明に成功するとき，それを補題として提案する．

本報告では，書き換え帰納法を利用した補題生成手法を提案する．書き換え帰納法において，発散系列を抽出するもととなる導出の仮定部が大きく生成されるように，本来，定理証明を目的としている書き換え帰納法のために必要となっていた戦略や制約を緩和する．さらに，生成された仮定部から発散系列を抽出する手法，発散系列から補題を生成する手法を提案する．また，提案手法の実装および実験結果について報告する．

Lemma Generation based on Rewriting Induction.

Hiroto Kato, Takahito Aoto, 新潟大学大学院, 自然科学研究科.

Simplify

$$\frac{\langle E \uplus \{s \approx t\}, H\rangle}{\langle E \cup \{s' \approx t\}, H\rangle} s \to_{\mathcal{R}\cup H} s'$$

Delete

$$\frac{\langle E \uplus \{s \approx s\}, H\rangle}{\langle E, H\rangle}$$

Expand

$$\frac{\langle E \uplus \{s \approx t\}, H\rangle}{\langle E \cup \mathrm{Expd}_u(s,t), H \cup \{s \to t\}\rangle} u \in \mathcal{B}(s), s > t$$

図 1　書き換え帰納法の推論規則

2　準備

2.1　項書き換えシステム

本報告で用いる定義および記法を紹介する．

2 項関係 $\to$ の**反射推移閉包**を $\xrightarrow{*}$，**等価閉包**を $\overset{*}{\leftrightarrow}$ と記す．**関数記号集合**および**変数集合**をそれぞれ $\mathcal{F}, \mathcal{V}$ で表す．$\mathcal{F}, \mathcal{V}$ 上の**項**の集合を $\mathcal{T}(\mathcal{F}, \mathcal{V})$ で表す．項 t の**根記号**とは $t \in \mathcal{V}$ のときは t，$t = f(t_1, \ldots, t_n)(f \in \mathcal{F})$ のときは，f である．$\mathcal{V}(t)$ は項 t に出現する変数全体の集合を表す．$\mathcal{V}(t) = \emptyset$ を満たすとき，項 t を**基底項**とよぶ．$C[t]$ は**文脈** C のホールを項 t で置き換えて得られる項を表す．項 u が項 t の**部分項**であるとは，$t = C[u]$ なる文脈 C があるときをいう．

関数 $\sigma : \mathcal{V} \to \mathcal{T}(\mathcal{F}, \mathcal{V})$ で，集合 $\{x \mid \sigma(x) \neq x\}$ が有限であるものを**代入**とよぶ．代入 σ の定義域を $\mathcal{T}(\mathcal{F}, \mathcal{V})$ に拡張した準同型拡張も同じく σ で表し，$\sigma(t)$ を $t\sigma$ と記す．$t\sigma$ が基底項となるとき，これを t の**基底具体化**とよぶ．等式 $s \approx t$ の基底具体化も同様に定める．

2 つの項 l, r が，$l \notin \mathcal{V}, \mathcal{V}(r) \subseteq \mathcal{V}(l)$ を満たすとき，$l \to r$ を**書き換え規則**とよぶ．書き換え規則の有限集合を**項書き換えシステム**とよぶ．

項書き換えシステムに含まれる書き換え規則の左辺の根記号となる関数記号を**定義記号**という．定義記号集合を $\mathcal{D}$ と記す．**構成子記号集合**を $\mathcal{C} = \mathcal{F} \setminus \mathcal{D}$ と定義する．定義記号を含まない項を**構成子項**とよぶ．構成子項の集合を $\mathcal{T}(\mathcal{C}, \mathcal{V})$ と記す．定義記号 f と構成子項 $c_1, \ldots, c_n$ からなる項 $f(c_1, \ldots, c_n)$ を**基本項**とよぶ．基本項の集合を $\mathcal{B}(\mathcal{D}, \mathcal{C}, \mathcal{V})$ と記す．項 s の**基本部分項**の集合を $\mathcal{B}(s)$ と記す．任意の $l \to r \in \mathcal{R}$ について，$l \in \mathcal{B}(\mathcal{D}, \mathcal{C}, \mathcal{V})$ であるとき，$\mathcal{R}$ を**構成子システム**とよぶ．

$\mathcal{R}$ を項書き換えシステムとする．書き換え規則 $l \to r \in \mathcal{R}$，代入 σ，文脈 C が存在して，$s = C[l\sigma]$ かつ $t = C[r\sigma]$ となるとき，$s \to_{\mathcal{R}} t$ と記す．代入および文脈に閉じている項上の整礎な半順序を**簡約順序**とよぶ．

2.2　帰納的定理と書き換え帰納法

等式 $s \approx t$ が，項書き換えシステム $\mathcal{R}$ の**帰納的定理**であるとは，$s \approx t$ の任意の基底具体化 $s\theta_g \approx t\theta_g$ について，$s\theta_g \overset{*}{\leftrightarrow}_{\mathcal{R}} t\theta_g$ が成立するときをいう．以下では，等式 $s \approx t$ と $t \approx s$ は同一視する．また，等式集合 E のすべての等式が帰納的定理であるとき，E は $\mathcal{R}$ の帰納的定理であるという．

例 2.1 (帰納的定理) 自然数上の加算を項書き換えシステム $\mathcal{R}$ で与える．ここで，自然数 $0, 1, 2, \ldots$ は $\mathsf{0}, \mathsf{s}(\mathsf{0}), \mathsf{s}(\mathsf{s}(\mathsf{0})), \ldots$ と表現する．

$$\mathcal{R} = \begin{cases} +(\mathsf{0}, x) \to x \\ +(\mathsf{s}(x), y) \to \mathsf{s}(+(x, y)) \end{cases}$$

このとき，$+(+(x, y), z) \approx +(x, +(y, z))$ は，任意の基底項 s_g, t_g, u_g について $+(+(s_g, t_g), u_g) \overset{*}{\leftrightarrow}_{\mathcal{R}} +(s_g, +(t_g, u_g))$ が成立する．よって，$\mathcal{R}$ の帰納的定理である．

帰納的定理の自動証明法として**書き換え帰納法**が知られている [4]．書き換え帰納法では，等式集合 E(**等式部**とよぶ) と項書き換え規則集合 H(**仮定部**とよぶ) の対 $\langle E, H\rangle$ に関する導出を行いながら証明を行う．図 1 に書き換え帰納法の推論規則を示す．ここで，$\mathcal{R}$ は項書き換えシステム，$\uplus$ は直和，$>$ は簡約順序，$\mathrm{mgu}(s, t)$ は s と t の最汎単一化子を表す．Expd は

以下のように定義される操作である．

$$\mathrm{Expd}_u(s,t) = \{C[r]\sigma \approx t\sigma \mid s = C[u], \\ \sigma = \mathrm{mgu}(u,l), l \to r \in \mathcal{R}\}$$

$\langle E, H\rangle$ から推論規則を適用して $\langle E', H'\rangle$ が得られることを $\langle E, H\rangle \rightsquigarrow \langle E', H'\rangle$ と記す．

等式集合 E が $\mathcal{R}$ の帰納的定理であることを証明するには，$\mathcal{R} \subseteq >$ なる簡約順序を与え，$\langle E, \emptyset\rangle$ から導出を始め，推論規則の適用を繰り返す．最終的に $\langle E, \emptyset\rangle \overset{*}{\rightsquigarrow} \langle \emptyset, H\rangle$ が導出されるとき，証明成功となる．しかし，等式部が空にならないまま無限に導出を繰り返して手続きが停止しないことがある．この場合の手続きが**発散する**という．

例 2.2 (書き換え帰納法の証明) 例 2.1 の項書き換えシステム $\mathcal{R}$ を与える．このとき，以下の等式集合 E が $\mathcal{R}$ の帰納的定理であることを書き換え帰納法により示す．

$$E = \left\{ \; +(+(x,y),z) \approx +(x,+(y,z)) \right\}$$

以下で，$>$ を $+ > \mathsf{s} > 0$ に基づく辞書式経路順序としたときの書き換え帰納法の証明過程を示す．

$$\left\langle \left\{ +(+(x,y),z) \approx +(x,+(y,z)) \right\}, \{\} \right\rangle \\ \overset{*}{\rightsquigarrow} \left\langle \begin{Bmatrix} +(y,z) \approx +(0,+(y,z)) \\ +(\mathsf{s}(+(x,y)),z) \approx +(\mathsf{s}(x),+(y,z)) \end{Bmatrix}, \left\{ +(+(x,y),z) \to +(x,+(y,z)) \right\} \right\rangle \\ \overset{*}{\rightsquigarrow} \left\langle \{\}, \left\{ +(+(x,y),z) \to +(x,+(y,z)) \right\} \right\rangle$$

$\langle \emptyset, H\rangle$ が導出されたため，証明成功となる．

例 2.3 (書き換え帰納法の発散) リスト結合，リスト反転を項書き換えシステム $\mathcal{R}$ で与える ($x :: []$ を $[x]$ で表す)．

$$\mathcal{R} = \begin{Bmatrix} @([], ys) \to ys \\ @(x :: xs, ys) \to x :: @(xs, ys) \\ \mathsf{rev}([]) \to [] \\ \mathsf{rev}(x :: xs) \to @(\mathsf{rev}(xs), [x]) \end{Bmatrix}$$

$$E = \left\{ \; \mathsf{rev}(\mathsf{rev}(@(xs,ys))) \approx @(xs,ys) \right\}$$

書き換え帰納法の実行過程を以下に示す．

$$\left\langle \left\{ \mathsf{rev}(\mathsf{rev}(@(xs,ys))) \approx @(xs,ys) \right\}, \{\} \right\rangle \\ \overset{*}{\rightsquigarrow} \left\langle \begin{Bmatrix} \mathsf{rev}(@(\mathsf{rev}(@(zs,ys)),[z])) \approx z :: @(zs,ys) \\ \mathsf{rev}(\mathsf{rev}(@(xs,ys))) \to @(xs,ys) \end{Bmatrix}, \right\rangle \\ \overset{*}{\rightsquigarrow} \left\langle \begin{Bmatrix} \mathsf{rev}(@(@(\mathsf{rev}(@(vs,ys)),[v]),[z])) \approx z :: (v :: @(vs,ys)) \\ \mathsf{rev}(@(\mathsf{rev}(@(zs,ys)),[z])) \to z :: @(zs,ys) \\ \mathsf{rev}(\mathsf{rev}(@(xs,ys))) \to @(xs,ys) \end{Bmatrix} \right\rangle \\ \overset{*}{\rightsquigarrow} \cdots$$

このように書き換え帰納法の手続きを繰り返し適用しても等式部が空にならず，発散する．このように発散した場合，適当な補題を追加することで発散を抑制し，証明に成功する場合がある．

2.3 発散鑑定法による補題生成

書き換え帰納法が発散する場合，同じパターンにより拡大する系列 (**発散系列**) が観測されることがある．発散鑑定法は，書き換え帰納法が発散した場合に，その発散系列を収束させるような書き換え規則を構成する．以下で，発散鑑定法による補題生成を簡単な例で説明する．

例 2.4 (発散鑑定法による補題生成例) 例 2.3 の $\mathcal{R}$ を与える．以下の等式集合 E を書き換え帰納法で証明することを考える．

$$E = \left\{ \; \mathsf{rev}(\mathsf{rev}(@(xs,ys))) \approx @(xs,ys) \right\}$$

このとき，E は $\mathcal{R}$ の帰納的定理である．しかし，書き換え帰納法は発散し，以下の発散系列が出現する．

$$\mathsf{rev}(\mathsf{rev}(@(xs,ys))) \to @(xs,ys) \quad (1)$$
$$\mathsf{rev}(@(\mathsf{rev}(@(xs,ys)),[x])) \to x :: @(xs,ys) \quad (2)$$
$$\vdots$$

(1) と (2) の書き換え規則の差分は以下のように表せる．

$$\mathsf{rev}(\boxed{@(\underline{\mathsf{rev}(@(xs,ys))},[x])}) \to \boxed{x :: \underline{@(xs,ys)}} \quad (3)$$

ここで差分 (3) の枠から下線部を取り除いた部分が書き換え規則 (1) と (2) の差分となっており，差分を取り除くと変数の違いを除いて書き換え規則 (1) が得られることに注意する．このような差分を求める手続きとして**差分照合**が知られている [1]．

G-Expand

$$\frac{\langle E \uplus \{s_i \approx t_i\}_i, H\rangle}{\langle E \cup \bigcup_i \bigcup_{u\in\mathcal{B}(s_i)} Expd_u(s_i,t_i), H \cup \{s_i \to t_i\}_i\rangle}$$

Decompose

$$\frac{\langle E \uplus \{f(s_1,\ldots,s_n) \approx f(t_1,\ldots,t_n)\}, H\rangle}{\langle E \cup \{s_1 \approx t_1, \ldots, s_n \approx t_n\}, H\rangle} f \in \mathcal{C}$$

図 2　補題生成のための書き換え帰納法

この差分 (3) から補題を次のように生成する．差分 (3) の右辺の下線部 $@(xs, ys)$ は書き換え規則 (1) の右辺 $@(xs, ys)$ と照合する．よって，書き換え規則 (1) を右辺から左辺へ逆向きに使って，書き換え規則 (2) の右辺を書き換えると次の書き換え規則を得る．

$$\mathsf{rev}(@(\mathsf{rev}(@(xs,ys)),[x])) \to x :: \mathsf{rev}(\mathsf{rev}(@(xs,ys))) \qquad (4)$$

最後に両辺に共通する部分項 $\mathsf{rev}(@(xs, ys))$ を新しい変数 zs を用いて，(4) を一般化すると，次の補題が生成される．

$$\mathsf{rev}(@(zs,[x])) \quto\quad x :: \mathsf{rev}(zs)$$

これを等式として，等式集合 E に追加して，書き換え帰納法を適用すると証明に成功する．

以上を手続きとしてまとめる．発散系列を $s_i \to t_i (i = 0, 1, \ldots)$，文脈を G, H, C，極大差分照合を $s_i = G[u_i], s_{i+1} = G[\boxed{H[u_i]}], t_i = v_i, t_{i+1} = \boxed{C[v_i]}$，とする．このとき生成される補題は，等式 $G[\boxed{H[u_0]}] = \boxed{C\,[G[\,u_0\,]\,]}$ の共通部分項を新しい変数で置き換えることで得られる．

例 2.5 (発散鑑定法による補題生成失敗例) 以下の項書き換えシステム $\mathcal{R}$ を与え，等式集合 E を書き換え帰納法で証明する．

$$\mathcal{R} = \begin{cases} +(0,x) \to x \\ +(\mathsf{s}(x),y) \to \mathsf{s}(+(x,y)) \\ \mathsf{dbl}(0) \to 0 \\ \mathsf{dbl}(\mathsf{s}(x)) \to \mathsf{s}(\mathsf{s}(\mathsf{dbl}(x))) \end{cases}$$

$$E = \begin{cases} \mathsf{dbl}(x) \approx +(x,x) \end{cases}$$

このとき，E は $\mathcal{R}$ の帰納的定理である．しかし，書き換え帰納法は発散し，以下の発散系列が出現する．

$$\mathsf{s}(+(x,\mathsf{s}(x))) \to \mathsf{s}(\mathsf{s}(\mathsf{dbl}(x))) \qquad (5)$$

$$\mathsf{s}(\mathsf{s}(+(x_1,\mathsf{s}(\mathsf{s}(x_1))))) \to \mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{dbl}(x_1))))) \qquad (6)$$

$$\vdots$$

発散鑑定法をを試みても，(5) と (6) の差分が発見できず，補題生成に失敗する．

3　書き換え帰納法を用いた補題生成法

3.1　補題生成のための書き換え帰納法

従来提案されている発散鑑定法付き書き換え帰納法 [5] では，書き換え帰納法の証明，発散系列の観測，補題の生成と追加を逐次的に繰り返す．このため，その証明過程に特有な補題しか生成されない，

一方、補題を発見するためには多くの発散系列の観測が有用であり，より多様性のある仮定部を得るのが望ましい．そこで，この目的に合うように *Expand* 規則を変更する．

書き換え帰納法の *Expand* 規則には以下の制約がある．

- 等式部に含まれる 1 つの等式 $s \approx t$ に対して実行する．
- 簡約順序 $s > t$ が成立する等式 $s \approx t$ に対してだけ実行する．
- 一つの基本部分項 $u \in \mathcal{B}(s)$ に対して実行する．

これらの制約を緩和することによって多様な仮定集合を生成する．変更した *Expand* 規則を図 2 示す．

G-Expand 規則では簡約順序 $s > t$ が成立するしないにかかわらず，等式集合に含まれる複数の等式 $s \approx t$ に対して，複数の基本部分項 u に対して並列に $\mathrm{Expd}_u(s,t)$ による展開を行う．

さらに (図 2)，反証機能付き書き換え帰納法 [5] の推論規則の 1 つである分解則 (*Decompose*) を組み込

む．分解則は必ずしも健全であるとは限らないが，分解則を書き換え帰納法の中に組み込むことで，より単純な等式を含む仮定部を生成できる．

3.2 発散系列の抽出

以上のようにして，多様な仮定部を生成した場合，様々な複合的な発散系列を観測することがある．そのような場合，一般的な補題を見つけるためにはその発散系列から単純な系列を発見することが望ましい．そこで以下では，単純な系列を抽出する手続きを与える．まず手続きに用いる**階層項**の定義を与える．

定義 3.1 (階層項) 階層項を次のように帰納的に定義する．

1. 基本項は階層項
2. $f \in \mathcal{D}$ かつ $t_1, \ldots, t_n$ が階層項ならば，$f(t_1, \ldots, t_n)$ は階層項

階層項の集合を $\mathcal{H}(\mathcal{D}, \mathcal{C}, \mathcal{V})$ と記す．

発散系列抽出手続き

ステップ 1 仮定部から $r_0 \in \mathcal{H}(\mathcal{D}, \mathcal{C}, \mathcal{V})$ となる書き換え規則 $l_0 \to r_0$ を選ぶ．

ステップ 2 $l_0 \to r_0$ との差分が存在する書き換え規則 $l_1 \to r_1$ を仮定部から選び，その差分照合を $G[\boxed{H[\underline{l_0}]}] \to D[\boxed{C[\underline{r_0}]}]$ とする．

ステップ 3 残りの仮定部から

$$\begin{aligned} G[H[H[l_0]]] &\to D[C[C[r_0]]] \\ G[H[H[H[l_0]]]] &\to D[C[C[C[r_0]]]] \\ &\vdots \end{aligned}$$

と差分が増えていく書き換え規則を発散系列として抽出．

なお，Step1 で書き換え規則を選ぶ際は，$r_0 \in \mathcal{B}(\mathcal{D}, \mathcal{V})$ を満たすものから優先する．ステップ 1 の条件を満たす書き換え規則がなくなるまで，手続きを繰り返す．

例 3.2 (発散系列の抽出例) 以下の項書き換えシステム $\mathcal{R}$ を与え，等式集合 E を書き換え帰納法で証明する．

$$\mathcal{R} = \begin{cases} \mathsf{minus}(\mathsf{0}, y) \to \mathsf{0} \\ \mathsf{minus}(\mathsf{s}(x), \mathsf{0}) \to \mathsf{s}(x) \\ \mathsf{minus}(\mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{minus}(x, y) \end{cases}$$

$$E = \left\{ \mathsf{minus}(\mathsf{minus}(x, y), z) \approx \mathsf{minus}(\mathsf{minus}(x, z), y) \right.$$

このとき，以下のような仮定部が現れる．

$$\begin{aligned} \mathsf{minus}(\mathsf{minus}(x, y), \mathsf{0}) &\to \mathsf{minus}(x, y) \\ \mathsf{minus}(\mathsf{minus}(x, y), z) &\to \mathsf{minus}(\mathsf{minus}(x, z), y) \\ \mathsf{minus}(\mathsf{minus}(\mathsf{s}(x), y), \mathsf{0}) &\to \mathsf{minus}(\mathsf{s}(x), y) \\ \mathsf{minus}(\mathsf{minus}(x, y), \mathsf{s}(\mathsf{0})) &\to \mathsf{minus}(x, \mathsf{s}(y)) \\ \mathsf{minus}(\mathsf{minus}(\mathsf{s}(x), y), \mathsf{s}(\mathsf{0})) &\to \mathsf{minus}(x, y) \\ &\vdots \end{aligned}$$

上記の手続きに基づいて発散系列を構成する．

まず，$l_0 \to r_0$ として $\mathsf{minus}(\mathsf{minus}(x, y), \mathsf{0}) \to \mathsf{minus}(x, y)$ を選ぶ．

次に，$l_1 \to r_1$ として $\mathsf{minus}(\mathsf{minus}(x, y), \mathsf{s}(\mathsf{0})) \to \mathsf{minus}(x, \mathsf{s}(y))$ を選ぶと以下の発散系列が得られる．

$$\begin{aligned} \mathsf{minus}(\mathsf{minus}(x, y), \mathsf{0}) &\to \mathsf{minus}(x, y) \\ \mathsf{minus}(\mathsf{minus}(x, y), \mathsf{s}(\mathsf{0})) &\to \mathsf{minus}(x, \mathsf{s}(y)) \\ \mathsf{minus}(\mathsf{minus}(x, y), \mathsf{s}(\mathsf{s}(\mathsf{0}))) &\to \mathsf{minus}(x, \mathsf{s}(\mathsf{s}(y))) \\ &\vdots \end{aligned} \tag{1}$$

なお，$l_1 \to r_1$ として $\mathsf{minus}(\mathsf{minus}(\mathsf{s}(x), y), \mathsf{s}(\mathsf{0})) \to \mathsf{minus}(x, y)$ を選ぶと，発散系列は見つからない．

次に，$l_0 \to r_0$ として $\mathsf{minus}(\mathsf{minus}(\mathsf{s}(x), y), \mathsf{s}(\mathsf{0})) \to \mathsf{minus}(x, y)$ を選ぶと，発散系列

$$\begin{aligned} \mathsf{minus}(\mathsf{minus}(\mathsf{s}(x), y), \mathsf{s}(\mathsf{0})) &\to \mathsf{minus}(x, y) \\ \mathsf{minus}(\mathsf{minus}(\mathsf{s}(x), y), \mathsf{s}(\mathsf{s}(\mathsf{0}))) &\to \mathsf{minus}(x, \mathsf{s}(y)) \\ \mathsf{minus}(\mathsf{minus}(\mathsf{s}(x), y), \mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{0})))) &\to \mathsf{minus}(x, \mathsf{s}(\mathsf{s}(y))) \\ &\vdots \end{aligned} \tag{2}$$

が現れる．

次に，$l_0 \to r_0$ として $\mathsf{minus}(\mathsf{minus}(x, y), z) \to \mathsf{minus}(\mathsf{minus}(x, z), y)$ を選ぶと，発散系列

$$\begin{aligned}\mathsf{minus}(\mathsf{minus}(x,y),z) &\to \mathsf{minus}(\mathsf{minus}(x,z),y)\\ \mathsf{minus}(\mathsf{minus}(x,y),\mathsf{s}(z)) &\to \mathsf{minus}(\mathsf{minus}(x,z),\mathsf{s}(y))\\ \mathsf{minus}(\mathsf{minus}(x,y),\mathsf{s}(\mathsf{s}(z))) &\to \mathsf{minus}(\mathsf{minus}(x,z),\mathsf{s}(\mathsf{s}(y)))\\ &\vdots\end{aligned} \tag{3}$$

が現れる．

最後に，$l_0 \to r_0$ として $\mathsf{minus}(\mathsf{minus}(x,y),y) \to \mathsf{minus}(\mathsf{minus}(\mathsf{s}(x),y),\mathsf{s}(y))$ を選ぶと，発散系列

$$\begin{aligned}\mathsf{minus}(\mathsf{minus}(x,y),y) &\to \mathsf{minus}(\mathsf{minus}(\mathsf{s}(x),y),\mathsf{s}(y))\\ \mathsf{minus}(\mathsf{minus}(x,y),\mathsf{s}(y)) &\to \mathsf{minus}(\mathsf{minus}(\mathsf{s}(x),y),\mathsf{s}(\mathsf{s}(y)))\\ \mathsf{minus}(\mathsf{minus}(x,y),\mathsf{s}(\mathsf{s}(y))) &\to \mathsf{minus}(\mathsf{minus}(\mathsf{s}(x),y),\mathsf{s}(\mathsf{s}(\mathsf{s}(y))))\\ &\vdots\end{aligned} \tag{4}$$

が現れる．この発散系列を抽出すると，$l_0 \to r_0$ の候補がなくなるため手続きが終了する．

3.3 発散系列からの補題生成

得られた発散系列から補題を生成するために発散鑑定法を試みる．しかし，発散鑑定法は必ず補題を発見できるというわけではない．例えば，発散鑑定法で補題を生成する場合，差分が発見できずに補題生成に失敗するなど補題を生成出来ない場合がある．そこで，より多くの補題を構成するために新たな補題構成法を提案する．

(1) 両辺に共通する基本部分項の抽象化.

例で説明する．以下の項書き換えシステム $\mathcal{R}$ と，等式集合 E を考える．

$$\mathcal{R} = \begin{cases} +(\mathsf{0},y) \to y\\ +(\mathsf{s}(x),y) \to \mathsf{s}(+(x,y))\\ \times(\mathsf{0},y) \to \mathsf{0}\\ \times(\mathsf{s}(x),y) \to +(y,\times(x,y)) \end{cases}$$

$$E = \begin{cases} \times(\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{0})))),x)\\ \qquad\qquad \approx \times(\mathsf{s}(\mathsf{s}(\mathsf{0})),\times(\mathsf{s}(\mathsf{s}(\mathsf{0})),x)) \end{cases}$$

このとき，以下の発散系列が出現する．

$$\begin{aligned}&+(+(x,+(x,\mathsf{0})),+(+(x,+(x,\mathsf{0})),\mathsf{0}))\\ &\qquad \to +(x,+(x,+(x,+(x,\mathsf{0}))))\end{aligned} \tag{7}$$

$$\begin{aligned}&+(+(x_1,\mathsf{s}(+(x_1,\mathsf{0}))),\mathsf{s}(+(+(x_1,\mathsf{s}(+(x_1,\mathsf{0}))),\mathsf{0})))\\ &\qquad \to +(x_1,\mathsf{s}(+(x_1,\mathsf{s}(+(x_1,\mathsf{s}(+(x_1,\mathsf{0})))))))\end{aligned} \tag{8}$$

$$\vdots$$

書き換え規則 (7) において，両辺に共通する基本部分項 $+(x,\mathsf{0})$ が存在する．これを書き換え規則の左辺とする補題を提案する．

$$+(x,\mathsf{0}) \to t$$

ここで t はユーザーに入力を求める．この場合，$t = x$ とすると，以下の補題が生成される．

$$+(x,\mathsf{0}) \to x$$

(2) 変数のみを一般化した書き換え規則を補題として提案.

例で説明する．以下の項書き換えシステム $\mathcal{R}$ を与え，等式集合 E を書き換え帰納法で証明する．

$$\mathcal{R} = \begin{cases} @([],ys) \to ys\\ @(x :: xs,ys) \to x :: @(xs,ys) \end{cases}$$

$$E = \begin{cases} @(@(xs,xs),xs) \approx @(xs,@(xs,xs)) \end{cases}$$

このとき，E は $\mathcal{R}$ の帰納的定理である．しかし，書き換え帰納法は発散し，以下の発散系列が出現する．

$$\begin{aligned}&@(@(xs,xs),xs) \to @(xs,@(xs,xs))\\ &@(@(xs,x :: xs),x :: xs)\\ &\qquad \to @(xs,x :: @(xs,x :: xs))\\ &\qquad\qquad \vdots\end{aligned} \tag{9}$$

書き換え規則 (9) において，変数を一般化した書き換え規則が生成される．

$$\begin{aligned}@(@(ys,xs),xs) &\to @(ys,@(xs,xs))\\ @(@(ys,xs),xs) &\to @(xs,@(ys,xs))\\ @(@(ys,xs),xs) &\to @(xs,@(xs,ys))\\ &\vdots\end{aligned}$$

ここで，非定理な補題の生成を減らすために，一般化した変数に適当な基底項を代入した場合に両辺の正規形が一致する書き換え規則を補題として提案する．この場合，以下の書き換え規則が補題として提案される．

$$@(@(ys,xs),xs) \to @(ys,@(xs,xs))$$

4 実装と実験

本研究で提案した補題生成法を実装した．実装には関数型言語 SML をもちい，プログラムコードは約 2000 行である．簡約順序には辞書式経路順序を用いる．

本実験では，(1) 発散鑑定法付き書き換え帰納法 (文献[5] を参考) における補題生成と，(2) 提案手法付き書き換え帰納法の補題生成について，比較を行った．文献[5] の補題生成実験例 (帰納的定理である等式 30 個) に対して，補題生成を試みた．実験結果を表 1 に示す.

(2) は適用不可のものを除く 29 個の等式のうち，25 個の等式に対して補題の生成に成功している．等式 10 は等式論理ではないので (1)，(2) ともに適用出来ない．(1) で補題生成に失敗し，(2) で補題生成に成功する等式は 11 個あった．等式 1,2,5 は，(1) では発散系列に対して差分照合を適用できなかったため補題の生成に失敗している．一方，(2) では分解則を発散鑑定法に組み込むことで差分照合を適用でき，補題の生成に成功した．等式 3,27,28 は，(1) では等式に辞書式経路順序をつけることができないため失敗した．一方，(2) では，*G-Expand* 規則が順序を付けることができない等式に対しても適用できたため，補題の生成に成功した．また (1)，(2) の両方で補題生成に成功した 15 個の等式のうち，7 個の等式については，(2) で生成できた補題の数が増えた．等式 13,15,22,24,25,26 では，発散鑑定法に導入した一般化するためのヒューリスティックスが違うことが補題生成に影響している．(1)，(2) の両方で補題生成に失敗した 2 例は，等式 27, 28 である．これらは，(1) では，等式に辞書式経路順序をつけることができないため失敗した．(2) では，*G-Expand* 規則の適用により仮定部の生成には成功したが，発散系列の発見で失敗した.

5 今後の課題

HipSpec などの他の補題生成手法との比較実験をすることや，発散系列からの補題生成については，より考察を加えて強力な補題生成法を考案することが今後の課題である.

参 考 文 献

[1] Basin, T. and Walsh, T.: Difference matching, *Proc. of 11th CADE*, LNCS, Vol. 607, Springer-Verlag, 1992, pp. 295–309.

[2] Bouhoula, A., Basin, D., and Ireland, A.: *Rippling: Meta-Level Fuidance for Mathematical Reasonig*, Cambridge University Press, Cambridge, 2005.

[3] Claessen, K., Johansson, M., Rosn, D., and Smallbone, N.: Automating inductive proofs using theory exploration, *Proc. of 24th CADE*, LNCS, Vol. 7898, Springer-Verlag, 2013, pp. 392–406.

[4] Reddy, U. S.: Term rewriting induction, *Proc. of 14th RTA*, LNCS, Vol. 2706, Springer-Verlag, 2003, pp. 352–366.

[5] 嶌津聡志, 青戸等人, 外山芳人: 反証機能付き書き換え帰納法のための補題自動生成法, Vol. 26, No. 2, 2009, pp. 41–55.

[6] Walsh, T.: A divergence critic for inductive proof, *Journal of Artificial Intelligence Research*, Vol. 4, 1996, pp. 209–235.

表 1　提案手法の生成された補題 (○:補題生成成功，×:補題生成失敗，-:適用不可)

No	証明する等式	(1)	(2)	(2) で生成された補題
1	$+(\mathsf{s}(x),x) \approx \mathsf{s}(+(x,x))$	×	○	$+(x,\mathsf{s}(y)) \to \mathsf{s}(+(x,y))$
2	$\mathsf{dbl}(x) \approx +(x,x)$	×	○	$+(x,\mathsf{s}(y)) \to \mathsf{s}(+(x,y))$
3	$\mathsf{len}(@(x,y)) \approx \mathsf{len}(@(x,y))$	×	○	$\mathsf{len}(@(ys,x::xs)) \to \mathsf{s}(\mathsf{len}(@(ys,xs)))$ $\mathsf{len}(@(ys,y::(x::xs))) \to \mathsf{s}(\mathsf{len}(@(ys,y::xs)))$
4	$\mathsf{len}(@(x,y)) \approx +(\mathsf{len}(x),\mathsf{len}(y))$	○	○	$+(\mathsf{0},\mathsf{s}(x)) \to \mathsf{s}(+(\mathsf{0},x))$
5	$\mathsf{len}(@(x,x)) \approx \mathsf{dbl}(\mathsf{len}(x))$	×	○	$\mathsf{len}(@(x,(y::z))) \to \mathsf{s}(\mathsf{len}(@(x,z)))$
6	$\mathsf{even}(+(x,x)) \approx \mathsf{true}$	○	○	$\mathsf{even}(+(x,\mathsf{s}(\mathsf{s}(y)))) \to \mathsf{even}(+(x,y))$ $\mathsf{even}(\mathsf{s}(+(x,\mathsf{s}(\mathsf{s}(y))))) \to \mathsf{even}(\mathsf{s}(+(x,y)))$
7	$\mathsf{odd}(+(\mathsf{s}(x),x)) \approx \mathsf{true}$	○	○	$\mathsf{odd}(+(x,\mathsf{s}(\mathsf{s}(y)))) \to \mathsf{odd}(+(x,y))$ $\mathsf{odd}(\mathsf{s}(+(x,\mathsf{s}(\mathsf{s}(y))))) \to \mathsf{odd}(\mathsf{s}(+(x,y)))$
8	$\mathsf{even}_m(+(x,x)) \approx \mathsf{true}$	○	○	$\mathsf{even}_m(+(x,\mathsf{s}(\mathsf{s}(y)))) \to \mathsf{even}_m(+(x,y))$ $\mathsf{odd}_m(+(x,\mathsf{s}(\mathsf{s}(y)))) \to \mathsf{odd}_m(+(x,y))$
9	$\mathsf{odd}_m(+(\mathsf{s}(x),x)) \approx \mathsf{true}$	○	○	$\mathsf{odd}_m(+(x,\mathsf{s}(\mathsf{s}(A)))) \to \mathsf{odd}_m(+(x,y))$ $\mathsf{even}_m(+(x,\mathsf{s}(\mathsf{s}(y)))) \to \mathsf{even}_m(+(x,y))$
10	$\mathsf{even}_m(x) \to +(\mathsf{half}(x),\mathsf{half}(x)) \approx x$	-	-	
11	$\mathsf{half}(+(x,x)) \approx x$	○	○	$\mathsf{half}(+(x,\mathsf{s}(\mathsf{s}(y)))) \to \mathsf{s}(\mathsf{s}(\mathsf{half}(+(x,y))))$ $\mathsf{half}(\mathsf{s}(+(x,\mathsf{s}(\mathsf{s}(y))))) \to \mathsf{s}(\mathsf{s}(\mathsf{half}(\mathsf{s}(+(x,y)))))$
12	$\mathsf{half}(+(\mathsf{s}(x),x)) \approx x$	○	○	$\mathsf{half}(+(x,\mathsf{s}(\mathsf{s}(y)))) \to \mathsf{s}(\mathsf{s}(\mathsf{half}(+(x,y))))$ $\mathsf{half}(\mathsf{s}(+(x,\mathsf{s}(\mathsf{s}(y))))) \to \mathsf{s}(\mathsf{s}(\mathsf{half}(\mathsf{s}(+(x,y))))))$
13	$\mathsf{rot}(\mathsf{len}(x),x) \approx x$	○	○	$\mathsf{rot}(z,@(y,[x])) \to x :: \mathsf{rot}(z,y)$
14	$\mathsf{len}(\mathsf{rot}(\mathsf{len}(x),x)) \approx \mathsf{len}(x)$	×	○	$\mathsf{len}(\mathsf{rot}(y,@(z,[x]))) \to \mathsf{s}(\mathsf{len}(\mathsf{rot}(y,z)))$
15	$\mathsf{rot}(\mathsf{len}(x),@(x,[y])) \approx \mathsf{true}$	×	○	$\mathsf{rot}(w,@(z,[x])) \to y :: (\mathsf{rot}(w,z))$
16	$\mathsf{len}(\mathsf{rev}(x)) \approx \mathsf{len}(x)$	○	○	$\mathsf{len}(@(y,[x])) \to \mathsf{s}(\mathsf{len}(y))$
17	$\mathsf{rev}(\mathsf{rev}(x)) \approx x$	○	○	$\mathsf{rev}(@(y,[x])) \to x :: \mathsf{rev}(y)$
18	$\mathsf{rev}(@(\mathsf{rev}(x),[y])) \approx y :: x$	○	○	$\mathsf{rev}(@(x,[y])) \to y :: \mathsf{rev}(x)$
19	$\mathsf{rev}(@(\mathsf{rev}(x),[y])) \approx y :: \mathsf{rev}(\mathsf{rev}(x))$	○	○	$\mathsf{rev}(@(x,[y])) \to y :: \mathsf{rev}(x)$
20	$\mathsf{len}(\mathsf{rev}(@(x,y))) \approx +(\mathsf{len}(x),\mathsf{len}(y))$	○	○	$\mathsf{len}(@(y,[x])) \to \mathsf{s}(\mathsf{len}(y))$
21	$\mathsf{len}(\mathsf{qrev}(x,[])) \approx \mathsf{len}(x)$	○	○	$\mathsf{len}(\mathsf{qrev}(xs,x::(w::z))) \to \mathsf{s}(\mathsf{len}(\mathsf{qrev}(xs,(y::z))))$
22	$\mathsf{qrev}(x,y) \approx @(\mathsf{rev}(x),y)$	×	○	$@(\mathsf{qrev}(xs,y::(x::zs)),ys) \to$ $\mathsf{qrev}(xs,y::(x::@(\mathsf{qrev}(xs,y::zs),ys)))$
23	$\mathsf{len}(\mathsf{qrev}(x,y)) \approx +(\mathsf{len}(x),\mathsf{len}(y))$	○	○	$+(y,\mathsf{s}(x)) \to \mathsf{s}(+(y,x))$ $+(\mathsf{0},\mathsf{s}(x)) \to \mathsf{s}(+(\mathsf{0},x))$
24	$\mathsf{qrev}(\mathsf{qrev}(x,[]),[]) \approx x$	×	○	$\mathsf{qrev}(\mathsf{qrev}(xs_2,x_2::(x_1::ys)),ys) \to x_1 :: \mathsf{qrev}(\mathsf{qrev}(xs_2,x_2::ys),ys)$
25	$\mathsf{rev}(\mathsf{qrev}(x,[])) \approx x$	×	○	$\mathsf{rev}(\mathsf{qrev}(xs_2,x_2::(x_1::ys))) \to x_1 :: \mathsf{rev}(\mathsf{qrev}(xs_2,x_2::ys))$
26	$\mathsf{qrev}(\mathsf{rev}(x),[]) \approx x$	×	○	$\mathsf{qrev}(@(ys,[x])),[]) \to x :: \mathsf{qrev}(ys,[])$
27	$\mathsf{nth}(i,\mathsf{nth}(j,x)) \approx \mathsf{nth}(j,\mathsf{nth}(i,x))$	×	×	
28	$\mathsf{nth}(i,\mathsf{nth}(j,\mathsf{nth}(k,x))) \approx \mathsf{nth}(k,\mathsf{nth}(j,\mathsf{nth}(i,x)))$	×	×	
29	$\mathsf{len}(\mathsf{isort}(x)) \approx \mathsf{len}(x)$	○	×	
30	$\mathsf{sorted}(\mathsf{isort}(x)) \approx \mathsf{true}$	○	×	