反証機能付き書き換え帰納法のための補題自動生成法

嶌津 聡志　　青戸 等人　　外山 芳人
Satoshi Shimazu　　Takahito Aoto　　Yoshihito Toyama

東北大学電気通信研究所
{shimazu, aoto, toyama}@nue.riec.tohoku.ac.jp

概要

書き換え帰納法 (Reddy, 1989) は，項書き換えシステムにもとづく帰納的定理の自動証明法である．書き換え帰納法による帰納的定理の自動証明が成功するためには，適切な補題が一般には必要である．このため，発散鑑定法 (Walsh, 1996) や健全一般化法 (Urso and Kounalis, 2004) などの補題自動生成法が提案されている．本論文では，健全な補題のみを生成する健全発散鑑定法を提案し，健全一般化法と組み合わせることで，反証機能付き書き換え帰納法に適した補題自動生成法が実現できることを実験をとおして明らかにする．

1　はじめに

関数型言語や代数的仕様記述法などの等式論理を基礎とする系では，さまざまな性質を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，等式論理における帰納的定理の自動証明手法 [1, 2, 5, 6, 7, 11, 12, 13, 14] 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重要である．書き換え帰納法 (rewriting induction) は，Reddy[12] により提案された帰納的定理の自動証明法であり，書き換え帰納法にもとづく定理自動証明システムとしては SPIKE[5, 6] が知られている．

書き換え帰納法では，証明すべき等式の集合と仮定の集合の対に対して推論規則を繰り返し適用する．このとき，推論規則の適用に失敗する場合や，推論規則の導出が無限に繰り返される場合には，証明すべき等式が帰納的定理かどうかを判定できない．後者の場合には，適切な補題の追加が証明の成功に有効である．

このような補題の自動生成法として，Walsh の提案した発散鑑定法 (divergence critic)[16] が知られている．発散鑑定法は明示的帰納法のヒューリスティクスであるリップリング法 [7] を書き換え帰納法の補題生成に応用したものである．書き換え帰納法の証明過程から得られる等式系列を差分照合 [4] によって特定し，差分照合の差分情報から適切な補題を生成する方法である．

一般には, 補題の自動生成機能を組み込むことにより, 帰納的定理自動証明システムの証明能力の向上を図ることができる．しかし，反証機能付き書き換え帰納法 (rewriting induction with disproof)[5, 6] では，与えられた等式集合に対して証明と反証を並行して実行するため，誤った補題を追加すると誤った反証が導かれる．この場合には，補題の追加を注意深く行う必要がある．実際，反証機能付き書き換え帰納法を実装している SPIKE では，発散鑑定法による補題生成機能を利用者に提供している．しかし，補題を自動的に生成・追加するわけではない．

従来の補題自動生成法では，生成される補題の健全性 (帰納的定理から生成された補題も帰納的定理となること) は必ずしも保証されていなかったが，Urso と Kounalis は健全な補題を自動生成する健全一般化法 (sound generalization) を提案した [15]．健全一般化法では，項書き換えシステムの書き換え規則の構造を注意深く解析することにより，帰納的定理である等式の両辺に出現する共通部分項のうち，新しい変数に置き換えて得られた等式も帰納的定理となる共通部分項を特定する．Urso と Kounalis は書き換え帰納法にもとづく定理自動証明システム NICE [14, 15] 上に健全一般化法を実装し，その有効性を示している．

一方，発散鑑定法は健全性をもたないため，誤まった補題を生成してしまうことがある．誤まった補題の生成を困難にするヒューリスティクスも文献 [16] で提示されてはいるが，そのヒューリスティクスを用いたとしても (自然な例に対して) 誤まった補題生成が起こる場合がある．し

かし，発散鑑定法は書き換え帰納法に対して極めて有効であり，健全一般化法では対応できない補題生成に成功する場合も多い．このため，両者の特徴を併せもつ健全な補題自動生成法の実現が望まれる．しかしながら，著者らの知る限り，このような補題自動生成法の提案はこれまでなされていない．

本論文では，発散鑑定法とほぼ同等な補題生成能力をもち，健全な補題生成法である健全発散鑑定法を提案する．これは，発散鑑定法の基本部分が，仮定集合に含まれる書き換え規則の変形と健全一般化法の組み合わせで実現できるという我々の観察にもとづいて構成したものである．さらに，我々の提案した健全発散鑑定法を従来の健全一般化法と組み合わせることで，両者の特徴を併せもつ強力な補題生成が実現できることを実験によって明らかにする．

本論文の構成は次のとおりである．2 節では必要となる定義を与え，書き換え帰納法について説明する．3 節では発散鑑定法と健全一般化法による補題生成を説明する．4 節では，健全な新しい補題生成法として健全発散鑑定法を提案し，反証機能付き書き換え帰納法上での証明実験をとおして補題生成能力を評価する．

2 準備

2.1 項書き換えシステム

本論文でもちいる定義および記法を紹介する．詳細は文献 [3, 9] 等を参照されたい．

ソート付き関数記号集合およびソート付き変数集合を $\mathcal{F}, V$ で表す．$\mathcal{F}, V$ 上のソート付き項の集合を $T(\mathcal{F}, V)$ で表す．項 t の**根記号**とは $t = x \in V$ のときは x，$t = f(t_1, \ldots, t_n)$ $(f \in \mathcal{F})$ のときは f を指す．$C[t]$ は文脈 C のホールを項 t で置き換えて得られる項を表わす．項 u が t の**部分項**である $(u \trianglelefteq t)$ とは，$t = C[u]$ なる文脈 C があるときをいう．

項 t に現れる変数がすべて異なるとき，t は**線形**であるという．t に現れる変数の集合を $V(t)$ と記す．$V(t) = \emptyset$ となる t を**基底項**とよぶ．定義域上の任意の変数 x について $\sigma_g(x)$ が基底項となる代入 σ_g を**基底代入**とよぶ．本論文では，項 t に対する基底代入 σ_g を考えるときには，$V(t\sigma_g) = \emptyset$ をみたすものと仮定する．

同一ソートの項 s, t の対を**等式**とよび，$s \doteq t$ と記す．ここで，等式 $s \doteq t$ と $t \doteq s$ は同一視する．$l \notin V$ かつ $V(r) \subseteq V(l)$ なる同一ソートの項 l, r の対を**書き換え規則**とよび，$l \to r$ と記す．書き換え規則の集合を**項書き換えシステム**という．以下では，項書き換えシステムからソートが自明な場合には，ソートを省略する．

項書き換えシステムが**左線形**であるとは，すべての書き換え規則の左辺が線形であるときをいう．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**とよぶ．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．項 $t \in T(\mathcal{C}, V)$ を**構成子項**とよぶ．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．$\mathcal{B}(t) = \{u \trianglelefteq t \mid u \text{ は基本項}\}$ は項 t の**基本部分項**集合を表す．項書き換えシステムが**構成子システム**であるとは，すべての書き換え規則の左辺が基本項であるときをいう．

$\mathcal{R}$ を項書き換えシステムとする．文脈 C，代入 σ，書き換え規則 $l \to r \in \mathcal{R}$ 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．$\to_{\mathcal{R}}$ の反射推移閉包を $\xrightarrow{*}_{\mathcal{R}}$，反射対称推移閉包を $\xleftrightarrow{*}_{\mathcal{R}}$ と記す．$t \to_{\mathcal{R}} u$ となる項 u が存在しないとき，項 t を**正規形**とよび，$s \xrightarrow{*}_{\mathcal{R}} t$ なる正規形 t を s の正規形とよぶ．

任意の (基底) 項 t, t_1, t_2 について $t_1 \xleftarrow{*}_{\mathcal{R}} t \xrightarrow{*}_{\mathcal{R}} t_2$ ならば，ある (基底) 項 s が存在して $t_1 \xrightarrow{*}_{\mathcal{R}} s \xleftarrow{*}_{\mathcal{R}} t_2$ となるとき，$\mathcal{R}$ は (**基底**) **合流性**をもつという．$\to_{\mathcal{R}}$ が整礎であるとき，$\mathcal{R}$ は**停止性**をもつという．代入および文脈に閉じている整礎な半順序を**簡約順序**とよぶ．任意の基底項 t に対してある基底構成子項 s が存在して $t \xrightarrow{*}_{\mathcal{R}} s$ となるとき，$\mathcal{R}$ は**十分完全性** [10] をもつという．本論文では，項書き換えシステム $\mathcal{R}$ は停止性と十分完全をもつ構成子システムを仮定する．

2.2 帰納的定理と書き換え帰納法

等式 $s \doteq t$ が項書き換えシステム $\mathcal{R}$ の**帰納的定理**であるとは，任意の基底代入 σ_g に対して $s\sigma_g \xleftrightarrow{*}_{\mathcal{R}} t\sigma_g$ が成り立つことをいい，$\mathcal{R} \models_{\mathrm{ind}} s \doteq t$ と記す．等式集合 E に含まれるすべての等式が $\mathcal{R}$ の帰納的定理であるとき，E を $\mathcal{R}$ の帰納的定理とよび，$\mathcal{R} \models_{\mathrm{ind}} E$ と記す．

例 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．

$$\mathcal{R} = \left\{ \begin{array}{lcl} 0 + y & \to & y \\ \mathrm{s}(x) + y & \to & \mathrm{s}(x + y) \end{array} \right.$$

このとき，等式 $x+0 \doteq x$ を考えると，$x+0 \xleftrightarrow{*}_{\mathcal{R}} x$ は成立しない．しかし，任意の基底代入 σ_g に対して $(x+0)\sigma_g \xleftrightarrow{*}_{\mathcal{R}} x\sigma_g$ は成立するので，$\mathcal{R} \models_{\mathrm{ind}} x + 0 \doteq x$. □

Simplify

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

Delete

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

Postulate

$$\frac{\langle E,\ H\rangle}{\langle E \cup L,\ H\rangle}\ L : \text{補題集合}$$

Expand

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

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

図 1　書き換え帰納法の推論規則

Sound Postulate

$$\frac{\langle E,\ H\rangle}{\langle E \cup L,\ H\rangle}\ \mathcal{R} \models_{\mathrm{ind}} E \Rightarrow \mathcal{R} \models_{\mathrm{ind}} L$$

Decompose

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

Disproof

$$\frac{\langle E \uplus \{s \doteq x\}, H\rangle}{\text{Disproof}}\ x \in V \setminus V(s)$$

$$\frac{\langle E \uplus \{f(s_1,\ldots,s_n) \doteq x\}, H\rangle}{\text{Disproof}}\ f \in \mathcal{C}, x \in V$$

$$\frac{\langle E \uplus \{f(s_1,\ldots,s_m) \doteq g(t_1,\ldots,t_n)\}, H\rangle}{\text{Disproof}}\ f \neq g,\ f,g \in \mathcal{C}$$

図 2　反証の推論規則

帰納的定理の自動証明法として**書き換え帰納法** (rewriting induction) が知られている [5, 6, 12]．書き換え帰納法は等式集合 E と書き換え規則集合 H (仮定集合とよぶ) の対 $\langle E, H\rangle$ に対する図 1 の推論規則として与えられる．$\uplus$ は直和，$>$ は $\mathcal{R} \subseteq >$ なる簡約順序，$\mathrm{mgu}(s,t)$ は s と t の最汎単一化子を表す．$\langle E, H\rangle$ に推論規則を適用して $\langle E', H'\rangle$ が得られたとき $\langle E, H\rangle \vdash \langle E', H'\rangle$ と記す．$\vdash$ の反射推移閉包を $\vdash^*$ と記す．このとき以下の命題が成り立つ．

命題 1 (書き換え帰納法の正当性 [5, 6, 12])． $\mathcal{R}$ は停止性と十分完全性をもつ構成子システムとする．このとき，$\langle E, \emptyset\rangle \vdash^* \langle \emptyset, H\rangle$ ならば $\mathcal{R} \models_{\mathrm{ind}} E$.

反証 (disproof) とは，与えられた等式が帰納的定理でないこと (非定理) を示すことである．反証機能付き書き換え帰納法 (rewriting induction with disproof) の推論規則は図 1 の *Simplify* 規則，*Delete* 規則，*Expand* 規則に図 2 の推論規則を付け加えたものである [5, 6]．ただし，反証付き書き換え帰納法では *Postulate* 規則の代わりに *Sound Postulate* 規則をもちいる．これは，任意の非定理を補題として追加すると正しい反証が行われないからである．非定理を示す Disproof が導出された場合には，反証機能付き書き換え帰納法は終了する．

本論文の反証機能付き書き換え帰納法の枠組みは，Bouhoula ら [5, 6] の部分システムとなっており，以下の命題が同様に成立する．

命題 2 (反証機能付き書き換え帰納法の正当性 [5, 6])． $\mathcal{R}$ は基底合流性，停止性と十分完全性をもつ構成子システムとする．このとき，$\langle E, \emptyset\rangle \vdash^* \langle \emptyset, H\rangle$ ならば $\mathcal{R} \models_{\mathrm{ind}} E$. さらに，$\langle E, \emptyset\rangle \vdash^*$ Disproof ならば $\mathcal{R} \not\models_{\mathrm{ind}} E$.

本論文では，(反証機能付き) 書き換え帰納法の手続きを以下の戦略にもとづいて実現する．まず，与えられた $\langle E, H\rangle$ に対して，*Simplify* 規則を繰り返し適用し，等式集合 E の各等式の両辺の正規形を求める．ついで，*Delete* 規則を繰り返し適用し，等式集合 E から両辺が同じ等式を取り除いて $\langle E_0, H_0\rangle$ を得る．$\langle E_n, H_n\rangle$ から $\langle E_{n+1}, H_{n+1}\rangle$ $(n \geq 0)$ を得るためには，$\langle E_n, H_n\rangle$ に対して *Expand* 規則の適用を試み，適用が不可能である場合には証明は失敗して終了する．成功した場合には，*Simplify* 規則を繰り返し適用し，等式集合 E の各等式の両辺の正規形を求める．ついで，*Delete* 規則を繰り返し適用し，等式集合 E から両辺が同じ等式を取り除く．次に，反証機能の有無で以下のどちらかを実行する．[反証機能無しの場合] *Postulate* 規則を適用し，等式集合 E に適当な等式 (補題) 集合 L を追加して $\langle E_{n+1}, H_{n+1}\rangle$ を求める．[反証機能付きの場合] 等式集合 E の各等式に *Decompose* 規則を可能な限り適用する．*Decompose* 規則の適用中に *Disproof* 規則が適用できたら，非定理が存在するので終了する．次に，*Sound Postulate* 規則に従い等式集合 E に健全な補題集合 L を追加して，$\langle E_{n+1}, H_{n+1}\rangle$ を得る．

なお，上記の手続きで補題集合 L は空集合でもよい．手

続きで得られた系列を

$$\begin{aligned}\langle E_0, H_0\rangle \rightsquigarrow \langle E_1, H_1\rangle \rightsquigarrow \langle E_2, H_2\rangle \rightsquigarrow \cdots \\ \rightsquigarrow \langle E_n, H_n\rangle \rightsquigarrow \langle E_{n+1}, H_{n+1}\rangle \rightsquigarrow \cdots\end{aligned}$$

と記す．ここで，仮定集合 H_n $(n \geq 0)$ は単調増加することに注意する．この系列が無限となる場合には，(反証機能付き) 書き換え帰納法は**発散する**という．

以下では簡単のために，とくに断らない限り書き換え帰納法の手続きは，補題生成法が健全な場合には反証機能付き書き換え帰納法の手続き，健全でない場合には反証機能無し書き換え帰納法の手続きを意味することとする．さらに，常に $L = \emptyset$ として補題の追加を行わない書き換え帰納法の手続きを RI と表す．

3 補題の自動生成法

書き換え帰納法が発散するときには，適当な補題集合 L を等式集合 E に追加すると，証明に成功する場合が多い．証明過程から補題集合 L を自動生成する手法として，Walsh の提案した発散鑑定法 (divergence critic)[16]，Urso と Kounalis が提案した健全一般化法 (sound generalization)[15] などが知られている．本節ではこれらの補題生成法について簡単に説明し，書き換え帰納法の手続きに組み込む方法について述べる．

3.1 発散鑑定法

発散鑑定法は無限系列 $\langle E_i, H_i\rangle$ $(i \geq 0)$ を観測し，H_i $(i = 0, 1, 2, \ldots)$ に含まれる書き換え規則の差分情報から新しい書き換え規則を構成する．さらに差分情報をもちいて，その書き換え規則の共通部分項を新しい変数に置き換えることで補題を生成する．以下では，発散鑑定法による補題生成を簡単な例で説明する．

例 2 (発散鑑定法による補題生成例)． リスト結合と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)．等式集合 E に書き換え帰納法を適用する．

$$\mathcal{R} = \left\{ \begin{array}{lcl} [\,] \mathbin{@} ys & \to & ys \\ (x :: xs) \mathbin{@} ys & \to & x :: (xs \mathbin{@} ys) \\ \mathrm{rev}([\,]) & \to & [\,] \\ \mathrm{rev}(x :: xs) & \to & \mathrm{rev}(xs) \mathbin{@} [x] \end{array} \right.$$

$$E = \left\{ \begin{array}{lcl} \mathrm{rev}(\mathrm{rev}(xs \mathbin{@} ys)) & \doteq & xs \mathbin{@} ys \end{array} \right.$$

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，H_i $(i = 0, 1, 2, \ldots)$ において，以下の発散系列を観測する．

$$\begin{aligned} \mathrm{rev}(\mathrm{rev}(xs \mathbin{@} ys)) &\to xs \mathbin{@} ys && (1) \\ \mathrm{rev}(\mathrm{rev}(zs \mathbin{@} ys) \mathbin{@} [z]) &\to z :: (zs \mathbin{@} ys) && (2) \\ \mathrm{rev}((\mathrm{rev}(vs \mathbin{@} ys) \mathbin{@} [v]) \mathbin{@} [z]) &\to z :: (v :: (zs \mathbin{@} ys)) \\ &\vdots \end{aligned}$$

発散鑑定法は発散系列からある特定のパターンを見つけ，そのパターンから補題を得る．発散系列のパターンを捉えるために 2 つの書き換え規則の差分照合 (difference matching) を考える．たとえば，書き換え規則 (1) と (2) の差分照合は，書き換え規則 (2) に以下のような枠と下線を付け加えることによって表現される．

$$\mathrm{rev}(\boxed{\underline{\mathrm{rev}(zs \mathbin{@} ys)} \mathbin{@} [z]}) \to \boxed{z :: (zs \mathbin{@} ys)} \qquad (3)$$

ここで，差分照合 (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．(枠 (wave-front) と下線 (wave-hole) の正確な定義については文献 [4, 7] を参照のこと．)

差分照合 (3) から補題を次のようにして生成する．差分照合 (3) の右辺の下線部 $zs \mathbin{@} ys$ は書き換え規則 (1) の右辺 $xs \mathbin{@} ys$ と照合するので，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 (4) を得る．

$$\mathrm{rev}(\mathrm{rev}(zs \mathbin{@} ys) \mathbin{@} [z]) \to z :: \mathrm{rev}(\mathrm{rev}(zs \mathbin{@} ys)) \quad (4)$$

このように書き換え規則を逆向きに適用することによってもうひとつの書き換え規則の右辺を変形することを**書き換え規則変形操作** M (modification) とよぶ．ここで，差分照合 (3) の左辺の下線部 $\mathrm{rev}(zs \mathbin{@} ys)$ は，差分情報から書き換え規則 (4) の右辺にも現れることがわかる．そこで，書き換え規則 (4) の両辺に現れる共通部分項 $\mathrm{rev}(zs \mathbin{@} ys)$ を新しい変数 ws で置き換えると以下の等式 (5) が補題として生成される．

$$\mathrm{rev}(ws \mathbin{@} [z]) \to z :: \mathrm{rev}(ws) \qquad (5)$$

このように差分照合から両辺の共通部分項を変数に置き換えることを**部分項消去操作** C (cancelation) とよぶ．補題 (5) を $Postulate$ 規則によって追加すると書き換え帰納法の証明は成功する． □

発散鑑定法付き書き換え帰納法 MC の手続きの概要を図 3 に示す．発散鑑定法では，書き換え帰納法の手続き

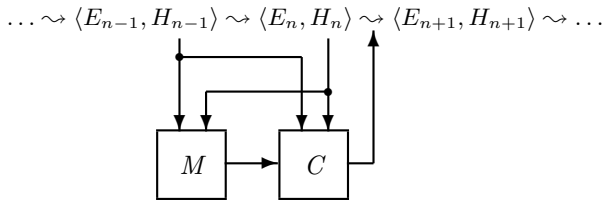

図 3　発散鑑定法付き書き換え帰納法 MC の補題生成

入力：仮定集合 H_{n-1}, H_n
出力：補題集合 L

$$
\begin{array}{rcll}
C_1[s] & \to & t & \in H_{n-1} \\
C_1\sigma[\boxed{C_2[s\sigma]}] & \to & \boxed{C_3[t\sigma]} & \in H_n \setminus H_{n-1} \quad (\sigma : \text{変数改名}) \\
\stackrel{M}{\Longrightarrow} \quad C_1\sigma[C_2[s\sigma]] & \to & C_3[C_1\sigma[s\sigma]] & \\
\stackrel{C}{\Longrightarrow} \quad C_1\sigma[C_2[x]] & \doteq & C_3[C_1\sigma[x]] & (x : \text{新しい変数})
\end{array}
$$

図 4　発散鑑定法の手続き

で得られた系列 $\langle E_{n-1}, H_{n-1}\rangle$，$\langle E_n, H_n\rangle$ $(n \geq 1)$ を観測し，H_{n-1} と H_n の間に特定の差分を発見すると，書き換え規則変形操作 M と部分項消去操作 C をもちいて補題生成を行う．このとき，各ステップで生成される補題は高々 1 個である．また，図 4 に示す発散鑑定法の手続きからわかるように，操作 M と C には差分情報が利用される．

例 3 (発散鑑定法の非健全な補題生成例). 反復的なリスト反転を行う項書き換えシステム $\mathcal{R}$ で与える．等式集合 E に書き換え帰納法を適用する．

$$
\mathcal{R} = \left\{ \begin{array}{lcl} \mathrm{qrev}([\,], ys) & \to & ys \\ \mathrm{qrev}(x :: xs, ys) & \to & \mathrm{qrev}(xs, x :: ys) \end{array} \right.
$$

$$
E = \left\{ \begin{array}{lcl} \mathrm{qrev}(\mathrm{qrev}(xs, [\,]), [\,]) & \doteq & xs \end{array} \right.
$$

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，H_i $(i = 0, 1, 2, \ldots)$ において，以下の発散系列を観測する．

$$
\begin{array}{rr}
\mathrm{qrev}(\mathrm{qrev}(xs, [\,]), [\,]) \to xs & (6) \\
\mathrm{qrev}(\mathrm{qrev}(ys, y :: [\,]), [\,]) \to y :: ys & (7) \\
\mathrm{qrev}(\mathrm{qrev}(zs, z :: (y :: [\,])), [\,]) \to y :: (z :: zs) & (8) \\
\vdots &
\end{array}
$$

書き換え規則 (6) と (7) の差分照合は

$$
\mathrm{qrev}(\mathrm{qrev}(ys, \boxed{\underline{y :: [\,]}}), [\,]) \to \boxed{\underline{y :: ys}} \qquad (9)
$$

となる．差分照合 (9) の差分情報から操作 M をもちいて書き換え規則 (7) を書き換え規則

$$
\mathrm{qrev}(\mathrm{qrev}(ys, y :: [\,]), [\,]) \to y :: (\mathrm{qrev}(\mathrm{qrev}(ys, [\,]), [\,]))
$$

に変形する．この書き換え規則に操作 C をもちいて共通部分項 $[\,]$ を新しい変数 ws に置き換えると補題

$$
\mathrm{qrev}(\mathrm{qrev}(ys, y :: ws), [\,]) \doteq y :: (\mathrm{qrev}(\mathrm{qrev}(ys, ws), [\,]))
$$

を得る．しかし，この補題は非定理である．　□

発散鑑定法はこのような非定理の生成を防ぐために，発散系列から得られる複数の差分照合を観察する．たとえば，書き換え規則 (7) と書き換え規則 (8) の差分照合

$$
\mathrm{qrev}(\mathrm{qrev}(zs, \boxed{z :: (\underline{y :: [\,]})}), [\,]) \to y :: \boxed{(z :: \underline{zs})}
$$

の右辺は，発散鑑定法の手続きに記述されている差分照合の右辺の形と異なるために補題を生成できない．複数の差分照合を観察することは，補題生成のに必要な計算時間が増加する．しかし，非定理を生成する可能性が低くなると考えられる．文献 [16] では，経験的に 3 つの書き換え規則から 2 つの差分照合を観察する方法が有効であると述べている．本論文の発散鑑定法もこの方法を採用する．

3.2　健全一般化法

等式 $s' \doteq t'$ が等式 $s \doteq t$ の**一般化**であるとは，ある代入 σ が存在して $s'\sigma = s$ かつ $t'\sigma = t$ となることをいう．健全一般化法は，項書き換えシステム $\mathcal{R}$ の構造にもとづき，等式に出現する共通部分項を適当な変数に置き換えて等式の一般化を行う．つまり，健全一般化法では等式の一般化のみによって補題生成を行う．このとき，$\mathcal{R}$ を**単相項書き換えシステム**に制限することで，等式の一般化の健全性が保証される [15]．以下では，単相項書き換えシステムについて簡単に紹介する．

ソート集合 $\mathcal{S}$ とソート付き関数記号集合 $\mathcal{F}$ の対 $\langle \mathcal{S}, \mathcal{F}\rangle$ をシグネチャとよぶ．関数記号 f は，ソート $\alpha_i, \beta \in \mathcal{S}$ に対して，$f : \alpha_1 \times \cdots \times \alpha_n \to \beta$ と記す．特に $n = 0$ のとき，$f : \to \beta$ は単に $f : \beta$ と書く．異なるソート α と β に対して，ある構成子記号 $f : \alpha_1 \times \cdots \times \beta \times \cdots \times \alpha_n \to \alpha \in \mathcal{C}$ が存在するときに $\alpha \succ_{\mathcal{S}} \beta$ と記す．**単相シグネチャ** (monomorphic signature) とは次の条件をみたすシグネチャをいう：(1) $\mathcal{S}$ 上の関係 $\succ_{\mathcal{S}}$ は狭義の半順序，(2) 任意のソート $\alpha \in \mathcal{S}$ に対して，(2-a) 構成子記号 $f : \alpha \in \mathcal{C}$ がただ一つ存在し，(2-b) 構成子記号 $f : \alpha_1 \times \cdots \times \alpha_n \to \alpha \in \mathcal{C}$ $(n \geq 1)$ に対して，$\alpha = \alpha_i$ となる i がただ一つ存在する．単相シグネチャの直感的な意

味は，単相シグネチャ上の構成子項がリスト構造になることである．

例 4 (単相シグネチャ).

$$\mathcal{S} = \{\mathrm{Nat},\ \mathrm{List}\},$$
$$\mathcal{C} = \left\{ \begin{array}{l} 0 : \mathrm{Nat},\ \mathrm{s} : \mathrm{Nat} \to \mathrm{Nat}, \\ [\,] : \mathrm{List},\ :: : \mathrm{Nat} \times \mathrm{List} \to \mathrm{List} \end{array} \right\},$$
$$\mathcal{D} = \{+ : \mathrm{Nat} \times \mathrm{Nat} \to \mathrm{Nat},\ \ @\ : \mathrm{List} \times \mathrm{List} \to \mathrm{List}\}.$$

このとき，シグネチャ $\langle \mathcal{S}, \mathcal{C} \cup \mathcal{D} \rangle$ は単相である． □

単相項書き換えシステムとは，単相シグネチャ上の基底合流性，停止性と十分完全性をもつ左線形項書き換えシステムをいう [15]．健全一般化法による補題生成を簡単な例で説明する．

例 5 (健全一般化法による補題生成例). 例 1 の単相項書き換えシステム $\mathcal{R}$ をもちいて，以下の等式集合 E に対して書き換え帰納法を適用する．

$$E = \{\ \ (x + x) + x\ \doteq\ x + (x + x) \qquad (10)$$

Expand 規則および *Simplify* 規則を E に適用すると，等式 $\mathrm{s}((x{+}\mathrm{s}(x)){+}\mathrm{s}(x)) \doteq \mathrm{s}(x{+}\mathrm{s}(x{+}\mathrm{s}(x)))$ が得られる．しかし，*Expand* 規則で得られた書き換え規則 $(x+x)+x \to x+(x+x) \in H$ はこの等式に適用できない．このため，等式は再び *Expand* 規則により展開される．この過程を繰り返すことにより書き換え帰納法は発散する．

一方，等式 (10) の共通部分項 x の一部を変数 y, z に置き換えて一般化して得られる等式 (11) を補題として E に追加した場合を考える．

$$E' = \left\{ \begin{array}{l} (x + x) + x \doteq x + (x + x) \\ (x + y) + z \doteq x + (y + z) \end{array} \right. \qquad (11)$$

Expand 規則を等式 (11) に適用すると，等式 $\mathrm{s}((x + y) + z) \doteq \mathrm{s}(x{+}(y{+}z))$ が得られる．今度は，*Expand* 規則で得られた書き換え規則 $(x+y)+z \to x+(y+z) \in H$ は，この等式の左辺に適用できるだけでなく，等式 $(x+x)+x \doteq x+(x+x)$ にも適用できるので，等式 (10) だけの場合に生じていた発散は生じず，証明に成功する． □

この例のように，一般化した等式を補題として追加することで証明に成功する場合がある．しかし，等式の共通部分項を適当に変数で置き換えるような一般化法では非定理となる補題を生成することがある．これに対し，健全一般化法ではそのようなことはない．以下の命題により生成される補題の健全性が保証されているからである．

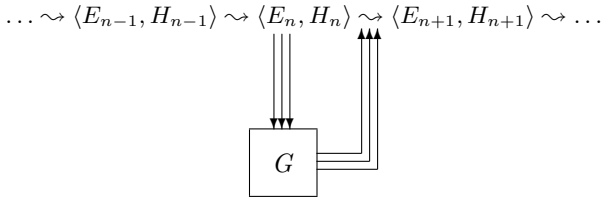

図 5　健全一般化法付き書き換え帰納法 G の補題生成

命題 3 (健全一般化法の健全性 [15]). $\mathcal{R}$ は単相項書き換えシステム，等式 $s' \doteq t'$ は等式 $s \doteq t$ から健全一般化法により生成された補題とする．このとき，$\mathcal{R} \models_{\mathrm{ind}} s \doteq t$ ならば $\mathcal{R} \models_{\mathrm{ind}} s' \doteq t'$.

健全一般化法付き書き換え帰納法 G の手続きの概要を図 5 に示す．健全一般化法では，書き換え帰納法で得られた系列 $\langle E_n, H_n \rangle$ $(n \geq 0)$ の等式集合 E_n に含まれる等式に対して，構成子記号の反射的引数位置 (reflective argument position) や定義記号の下降位置 (downward position) および上昇位置 (upward position) などを解析して，一般化すべき共通部分項の位置を特定する [15]．このとき，発散鑑定法とは異なり，発散系列の差分情報は利用されず，書き換え規則変形操作も行われない．また，E_n のすべての等式に対して一般化を行うため，系列の各ステップで生成される補題の最大個数は E_n の要素数となることに注意する．

4　健全な発散鑑定法

後述する 4.2 節の表 1 で明らかになるが，発散鑑定法と健全一般化法の有効範囲が互いに補完する関係になる例題が存在する．そこで，書き換え帰納法の手続きの系列の各ステップにおいて，発散鑑定法で生成される補題と健全一般化法で生成される補題を同時に追加することで，両者の有効範囲を併せもつ強力な補題生成を実現する方法が考えられる．発散鑑定・健全一般化法付き書き換え帰納法の手続き $MC{+}G$ の概要は図 6 となる．ここでは，発散鑑定法で生成された補題に対して健全一般化法を適用するので，MC よりも一般的な補題生成が可能となる．したがって，MC と G のどちらも発散する場合にも $MC{+}G$ は証明に成功することがある．しかし，このような単純な組み合わせでは，発散鑑定法が健全でないため，反証機能が正しく動作しない．

そこで，この問題点を解決するひとつの方法として，本

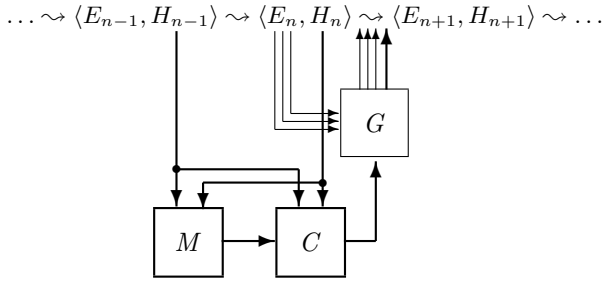

図 6　発散鑑定・健全一般化法付き書き換え帰納法 $MC{+}G$ の補題生成

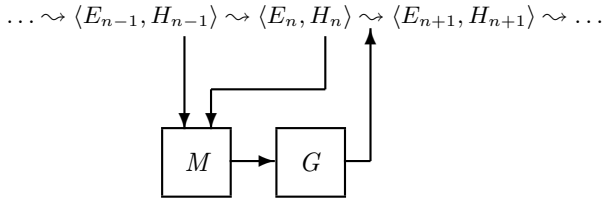

図 7　健全発散鑑定法付き書き換え帰納法 MG の補題生成

節では発散鑑定法と同程度の有効範囲をもつ健全な新しい補題生成法，健全発散鑑定法を提案する．さらに，健全発散鑑定法を従来の健全一般化法と組み合わせることで，反証機能に対しても正しく動作する強力な補題生成法が実現できることを明らかにする．

4.1　健全発散鑑定法

発散鑑定法による補題生成の本質は，発散系列の差分情報にもとづく書き換え規則変形操作 M 部分項消去操作 C にあり，操作 M は健全一般化法のような静的な一般化のみでは困難な補題生成にきわめて有効に働く．一方，差分情報にもとづく操作 C は，発散鑑定法の健全性が保証されない原因となっている．そこで，発散鑑定法の部分項消去操作 C を，健全一般化法 G による部分項消去に置き換えることが考えられる．この置き換えによって，従来の発散鑑定法の有効範囲を保ったままで健全性が保証される可能性があるからである．このような健全発散鑑定法付き書き換え帰納法 MG の手続きの概要を図 7 で示す．以下では，簡単な例で健全発散鑑定法の手続きを説明する．

例 6 (健全発散鑑定法による補題生成例). 例 2 の単相項書き換えシステム $\mathcal{R}$ をもちいて，以下の等式集合 E に対して書き換え帰納法を適用する．

$$E = \left\{ \begin{array}{rcl} \mathrm{rev}(\mathrm{rev}(xs \mathbin{@} ys)) & \doteq & xs \mathbin{@} ys \end{array} \right.$$

このとき，例 2 ですでに述べたように，書き換え帰納法は発散し，H_i $(i = 0, 1, 2, \ldots)$ において，以下の発散系列を観測する．

$$\mathrm{rev}(\mathrm{rev}(xs \mathbin{@} ys)) \to xs \mathbin{@} ys \tag{12}$$
$$\mathrm{rev}(\mathrm{rev}(zs \mathbin{@} ys) \mathbin{@} [z]) \to z :: (zs \mathbin{@} ys) \tag{13}$$
$$\mathrm{rev}((\mathrm{rev}(vs \mathbin{@} ys) \mathbin{@} [v]) \mathbin{@} [z]) \to z :: (v :: (zs \mathbin{@} ys))$$
$$\vdots$$

ここで, 書き換え規則 (12) を右辺から左辺へ逆向きに使って，書き換え規則 (13) の右辺を書き換えると次の書き換え規則 (14) を得る．

$$\mathrm{rev}(\mathrm{rev}(zs \mathbin{@} ys) \mathbin{@} [z]) \to z :: \mathrm{rev}(\mathrm{rev}(zs \mathbin{@} ys)) \tag{14}$$

このとき，書き換え規則 (12), (13) が帰納的定理ならば書き換え規則 (14) も帰納的定理である．次に，書き換え規則 (14) に対して健全一般化法を適用すると，両辺の共通部分項 $\mathrm{rev}(zs \mathbin{@} ys)$ を新しい変数 ws に置き換えた等式 (15) が補題として生成される．

$$\mathrm{rev}(ws \mathbin{@} [z]) \to z :: \mathrm{rev}(ws) \tag{15}$$

ここで，$\mathcal{R}$ は単相項書き換えシステムであるから，命題 3 より等式 (15) は健全な補題であることが保証される．さらに，例 2 で発散鑑定法によって生成された補題 (5) と補題 (15) は一致していることに注意する．つまり，発散鑑定法で生成された補題が，発散系列に含まれる書き換え規則の変形と健全一般化の組み合わせで健全に生成されたことになる．　□

健全発散鑑定法の手続きを図 8 に示す．ここでは，発散系列に含まれる 2 つの書き換え規則をもちいて変形を行い，その結果に健全一般化法を適用している．健全発散鑑定法では，補題生成に差分情報が必要ないので，発散鑑定法の書き換え規則変形操作 M よりも一般的な条件で書き換え規則の変形を行っている．単相項書き換えシステムに対して，この手続きが健全な補題を生成することは以下の定理により保証される．

定理 1 (健全発散鑑定法の健全性). $\mathcal{R}$ は単相項書き換えシステム，等式 $u \doteq v$ を発散系列 $s \to t, s' \to t', \ldots$ から健全発散鑑定法により生成された補題とする．このとき，$\mathcal{R} \models_{\mathrm{ind}} s \doteq t$, $\mathcal{R} \models_{\mathrm{ind}} s' \doteq t'$ ならば $\mathcal{R} \models_{\mathrm{ind}} u \doteq v$.

(証明)　操作 G が健全であることは命題 3 により保証されているので，操作 M の健全性を示せばよい．$s \to t, s' \to$

入力：仮定集合 H_{n-1}, H_n
出力：補題集合 L

$$
\begin{array}{llll}
s & \to & t & \in H_{n-1} \\
s' & \to & C[t\sigma] & \in H_n \setminus H_{n-1} \quad (\sigma：変数改名) \\
\stackrel{M}{\Longrightarrow} & s' & \to & C[s\sigma] \\
\stackrel{G}{\Longrightarrow} & u & \doteq & v \qquad\qquad (x：新しい変数)
\end{array}
$$

図 8　健全発散鑑定法の手続き

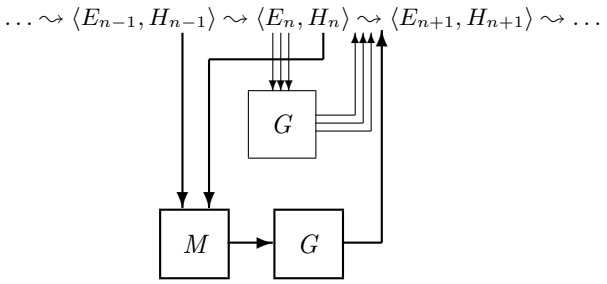

図 9　健全発散鑑定・健全一般化法付き書き換え帰納法 $MG{+}G$ の補題生成

$C[t\sigma]$ から M により生成される等式は，$s' \doteq C[s\sigma]$．このとき，$s \doteq t, s' \doteq t'$ を $\mathcal{R}$ の帰納的定理とすると，$s'\sigma_g \stackrel{*}{\leftrightarrow}_{\mathcal{R}} C[t\sigma]\sigma_g = C\sigma_g[t\sigma\sigma_g] \stackrel{*}{\leftrightarrow}_{\mathcal{R}} C\sigma_g[s\sigma\sigma_g] = C[s\sigma]\sigma_g$ より，$s' \doteq C[s\sigma]$ も $\mathcal{R}$ の帰納的定理である．　□

健全発散鑑定法と従来の健全一般化法を組み合わせることにより，両者の有効範囲を併せもち，反証機能にも対応可能な補題生成が実現できる．このような健全発散鑑定・健全一般化法付き書き換え帰納法 $MG{+}G$ の手続きの概要を図 9 で示す．

4.2　実験および考察

これまで議論してきた補題生成手法を組み込んだ書き換え帰納法の実装および実験を行った．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．本実験では，単相項書き換えシステム $\mathcal{R}$ を対象に，(1) 書き換え帰納法 RI，(2) 発散鑑定法付き書き換え帰納法 MC，(3) 健全一般化法付き書き換え帰納法 G，(4) 健全発散鑑定法付き書き換え帰納法 MG，(5) 健全発散鑑定・健全一般化法付き書き換え帰納法 $MG{+}G$ の証明能力および反証能力の比較を行った．

さらに，我々のシステムの有効性の指標として定理自動証明システム SPIKE と NICE との比較も行った．NICE は帰納的定理の証明手法として 2 つの機能をもつ．ひとつは健全一般化法付きの書き換え帰納法にもとづく手法 (NICE$_G$ と記す) と，もうひとつは項分割 (term partition)[14] とよばれる証明手法 (NICE$_P$ と記す) である．なお，NICE$_P$ は補題自動生成法を備えておらず，しかも単相項書き換えシステムのもとでの帰納的定理の自動証明に限定される．以下では，両者の機能をもつシステムを NICE と記す．

なお，我々の実験システムでは，等式論理上の帰納的定理の証明・反証のみを取り扱うが，SPIKE，NICE では，擬等式論理 (条件付等式やホーン節) 上の帰納的定理の証明・反証も可能であることに注意する．

システムの証明・反証能力を比較するため，RI で発散する 33 個の帰納的定理 (等式 1–33) と 2 個の非定理 (等式 34, 35) の証明を試みた．実験結果を表 1 に示す．反証書き換え帰納法の完全性 [5, 6] により，反証の成功は補題の追加と無関係なので，反証実験は動作確認を行うに留めた．システムの発散判定は，書き換え帰納法の手続きで得られる系列 $\langle E_0, H_0\rangle \stackrel{*}{\rightsquigarrow} \langle E_n, H_n\rangle$ が $E_{20} \neq \emptyset$ となるときとした．自然数上の加算・乗算，リスト結合およびリスト反転を項書き換えシステム $\mathcal{R}$ で与える．$\mathcal{R}$ は，合流性，停止性および十分完全性もっており，単相項書き換えシステムでもある．なお，$\mathcal{R}$ が合流性をみたせば，基底合流性をみたしていることに注意する．

$$
\mathcal{R} = \left\{
\begin{array}{lcl}
0 + y & \to & y \\
\mathrm{s}(x) + y & \to & \mathrm{s}(x + y) \\
0 * y & \to & 0 \\
\mathrm{s}(x) * y & \to & y + (x * y) \\
[\,] \mathbin{@} ys & \to & ys \\
(x :: xs) \mathbin{@} ys & \to & x :: (xs \mathbin{@} ys) \\
\mathrm{rev}([\,]) & \to & [\,] \\
\mathrm{rev}(x :: xs) & \to & \mathrm{rev}(xs) \mathbin{@} [x] \\
\mathrm{qrev}([\,], ys) & \to & ys \\
\mathrm{qrev}(x :: xs, ys) & \to & \mathrm{qrev}(xs, x :: ys)
\end{array}
\right.
$$

MC は 9 個の等式の証明に成功し，G は 19 個の等式の証明に成功した．G は等式 25, 33 を除いて NICE$_G$ と同じ結果となっている．等式 25, 33 の証明では両者は同じ補題を生成した．両者の証明能力に差が生じたのは，*Simplify* 規則での書き換え戦略と *Expand* 規則での等式の選択方法が異なるからである．MG は MC で成功した 9 個の等式のうち 8 個について証明に成功した．また，MC では発散した等式 17 の証明に成功した．これは MG の補題生成にもちいている健全発散鑑定法が健全一般化法に依存しているからと考えられる．$MG{+}G$ は，MG または G で成功したすべての等式の証明に成功した．

一方，SPIKE はすべての帰納的定理の証明で発散した．

表 1　補題自動生成法の比較 (○：定理，×：非定理，∞：発散，–：失敗，⊠：適用不可)

No	証明する等式	*RI*	*MC*	*G*	*MG*	*MG*+*G*	SPIKE	NICE$_G$	NICE$_P$
1	$x + \mathrm{s}(x) \doteq \mathrm{s}(x + x)$	∞	○	○	○	○	∞	○	○
2	$(x + x) + x \doteq x + (x + x)$	∞	∞	○	∞	○	∞	○	○
3	$(x + y) * z \doteq (x * z) + (y * z)$	∞	∞	○	∞	○	∞	○	○
4	$x * (y + z) \doteq (x * y) + (x * z)$	∞	∞	∞	∞	∞	∞	∞	–
5	$(x * y) * z \doteq x * (y * z)$	∞	∞	○	∞	○	∞	○	○
6	$\mathrm{rev}(\mathrm{rev}(xs)) \doteq xs$	∞	○	∞	○	○	∞	∞	○
7	$\mathrm{rev}(\mathrm{rev}(xs) \mathbin{@} [\,]) \doteq xs$	∞	○	∞	○	○	∞	∞	○
8	$\mathrm{rev}(\mathrm{rev}(xs \mathbin{@} [\,])) \doteq xs$	∞	○	∞	○	○	∞	∞	○
9	$\mathrm{rev}(\mathrm{rev}(xs \mathbin{@} ys)) \doteq xs \mathbin{@} ys$	∞	○	∞	○	○	∞	∞	○
10	$\mathrm{rev}(xs \mathbin{@} \mathrm{rev}(ys)) \doteq ys \mathbin{@} \mathrm{rev}(xs)$	∞	○	∞	○	○	∞	∞	○
11	$\mathrm{qrev}(\mathrm{rev}(xs), ys) \doteq xs \mathbin{@} ys$	∞	○	∞	∞	∞	∞	∞	–
12	$\mathrm{qrev}(xs, [\,]) \doteq \mathrm{rev}(xs)$	∞	∞	○	∞	○	∞	○	○
13	$\mathrm{qrev}(xs \mathbin{@} [\,], [\,]) \doteq \mathrm{rev}(xs)$	∞	∞	○	∞	○	∞	○	○
14	$\mathrm{qrev}(xs, ys) \doteq \mathrm{rev}(xs) \mathbin{@} ys$	∞	∞	○	∞	○	∞	○	○
15	$\mathrm{rev}(xs \mathbin{@} ys) \doteq \mathrm{rev}(ys) \mathbin{@} \mathrm{rev}(xs)$	∞	∞	○	∞	○	∞	○	○
16	$\mathrm{rev}(xs \mathbin{@} xs) \doteq \mathrm{rev}(xs) \mathbin{@} \mathrm{rev}(xs)$	∞	∞	○	∞	○	∞	○	○
17	$\mathrm{rev}(\mathrm{qrev}(xs, ys)) \doteq \mathrm{rev}(ys) \mathbin{@} xs$	∞	∞	○	○	○	∞	○	○
18	$\mathrm{qrev}(xs \mathbin{@} ys, [\,]) \doteq \mathrm{rev}(xs \mathbin{@} ys)$	∞	∞	○	∞	○	∞	○	○
19	$\mathrm{qrev}(xs, \mathrm{rev}(ys)) \doteq \mathrm{rev}(xs) \mathbin{@} \mathrm{rev}(ys)$	∞	∞	○	∞	○	∞	○	○
20	$\mathrm{qrev}(xs \mathbin{@} ys, [\,]) \doteq \mathrm{rev}(ys) \mathbin{@} \mathrm{rev}(xs)$	∞	∞	○	∞	○	∞	○	–
21	$\mathrm{qrev}(xs \mathbin{@} ys, zs) \doteq \mathrm{rev}(xs \mathbin{@} ys) \mathbin{@} zs$	∞	∞	○	∞	○	∞	○	○
22	$\mathrm{qrev}(\mathrm{qrev}(xs, ys), [\,]) \doteq \mathrm{rev}(ys) \mathbin{@} xs$	∞	∞	○	∞	○	∞	○	–
23	$\mathrm{qrev}(\mathrm{qrev}(xs, ys), zs) \doteq (\mathrm{rev}(ys) @ xs) @ zs$	∞	∞	○	∞	○	∞	○	–
24	$\mathrm{qrev}(\mathrm{qrev}(xs, ys), zs) \doteq \mathrm{rev}(ys) @ (xs @ zs)$	∞	∞	○	∞	○	∞	○	–
25	$\mathrm{rev}(\mathrm{rev}(xs \mathbin{@} ys)) \doteq \mathrm{rev}(\mathrm{rev}(xs)) \mathbin{@} ys$	∞	○	○	○	○	∞	∞	○
26	$\mathrm{rev}(\mathrm{rev}(xs \mathbin{@} ys)) \doteq xs \mathbin{@} \mathrm{rev}(\mathrm{rev}(ys))$	∞	○	○	○	○	∞	○	○
27	$\mathrm{rev}(\mathrm{rev}(xs)) \mathbin{@} [\,] \doteq xs$	∞	∞	∞	∞	∞	∞	∞	○
28	$\mathrm{qrev}(\mathrm{rev}(xs), [\,]) \doteq xs$	∞	∞	∞	∞	∞	∞	∞	–
29	$\mathrm{rev}(\mathrm{qrev}(xs, [\,])) \doteq xs$	∞	∞	∞	∞	∞	∞	∞	○
30	$\mathrm{qrev}(\mathrm{qrev}(xs, [\,]), [\,]) \doteq xs$	∞	∞	∞	∞	∞	∞	∞	–
31	$\mathrm{qrev}(\mathrm{qrev}(xs, [\,]), zs) \doteq xs \mathbin{@} zs$	∞	∞	∞	∞	∞	∞	∞	–
32	$\mathrm{rev}(\mathrm{rev}(xs) \mathbin{@} \mathrm{rev}(ys)) \doteq ys \mathbin{@} xs$	∞	∞	∞	∞	∞	∞	∞	–
33	$\mathrm{rev}(\mathrm{rev}(xs) \mathbin{@} ys) \doteq \mathrm{rev}(ys) \mathbin{@} xs$	∞	∞	∞	∞	∞	∞	○	○
34	$\mathrm{qrev}(xs, xs) \doteq \mathrm{rev}(xs)$	×	⊠	×	×	×	×	∞	–
35	$\mathrm{rev}(xs @ ys) \doteq ys @ xs$	×	⊠	×	×	×	×	∞	–

SPIKE に備わっている発散鑑定法で補題を生成しようと試みたが，補題を生成されるまでの計算時間が我々のシステムと比べて非常に大きいため，補題を生成することができなかった．*MC* は反証機能がないので, 非定理 (等式 34, 35) に対しては適用不可とした．NICE$_G$ および NICE$_P$ は反証機能をもつが，基底項上の等式の反証にしか適用できないため，反証に成功しなかった．

次に，帰納的定理の自動証明における標準的な例題集である Dream Corpus[*1] から例を抜粋し同様の実験を行った．Dream Corpus はエジンバラ大学の Bundy らのグループにより，Boyer-Moore[8] Corpus (Dmacs Corpus) から抜粋・自動変換により得られた 1000 個以上もの述語論理上の帰納的定理の例題集 (中に多数の重複を含む) である．そのうち，本論文で対象としている等式論理上の帰納的定理の例題は 69 個 (重複削除後) 含まれている．そのうち 33 個は *RI* で証明に成功した．失敗もしくは発散した 36 個の等式についての結果を表 2 に示す．なお，これらの例題では，単相項書き換えシステムが用いられている．表中の番号は Dream Corpus での例題番号を表わす．

我々のシステムの失敗例は，全て *Expand* 規則における等式の向き付け失敗によるものである．SPIKE，NICE は，簡約順序による向き付けに失敗した等式を取り扱うための拡張機能 [5, 6] を備えているが，我々のシステムにその拡張機能は組み込まれていない．*MG* は *MC* で成功し

[*1] Dream corpus of induction conjectures
`http://dream.inf.ed.ac.uk/dc/lib.html`

表 2　Dream Corpus(抜粋) による補題自動生成法の比較 (○：定理，∞：発散，–：失敗)

No	証明する等式	*RI*	*MC*	*G*	*MG*	*MG*+*G*	SPIKE	NICE$_G$	NICE$_P$
3	$x+(y+z) \doteq y+(x+z)$	–	–	–	–	–	○	○	–
4	$x+y \doteq y+x$	–	–	–	–	–	○	○	–
25	$\mathrm{rev}(xs \mathbin{@} ys) \doteq \mathrm{rev}(ys) \mathbin{@} \mathrm{rev}(xs)$	∞	∞	○	∞	○	∞	○	○
27	$x*(y+z) \doteq (x*y)+(x*z)$	∞	∞	∞	∞	∞	∞	∞	–
28	$x*\mathrm{s}(y) \doteq x+(x*y)$	–	–	–	–	–	∞	∞	–
29	$x*y \doteq y*x$	–	–	–	–	–	∞	∞	–
30	$x*(y*z) \doteq y*(x*z)$	–	–	–	–	–	∞	∞	–
31	$(x*y)*z \doteq x*(y*z)$	∞	∞	○	∞	○	∞	○	○
43	$\mathrm{rev}(\mathrm{rev}(x :: xs)) \doteq x :: xs$	∞	○	∞	○	○	∞	∞	○
47	$\mathrm{len}(\mathrm{rev}(xs)) \doteq \mathrm{len}(xs)$	∞	○	∞	∞	∞	∞	∞	–
60	$\mathrm{dbl}(i) \doteq \mathrm{s}(\mathrm{s}(0))*i$	∞	○	○	○	○	∞	○	○
63	$\mathrm{last}(\mathrm{rev}(x :: xs)) \doteq x :: [\,]$	∞	∞	∞	∞	∞	∞	∞	–
64	$\exp(i, j+k) \doteq \exp(i,j)*\exp(i,k)$	∞	∞	∞	∞	∞	∞	○	–
79	$\mathrm{factloop}(j,i) \doteq i*\mathrm{fact}(j)$	∞	∞	∞	∞	∞	∞	∞	–
80	$\mathrm{fact2}(i) \doteq \mathrm{fact}(i)$	∞	∞	∞	∞	∞	∞	∞	–
81	$\mathrm{qrev}(xs, ys) \doteq \mathrm{rev}(xs) \mathbin{@} ys$	∞	∞	○	∞	○	∞	○	○
82	$\mathrm{qrev}(xs, [\,]) \doteq \mathrm{rev}(xs)$	∞	∞	○	∞	○	∞	○	○
83	$\mathrm{rev2}(xs, ys) \doteq \mathrm{rev2}(ys) \mathbin{@} \mathrm{rev2}(xs)$	∞	∞	–	∞	–	∞	○	–
84	$\mathrm{rev2}(\mathrm{rev2}(x :: xs)) \doteq x :: xs$	∞	∞	∞	∞	∞	∞	∞	–
109	$\mathrm{nth}(xs@ys, i) \doteq \mathrm{nth}(xs,i)@\mathrm{nth}(ys, i-\mathrm{len}(xs))$	–	–	–	–	–	○	∞	–
111	$(x+y)-y \doteq y$	∞	∞	∞	∞	∞	∞	∞	–
113	$x*(y-z) \doteq (y*x)-(z*x)$	–	–	–	–	–	∞	∞	–
116	$\mathrm{s}(x+y)-y \doteq \mathrm{s}(x)$	∞	∞	∞	∞	∞	∞	∞	–
158	$\mathrm{timeslist}(xs@ys) \doteq \mathrm{timeslist}(xs)*\mathrm{timeslist}(ys)$	∞	∞	○	○	○	∞	○	–
232	$\mathrm{s}(\mathrm{s}(0))*x \doteq x+x$	∞	∞	○	∞	○	∞	○	○
236	$\exp(i*j,k) \doteq \exp(i,k)*\exp(j,k)$	–	–	–	–	–	∞	∞	–
237	$\exp(\exp(i,j),k) \doteq \exp(i, j*k)$	∞	∞	∞	∞	∞	∞	∞	–
270	$\mathrm{s}(\mathrm{s}(\mathrm{s}(\mathrm{s}(0))))*x \doteq \mathrm{s}(\mathrm{s}(0))*(\mathrm{s}(\mathrm{s}(0))*x)$	∞	∞	○	∞	○	∞	○	○
300	$\mathrm{h}(x,\mathrm{h}(y,z)) \doteq \mathrm{h}(y,\mathrm{h}(x,z))$	–	–	–	–	–	○	○	–
301	$\mathrm{ach}(xs,i) \doteq \mathrm{prh}(xs,i)$	–	–	–	–	–	∞	∞	–
318	$\mathrm{fn2}(x,y) \doteq \mathrm{fn2}(y,x)$	–	–	–	–	–	○	○	–
319	$\mathrm{foldrfn}(xs,i) \doteq \mathrm{foldlfn}(\mathrm{rev}(xs),i)$	∞	–	∞	∞	∞	∞	○	–
370	$(x*x)-\mathrm{s}(0) \doteq \mathrm{s}(x)*\mathrm{sub}(x)$	∞	∞	∞	∞	∞	∞	∞	–
375	$\mathrm{sub}((x-\mathrm{s}(\mathrm{s}(0)))*(x-\mathrm{s}(\mathrm{s}(0)))) \doteq (x-\mathrm{s}(\mathrm{s}(\mathrm{s}(0))))*\mathrm{sub}(x)$	∞	∞	∞	∞	∞	∞	○	–
694	$\mathrm{timesfn}(i,j,\mathrm{ans}) \doteq (i*j)+\mathrm{ans}$	–	–	–	–	–	∞	○	–
1052	$(x+y)*z \doteq (x*z)+(y*z)$	∞	∞	○	∞	○	∞	○	○

た 3 個の等式のうち 2 個について証明に成功し，*MC* では発散した等式 158 の証明に成功した．NICE$_G$ と *G* の間で差があったのは，向き付け失敗による違いを除くと等式 64, 319 であるが，NICE$_G$ の証明出力をチェックしたところ出力された証明に間違いがあり，この結果は NICE$_G$ の実装上のバグによるものを考えられる．等式 64, 319 も除外すると，*G* と NICE$_G$ は同じ結果となり，*MG*+*G* と NICE も同じ結果が得られた．

以上の考察から，*MG*+*G* は *RI*, *MC*, *G*, *MG*, SPIKE よりも強力な証明・反証能力をもつ．さらに，証明能力に関しては NICE とほぼ同等であり．反証能力は優れている．したがって，証明および反証機能の両面を考慮すると，我々の提案した *MG*+*G* は，これらのシステムのうち，最も強力な帰納的定理自動証明システムであるといえる．

5　おわりに

本論文では，従来から知られていた補題自動生成法である発散鑑定法の部分項消去操作を健全一般化法に置き換えることによって，健全発散鑑定法という新しい補題自動生成法を提案した．ここで提案された健全発散鑑定法は，従来の発散鑑定法とほぼ同等の補題生成能力をもちながら，発散鑑定法には欠けていた健全性が保証されている．したがって，これまで発散鑑定法と両立させることができなかった反証機能付き書き換え帰納法にも組み込むことが可能である．実際，健全発散鑑定法と従来の健全一般化法を

組み入れた健全発散鑑定・健全一般化法付き書き換え帰納法は，反証機能を損なわずに優れた証明能力をもつことが実験で明らかになった．自動証明と自動反証の両方に対して有効な補題自動生成法は，これまで健全一般化法しか知られておらず，本論文の結果は，強力な反証機能付き自動証明システムを実現する基礎技術としてきわめて有効であると考える．ここで得られた成果にもとづいて，強力な帰納的定理自動証明システムを開発することは今後の課題である．

なお，本論文では，最も基本的な発散鑑定法に対象を限定して考察を行なったが，文献 [16] の発散鑑定法では，他にもさまざまなヒューリスティクスを用いた補題生成法が提案されている．例えば，2 つの関連発散系列の差分情報の利用や，アキュムレータとの差分照合などである．これらのヒューリスティクスを参考にして，より強力かつ健全な発散鑑定法を提案することも今後の検討課題である．

謝辞

本論文を改善するための貴重なコメントを頂きました査読者に感謝致します．なお，本研究は一部日本学術振興会科学研究費 17700002, 19500003 の補助を受けて行われた．

参考文献

[1] T. Aoto, “Dealing with non-orientable equations in rewriting induction,” Proc. RTA’06, LNCS, vol.4098, pp.242–256, Springer-Verlag, 2006.

[2] T. Aoto, “Soundness of rewriting induction based on an abstract principle,” IPSJ Trans. on Prog., to appear.

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

[4] D. Basin and T. Walsh, “Difference matching,” Proc. CADE-11, LNCS, vol.607, pp.295–309, Springer-Verlag, 1992.

[5] A. Bouhoula, E. Kounalis, and M. Rusinowitch, “Automated mathematical induction,” J. Logic and Comput., vol.5, no.5, pp.631–668, 1995.

[6] A. Bouhoula, “Automated theorem proving by test set induction,” J. Symbolic Comput., vol.23, pp.47–77, 1997.

[7] A. Bundy, D. Basin, D. Hutter, and A. Ireland, *Rippling: Meta-Level Guidance for Mathematical Reasoning*, Cambridge University Press, Cambridge, 2005.

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

[9] G. Huet and D. Oppen. “Equations and rewrite rules: a survey,” In Formal Languages: Perspectives and Open Problems, Ed. Book R., Academic Press, 1980, pp. 349-405.

[10] D. Kapur, P. Narendran, and H. Zhang, “On sufficient-completeness and related properties of term rewriting systems,” Acta Inf., vol.24, no.4, pp.395-415, 1987.

[11] 小池宏高，外山芳人，“潜在帰納法と書き換え帰納法の比較，” コンピュータソフトウェア，vol.17，no.6，pp.1–12，2000.

[12] U. S. Reddy, “Term rewriting induction,” Proc. CADE-10, LNCS, vol.449, pp.162–177, Springer-Verlag, 1990.

[13] Y. Toyama, “How to prove equivalence of term rewriting systems without induction,” Theor. Comput. Sci., vol.90, no.2, pp.369–390, 1991.

[14] P. Urso and E. Kounalis, “Term partition for mathematical induction,” Proc. RTA’03, LNCS, vol.2706, pp.352–366, Springer-Verlag, 2003.

[15] P. Urso and E. Kounalis, “Sound generalizations in mathematical induction,” Theor. Comput. Sci., vol.323, pp.443–471, 2004.

[16] T. Walsh, “A divergence critic for inductive proof,” J. Artif. Intell. Res., vol.4, pp.209–235, 1996.