# 条件付き項書き換えシステムにおける ホーン節帰納的定理の自動証明

栗田 泰智, 青戸 等人

新潟大学大学院自然科学研究科

{ kurita@nue., aoto@ }ie.niigata-u.ac.jp

**概 要**　項書き換えシステムの重要な特徴付けの一つに帰納的定理があり，その自動証明法として書き換え帰納法が知られている．従来知られている基本的な書き換え帰納法は，条件なしの項書き換えシステムと等式帰納的定理を対象としている．本論文では，まず，項書き換えシステムだけでなく，条件付き項書き換えシステムが扱えるようにアンラベリング変換を導入する．次に，帰納的定理についても，等式だけでなく，等式のホーン節を扱えるように書き換え帰納法の拡張を行う．また，提案手法の正当性を証明するとともに，提案手法に基づく帰納的定理自動証明システムを実装する．実装システムを用いた帰納的定理証明の実験を行い，その結果を報告する．

## 1　はじめに

帰納的定理は，項書き換えシステムにおける重要な論理的妥当性の一つとして知られている [?]. 帰納的定理は，直観的には項書き換えシステムをプログラムとして見たときに正しい式に対応し，代数仕様記述や項書き換えシステムに基づくプログラム検証などにおいて重要な概念である．帰納的定理の自動証明法の 1 つとして書き換え帰納法 [?] が知られている．理論的基盤がよく調べられている基本的な書き換え帰納法は，項書き換えシステムと等式帰納的定理を対象としている [?, ?, ?, ?].

一方，実際のプログラムのモデル化に際しては条件付きの書き換え規則による計算システムである条件付き項書き換えシステム [?] が重要である．また，帰納的定理についても，等式だけでなく，条件付き等式 (等式のホーン節) についても考えることができ，条件付き等式を用いることでより広範囲な性質の記述が可能となる．実際，条件付き項書き換えシステムや条件付き等式に対する書き換え帰納法も提案されている [?, ?]．しかしながら，これらの先行研究は，非常に複雑な体系を用いているばかりか，その理論的基盤の検証が困難であり，その後，これらの体系を基礎とした研究はほとんど行われていない．

本論文では，まず，条件付き項書き換えシステムから項書き換えシステムへの変換法であるアンラベリング変換 [?] を利用して，条件付き項書き換えシステムに対する帰納的定理証明を扱うことを考える．アンラベリング変換の正当性が保証される条件付き項書き換えシステムのクラス [?, ?] と，条件付き項書き換えシステムにおいて計算モデルとしての性質と論理上の性質の対応がつくクラス [?] とを対応付けることで，アンラベリング変換を用いて帰納的定理証明を扱えるクラスを明らかにする．次に，帰納的定理についても，等式だけでなく，等式のホーン節を扱えるように書き換え帰納法の拡張を行い，その正当性を証明する．そして，この両者を結び付けることで，条件付き項書き換えシステムに対する，ホーン節帰納的定理の自動証明を実現する．また，提案手法に基づく帰納的定理自動証明システムを実装するとともに，帰納的定理証明の実験を行い，その有効性や問題点を考察する．

本論文の構成は次のとおりである．第 2 節では本論文で通して用いる基本概念や記法について説明する．第 3 節では条件付き項書き換えシステムにおける等式帰納的定理証明について説明する．第 4 節ではホーン節帰納的定理の自動証明について説明する．第 5 節では実装した帰納的定理自動証明システムについて説明し，その実験結果を示す．第 6 節は結論である．

## 2　準備

本節では，本論文で通して用いる基本概念や記法について説明する．

### 2.1　条件付き項書き換えシステム

2 項関係 $\to$ の**反射推移閉包**を $\to^*$，**等価閉包**を $\leftrightarrow^*$ と記す．関係 $\to$ が**合流性**をもつとは，$x \leftarrow^* y \to^* z$ に対して，$x \to^* u \leftarrow^* z$ となる $u$ が存在するときをいう．関係 $\to$ が**停止性**をもつとは，$x_0 \to x_1 \to \cdots$ なる無限列が存在しないときをいう．関係 $\to$ が**完備**であるとは，関係 $\to$ が合流性と停止性をもつことをいう．要素 $a$ が**正規形**であるとは，$a \to b$ となる $b$ が存在しないときをいう．$a \to^* b$ なる正規形 $b$ を $a$ の正規形とよび，任意の要素がその正規形をもつとき，関係 $\to$ は**弱正規化可能**であるという．

$\mathcal{S}$ を**ソート集合**としたとき，$\mathcal{S}$ 上の**多ソートシグニチャ**とは，**関数記号集合** $\mathcal{F}$ で，それぞれの関数記号 $f \in \mathcal{F}$ に，$sort(f) = \tau_1 \times \cdots \times \tau_n \to \tau_0$ $(\tau_0, \ldots, \tau_n \in \mathcal{S}, n \geq 0)$ が定まっているものをいう．$\mathcal{S}$ をソート集合，$\mathcal{F}$ を $\mathcal{S}$ 上の多ソートシグニチャとする．このとき，ソートのそれぞれに対して，そのソートの**変数集合**を用意する．ソート $\tau$ の変数集合を $\mathcal{V}^\tau$ と記し，$\mathcal{S}$ 上の変数集合を $\mathcal{V} = \uplus_{\tau \in \mathcal{S}} \mathcal{V}^\tau$ とおく．このときすべての $\tau \in \mathcal{S}$ について，ソート $\tau$ の**項**の集合 $\mathcal{T}(\mathcal{F}, \mathcal{V})^\tau$ を以下のように帰納的に定義する：(1) $x \in \mathcal{V}^\tau$ ならば $x \in \mathcal{T}(\mathcal{F}, \mathcal{V})^\tau$，(2) $sort(f) = \tau_1 \times \cdots \times \tau_n \to \tau_0$，それぞれの $1 \leq i \leq n$ について $t_i \in \mathcal{T}(\mathcal{F}, \mathcal{V})^{\tau_i}$ ならば，$f(t_1, \ldots, t_n) \in \mathcal{T}(\mathcal{F}, \mathcal{V})^{\tau_0}$．また，項の集合を，$\mathcal{T}(\mathcal{F}, \mathcal{V}) = \uplus_{\tau \in S} \mathcal{T}(\mathcal{F}, \mathcal{V})^\tau$ とおく．項上の**等式** $l \approx r$ は $sort(l) = sort(r)$ を満たすものとし，左辺と右辺を区別しない場合は，$l \approxeq r$ と記す．

項 $t$ の**根記号** $root(t)$ とは $t \in \mathcal{V}$ のとき $t$，$t = f(t_1, \ldots, t_n)(f \in \mathcal{F})$ のとき $f$ である．項 $t$ に現れる変数全体の集合を $\mathcal{V}(t)$ と記す．項の列や等式に現れる変数についても同様に $\mathcal{V}(\ldots)$ と記す．項 $t$ に変数 $x$ が現れる回数を $|t|_x$ と記し，項の列や等式に現れる回数についても同様に $|\ldots|_x$ と記す．$\mathcal{V}(t) = \emptyset$ のとき項 $t$ を**基底項**という．**文脈**とは特別な定数 $\Box^\tau$(**ホール**) を丁度 1 つ持つ項のことである．ただし，$\tau \in \mathcal{S}$ で，$sort(\Box^\tau) = \tau$ であるものとする．$C[t]$ 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 $C$ が存在するときをいい，$t \trianglerighteq u$ と記す．項 $t$ の部分項 $u$ が $t$ と異なるとき，$u$ を**真部分項**とよび，$t \triangleright u$ と記す．

関数 $\sigma : \mathcal{V} \to \mathcal{T}(\mathcal{F}, \mathcal{V})$ で，$\forall x \in \mathcal{V}.\ sort(\sigma(x)) = sort(x)$ かつ集合 $\{x \mid \sigma(x) \neq x\}$ が有限であるものを**代入**とよぶ．代入 $\sigma$ 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 $\sigma$ で表し，$\sigma(t)$ を $t\sigma$ と記す．$t\sigma$ が基底項になるとき，これを $t$ の**基底具体化**という．等式 $s \approx t$ の基底具体化も同様に定める．項 $s$ と $t$ の**最汎単一化子**を，$mgu(s, t)$ と記す．文脈及び代入に閉じた項上の関係 $>$ を，**書き換え関係**という．整礎な書き換え関係を**簡約関係**という．特に，簡約関係が半順序 (非反射的かつ推移的) のとき，**簡約順序**という．簡約順序 $>$ で $\triangleright \subseteq >$ を満たすものを，**単純化順序**とよぶ．

$l \approx r$ を等式，$c$ を $s_1 \approx t_1, \ldots, s_k \approx t_k$ なる等式の列とするとき，$c \Rightarrow l \approx r$ を**条件付き等式**とよぶ．条件付き等式 $c \Rightarrow l \approx r$ が**条件付き書き換え規則**であるとは，$l \notin \mathcal{V}$ を満たすときをいい，$l \to r \Leftarrow c$ と記す．$c$ を条件付き書き換え規則の**条件部**とよぶ．条件付き書き換え規則の有限集合を**条件付き項書き換えシステム** (略して CTRS) とよぶ．条件付き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．定義記号の集合を $\mathcal{D}$ と記す．**構成子記号**の集合を $\mathcal{C} = \mathcal{F} \setminus \mathcal{D}$ と定義する．定義記号を含まない項を**構成子項**とよび，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 $\mathcal{B}(\mathcal{D}, \mathcal{C}, \mathcal{V})$ と記し，項 $s$ の**基本部分項**の集合を $\mathcal{B}(s)$ と記す．条件付き書き換え規則 $l \to r \Leftarrow c$ が**左線形**であるとは，$l$ にはどの変数もたかだか 1 回のみ現れることをいう．CTRS $R$ が左線形であるとは，CTRS $R$ 中のすべての条件付き書き換え規則が左線形であるときをいう．CTRS $R$ のすべての条件付き書き換え規則 $l \to r \Leftarrow c$ について $\mathcal{V}(r) \subseteq \mathcal{V}(l) \cup \mathcal{V}(c)$ となるときに，$R$ は 3 **型**であるという．3 型の CTRS $R$ において，どの書き換え規則も条件を持たないとき，$R$ を

項書き換えシステム (略して TRS) とよぶ．条件付き書き換え規則の条件部をどのように解釈して書き換えを定義するかによって，CTRS はいくつかの種類に分けられる．CTRS $R$ における**書き換え関係** $\to_R$ を以下のように定義するとき，$R$ を**指向式 CTRS** とよぶ：

$$\begin{aligned}
\overset{0}{\to}_R &= \emptyset \\
\overset{n+1}{\to}_R &= \{(C[l\sigma], C[r\sigma]) \mid l \to r \Leftarrow s_1 \approx t_1, \ldots, s_k \approx t_k \in R, \forall i.\, s_i\sigma \overset{n_*}{\to}_R t_i\sigma)\} \\
\to_R &= \bigcup_{n\in\mathbb{N}} \overset{n}{\to}_R
\end{aligned}$$

項が正規形であるとは，関係 $\to_R$ について正規形であるときをいい，任意の $x \in \mathcal{V}$ について $\sigma(x)$ が正規形であるとき，代入 $\sigma$ を**正規代入**という．CTRS $R$ が合流性 (停止性，完備性，弱正規化可能性) を持つとは，関係 $\to_R$ がそれらの性質をもつときをいう．3 型指向式 CTRS が**決定的**であるとは，$R$ のすべての条件付き書き換え規則 $l \to r \Leftarrow s_1 \approx t_1, \ldots, s_k \approx t_k$ に対して，$\mathcal{V}(s_i) \subseteq \mathcal{V}(l) \cup \bigcup_{j=1}^{i-1} \mathcal{V}(t_j)$ $(1 \le i \le k)$ が成り立つときをいう．**決定的な 3 型指向式 CTRS を，以下では，3DCTRS と略す**．

## 2.2 アンラベリング変換

指向式 CTRS から TRS への変換として，アンラベリング変換 [**?**] が知られている．アンラベリング変換は，直観的には，条件の評価を再帰的に行うことで項を書き換えていく操作を，TRS の書き換え規則でエンコードしたものとなっており，元の CTRS $R$ とアンラベリング変換により得られる TRS $U(R)$ の等価性の十分条件が知られている [**?**, **?**]．なお，アンラベリング変換にはいくつかのバージョンが知られているが，本論文で用いるのは，文献 [**?**] で導入され，文献 [**?**] で $U_{\mathsf{conf}}$ と表記されているものである．

まず，準備として，変数集合 $\mathcal{V}$ については，その上である全順序 $>$ が定まっているものとし，変数の有限集合 $X = \{x_1, \ldots, x_n\}$ について，$x_1 < \cdots < x_n$ であるとき，$\vec{X}$ で変数列 $x_1, \ldots, x_n$ を表わすものとする．また，このとき，例えば $f(s, \vec{X})$ で項 $f(s, x_1, \ldots, x_n)$ を表わすものとする．

また，CTRS $R$ の書き換え規則には，それぞれ自然数 $1, \ldots, |R|$ がインデックスとして付けられているものとして，条件付き書き換え規則 $l \to r \Leftarrow s_1 \approx t_1, \ldots, s_k \approx t_k$ がインデックス $\rho$ をもつことを $\rho:\ l \to r \Leftarrow s_1 \approx t_1, \ldots, s_k \approx t_k$ と記し，$\rho$ と $l \to r \Leftarrow s_1 \approx t_1, \ldots, s_k \approx t_k$ を同一視する．

条件書き換え規則 $\rho:\ l \to r \Leftarrow s_1 \approx t_1, \ldots, s_k \approx t_k$ に対して，アンラベリング変換により得られる $k+1$ 個の書き換え規則の集合 $U(\rho)$ を次のように定義する．

$$U(\rho) = \{l \to U_1^\rho(s_1, \vec{Z_1}), U_1^\rho(t_1, \vec{Z_1}) \to U_2^\rho(s_2, \vec{Z_2}), \ldots, U_k^\rho(t_k, \vec{Z_k}) \to r\}$$

ここで，$U_1^\rho, \ldots, U_k^\rho$ は，変換の過程で導入する新しい関数記号を表わしており，これを $U$ **記号**とよぶ．$U$ 記号全体の集合を $\mathcal{U}$ と記す．各規則で導入する $U$ 記号は，以下で説明するように，異なる条件付き書き換え規則 $\rho, \delta$ において $U_i^\rho = U_i^\delta$ となる場合があることに注意する．変数集合 $Z_i$ $(1 \le i \le k)$ は，$Z_i = \mathcal{V}(l, t_1, \ldots, t_{i-1})$ により与えられる．なお，$k = 0$ の場合は，$U(l \to r) = \{l \to r\}$ と定める．CTRS $R$ に対するアンラベリング変換の結果を $U(R) = \bigcup_{\rho\in R} U(\rho)$ により定義する．ただし，2 つの条件付き書き換え規則 $\rho : l \to r \Leftarrow s_1 \approx t_1, \ldots, s_k \approx t_k$，$\rho' : l' \to r' \Leftarrow s'_1 \approx t'_1, \ldots, s_m \approx t_m$ について，$l = l', s_1 = s'_1, t_1 = t'_1, \ldots, s_{i-1} = s'_{i-1}, t_{i-1} = t'_{i-1}, s_i = s'_i$ が成立するとき，$U$ 記号 $U_i^\rho$ と $U_i^{\rho'}$ は同一の関数記号を用いるものとする．また，$R$ が 3DCTRS ならば $U(R)$ は TRS となることに注意する．

**例 2.1 (アンラベリング変換)** 以下の $R$ は，$\mathcal{S} = \{Nat, NatList, ListPair, Bool\}$ 上の 3DCTRS で

あり，クイックソートを行う．なお，関数記号のソートについては記載を省略する．

$$R = \left\{ \begin{array}{lcll} 0 \leq x & \to & true & \\ s(x) \leq 0 & \to & false & \\ s(x) \leq s(y) & \to & x \leq y & \\ [\,]@xs & \to & xs & \\ (x : xs)@ys & \to & x : (xs@ys) & \\ split(x, [\,]) & \to & \langle [\,], [\,] \rangle & \\ split(x, y : ys) & \to & \langle xs, y : zs \rangle & \Leftarrow \quad split(x, ys) \approx \langle xs, zs \rangle, x \leq y \approx true \quad (\rho_1) \\ split(x, y : ys) & \to & \langle y : xs, zs \rangle & \Leftarrow \quad split(x, ys) \approx \langle xs, zs \rangle, x \leq y \approx false \quad (\rho_2) \\ qs([\,]) & \to & [\,] & \\ qs(x : xs) & \to & qs(ys)@(x : qs(zs)) & \Leftarrow \quad split(x, xs) \approx \langle ys, zs \rangle \quad (\rho_3) \end{array} \right\}$$

$R$ に対してアンラベリング変換を適用すると

$$U(R) = \left\{ \begin{array}{lcll} 0 \leq x & \to & true & \\ s(x) \leq 0 & \to & false & \\ s(x) \leq s(y) & \to & x \leq y & \\ [\,]@xs & \to & xs & \\ (x : xs)@ys & \to & x : (xs@ys) & \\ split(x, [\,]) & \to & \langle [\,], [\,] \rangle & \\ split(x, y : ys) & \to & U_0(split(x, ys), x, y, ys) & (a) \\ U_0(\langle xs, zs \rangle, x, y, ys) & \to & U_1(x \leq y, x, y, ys, xs, zs) & (b) \\ U_1(true, x, y, ys, xs, zs) & \to & \langle xs, y : zs \rangle & (c) \\ U_1(false, x, y, ys, xs, zs) & \to & \langle y : xs, zs \rangle & (d) \\ qs([\,]) & \to & [\,] & \\ qs(x : xs) & \to & U_2(split(x, xs), x, xs) & (e) \\ U_2(\langle ys, zs \rangle, x, xs) & \to & qs(ys)@(x : qs(zs)) & (f) \end{array} \right\}$$

と変換できる．条件部をもつ書き換え規則 $\rho_1, \rho_2, \rho_3 \in R$ に対するアンラベリング変換は以下のとおりになっている：

$$\begin{aligned} U(\rho_1) &= \{(a), (b), (c)\} \\ U(\rho_2) &= \{(a), (b), (d)\} \\ U(\rho_3) &= \{(e), (f)\} \end{aligned}$$

さらに，$U_0 = U_1^{\rho_1} = U_1^{\rho_2}$，$U_1 = U_2^{\rho_1} = U_2^{\rho_2}$，$U_2 = U_1^{\rho_3}$ となっていることに注意する．

任意の項 $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ について，$s \xrightarrow{*}_R t$ ならば $s \xrightarrow{*}_{U(R)} t$ が成立するとき，アンラベリング変換は**完全**であるといい，任意の項 $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ について $s \xrightarrow{*}_{U(R)} t$ ならば $s \xrightarrow{*}_R t$ が成立するとき，アンラベリング変換は**健全**であるという．完全性と健全性について，以下の結果が知られている．

**命題 2.2 (アンラベリングの健全性と完全性 [?]，定理 10&13)** $R$ を 3DCTRS とするとき，$U(R)$ が左線形であれば，アンラベリング $U$ は健全かつ完全である．

### 2.3 3DCTRS の性質

次に，本論文で用いる CTRS の停止性，論理性，合流性に関する条件を説明する．

$>$ を文脈に閉じた整礎半順序とする．指向式 CTRS $R$ が $>$ について**擬還元的**であるとは，$R$ のすべての条件付き書き換え規則 $l \to r \Leftarrow s_1 \approx t_1, \ldots, s_k \approx t_k$ に対して，(1) $\forall j < i.\ s_j\sigma \geq t_j\sigma$ なる任意の代入 $\sigma$ について，$l\sigma (> \cup \rhd)^+ s_{i+1}\sigma$，(2) $\forall j \leq k.\ s_j\sigma \geq t_j\sigma$ なる任意の代入 $\sigma$ について，$l\sigma > r\sigma$，が成立するときをいう．ただし，ここで，$\geq$ は $>$ の反射閉包を表わす．

擬還元的な指向式 CTRS $R$ は停止性をもち，その書き換え関係 $\to_R$ が決定可能な関係となることが知られている．

**命題 2.3 (3DCTRS の擬還元性 [?]，定理 11)** $R$ を 3DCTRS とする．$U(R)$ が停止性をもつとき，$R$ は簡約順序 $> = \to^+_{U(R)}$ について擬還元的である．

次に，本論文で用いる CTRS の合流性に関する条件を説明する．

3DCTRS $R$ が**弱左線形**であるとは，任意の $l \to r \Leftarrow s_1 \approx t_1, ..., s_k \approx t_k \in R$ について，すべての $x \in \mathcal{V}$ について $|l, t_1, \ldots, t_k|_x > 1 \Rightarrow x \notin \mathcal{V}(s_1, \ldots, s_k, r)$ が成立するときをいう．

**命題 2.4 ([?]，定理 9)** 3DCTRS $R$ が弱左線形であるとき，$U(R)$ が合流性をもつならば，$R$ は合流性をもつ．

条件付き等式の集合 $E$ から導出される (条件付き) 等式は，以下の推論規則で与えられる．

$$\frac{u_1 \approx v_1, \ldots, u_n \approx v_n \Rightarrow l \approx r \in E}{u_1\theta \approx v_1\theta, \ldots, u_n\theta \approx v_n\theta \Rightarrow l\theta \approx r\theta} \quad \frac{}{s \approx s} \quad \frac{t \approx s}{s \approx t}$$

$$\frac{t \approx s \quad s \approx u}{t \approx u} \quad \frac{s_1 \approx t_1 \quad \cdots \quad s_n \approx t_n}{f(s_1, \ldots, s_n) \approx f(t_1, \ldots, t_n)}$$

$$\frac{u_1 \approx v_1, u_2 \approx v_2, \ldots, u_n \approx v_n \Rightarrow l \approx r \quad u_1 \approx v_1}{u_2 \approx v_2, \ldots, u_n \approx v_n \Rightarrow l \approx r}$$

条件付き等式集合 $E$ から等式 $s \approx t$ が導出されるとき，$E \vdash s \approx t$ と記す．また，CTRS $R$ について，$\{c \Rightarrow l \approx r \mid l \to r \Leftarrow c \in R\} \vdash s \approx t$ であるとき，$R \vdash s \approx t$ と記す．$R \vdash s \approx t$ となるとき，等式 $s \approx t$ は $R$ の**定理**であるという．TRS $R$ については，$R \vdash s \approx t$ と $s \leftrightarrow^*_R t$ が同値になることは容易に確かめられる．一方，指向式 CTRS $R$ について，$R \vdash s \approx t$ となることと $s \leftrightarrow^*_R t$ となることは，一般には一致しない．$R \vdash s \approx t$ と $s \leftrightarrow^*_R t$ が同値となるとき，指向式 CTRS $R$ は**論理性**をもつという．指向式 CTRS $R$ が論理性をもつための十分条件がいくつか知られている [?].

項 $t$ が，CTRS $R$ において**強簡約不能**であるとは，任意の正規代入 $\sigma$ について，$t\sigma$ が正規形となるときをいう．CTRS $R$ が強簡約不能であるとは，$R$ の条件付き書き換え規則における条件部のすべての右辺が $R$ において強簡約不能であるときをいう．指向式 CTRS の論理性について，以下の結果が知られている．

**命題 2.5 (指向式 CTRS の論理性 [?]，定理 15)** $R$ を指向式 CTRS とする．$R$ が合流性，弱正規化可能性，および，強簡約不能性をもつならば，$R$ は論理性をもつ．

**系 2.6** $R$ を弱左線形かつ強簡約不能な 3DCTRS とし，$U(R)$ が完備な左線形項書き換えシステムであるとする．このとき，$U$ は健全かつ完全，かつ，$R$ は論理性をもつ．

**(証明)** まず，$R$ が 3DCTRS であり，$U(R)$ が左線形であるから，命題 2.2 より，$U$ は健全かつ完全である．また，$U(R)$ の停止性と命題 2.3 より，$R$ は擬還元的．よって，$R$ は停止性をもち，弱正規化可能性をもつ．また，$R$ が弱左線形 3DCTRS であり，$U(R)$ が合流性をもつので，命題 2.4 より，$R$ は合流性をもつ．よって，仮定より $R$ は強簡約不能性をもつので，命題 2.5 より，$R$ は論理性をもつ． □

定理 3.2 の証明のために必要な補題について説明する．

**補題 2.7** $R$ をシグニチャ $\mathcal{F}$ 上の 3DCTRS で論理性をもつとする．また $R$ に対するアンラベリング $U$ が健全かつ完全であるものとする．$\mathcal{F}$ 上の任意の等式 $s \approx t$ について，$U(R) \vdash s \approx t \Leftrightarrow R \vdash s \approx t$.

**(証明)** ($\Rightarrow$) $U(R)$ は TRS なので，$U(R) \vdash s \approx t$ ならば，$s \leftrightarrow^*_{U(R)} t$ が成立．このとき，$U$ の健全性より，$s \leftrightarrow^*_R t$ が成立する．よって，$R$ の論理性より，$R \vdash s \approx t$. ($\Leftarrow$) $R \vdash s \approx t$ とする．このとき，$R$ の論理性より，$s \leftrightarrow^*_R t$ が成立する．よって，$U$ の完全性より，$s \leftrightarrow^*_{U(R)} t$. したがって，$R$ の論理性より，$U(R) \vdash s \approx t$. □

以下，本論文では，$U(R)$ が完備な左線形項書き換えシテムとなっているような弱左線形かつ強簡約不能な 3DCTRS を対象とする．

## 3　条件付き項書き換えシステムにおける等式帰納的定理証明

本節では CTRS における等式帰納的定理証明について説明する．まずは，帰納的定理について説明する．

等式 $s \approx t$ が，CTRS $R$ の**帰納的定理**であるとは，$s \approx t$ の任意の基底具体化 $s\sigma_g \approx t\sigma_g$ について，$R \vdash s\sigma_g \approx t\sigma_g$ が成立するときをいう．等式 $s \approx t$ が $R$ の帰納的定理であることを，$R \models_{ind} s \approx t$ と記す．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 $R$ の帰納的定理であるといい，$R \models_{ind} E$ と記す．

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 [?]．書き換え帰納法では，等式集合 $E$ と項書き換え規則集合 $H$ の対 $\langle E, H\rangle$ に関する導出をおこないながら証明を行う．用いる導出規則を以下に示す．

- *Expand*
$$\frac{\langle E \uplus \{s \stackrel{\cdot}{\approx} t\}, H\rangle}{\langle E \cup Expd_u(s,t), H \cup \{s \to t\}\rangle} u \in B(s), s > t$$
- *Simplify*
$$\frac{\langle E \uplus \{s \stackrel{\cdot}{\approx} t\}, H\rangle}{\langle E \cup \{s' \approx t\}, H\rangle} s \to_{R \cup H} s'$$
- *Delete*
$$\frac{\langle E \uplus \{s \approx s\}, H\rangle}{\langle E, H\rangle}$$

また，*Expand* 規則中で用いる $Expd_u$ を以下のように定義する．

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

なお，適宜，$(\mathcal{V}(s) \cup \mathcal{V}(t)) \cap \mathcal{V}(l) = \emptyset$ となるように，書き換え規則の変数は名前変えする．ここで，$\uplus$ は直和，$>$ は簡約順序を表す．$\langle E, H\rangle$ から導出規則を利用して $\langle E', H'\rangle$ が導かれることを，$\langle E, H\rangle \rightsquigarrow_{>,R} \langle E', H'\rangle$ と記す．等式集合 $E$ が $R$ の帰納的定理であることを証明するには，$R \subseteq >$ なる簡約順序を与え，$\langle E, \emptyset\rangle$ から導出を始め，導出規則を繰り返し適用する．最終的に，$\langle E, \emptyset\rangle \rightsquigarrow^*_{>,R} \langle \emptyset, H\rangle$ が導出されるとき証明成功となる．$E$ が空にならないまま無限に導出を繰り返すとき，手続きが発散するという．

**命題 3.1 (書き換え帰納法の正当性 [?])** $R$ を擬簡約な項書き換えシステム，$>$ を $R \subseteq >$ なる簡約順序とする．このとき，ある $H$ に対して，$\langle E, \emptyset\rangle \rightsquigarrow^*_{>,R} \langle \emptyset, H\rangle$ となるならば，$R \models_{ind} E$.

**定理 3.2 (CTRS における帰納的定理証明)** $R$ を弱左線形かつ強簡約不能な 3DCTRS とする．また，$U(R)$ は完備かつ擬簡約な左線形 TRS であり，$>$ を $U(R) \subseteq >$ なる簡約順序とする．このとき，ある $H$ に対して，$\langle E, \emptyset\rangle \rightsquigarrow^*_{>,U(R)} \langle \emptyset, H\rangle$ となるならば，$R \models_{ind} E$.

**(証明)** $R$ が 3DCTRS で，弱左線形性と強簡約不能性を持つこと，また，$U(R)$ は完備かつ左線形であることから，系 2.6 により，$U$ は健全かつ完全であり，$R$ は論理性をもつ．$U(R)$ および $U(R)$ における書き換え帰納法の導出に関する仮定から，命題 3.1 より，$U(R) \models_{ind} E$ が成立する．よって，$s \approx t \in E$ の任意の基底代入例 $s\sigma_g \approx t\sigma_g$ について，$U(R) \vdash s\sigma_g \approx t\sigma_g$ が成立する．特に，$\mathcal{F}$ 上の等式 $s\sigma_g \approx t\sigma_g$ について，$U(R) \vdash s\sigma_g \approx t\sigma_g$ が成立．よって，補題 2.7 より，$R \vdash s\sigma_g \approx t\sigma_g$ が成立する．したがって，$R \models_{ind} E$. □

## 4 ホーン節帰納的定理の自動証明

本節では，等式帰納的定理に対する書き換え帰納法をより一般的なホーン節帰納的定理を扱えるよう拡張する．条件付き等式 $s_1 \approx t_1, \ldots, s_k \approx t_k \Rightarrow l \approx r$ が TRS $R$ の**ホーン節帰納的定理**であるとは，その任意の基底代入例 $s_1\sigma_g \approx t_1\sigma_g, \ldots, s_k\sigma_g \approx t_k\sigma_g \Rightarrow l\sigma_g \approx r\sigma_g$ について，$s_1\sigma_g \leftrightarrow_R^* t_1\sigma_g, \ldots, s_k\sigma_g \leftrightarrow_R^* t_k\sigma_g$ が成立するならば，$l\sigma_g \leftrightarrow_R^* r\sigma_g$ が成立するときをいう．

以下では，等式集合 $\Gamma$ の合同閉包を $\approx_\Gamma$ と記し，条件付き書き換え規則 $c \Rightarrow l \to r \in H$ が存在して，$s = C[l\sigma]$，$s' = C[r\sigma]$，$\forall u \approx v \in c.\, u\sigma \approx_\Gamma v\sigma$ となるとき，$\Gamma \vdash s \to_H s'$ と記す．また，導出規則中で用いる $Expd_u$ を以下のように定義する：$Expd_u(\Gamma, s, t) =$

$$\begin{cases} \{\Gamma\sigma_i \Rightarrow C[r_i]\sigma_i \approx t\sigma_i \mid s = C[u], \sigma_i = mgu(l_i, u), l_i \to r_i \in R\} \\ \qquad\qquad (u \in \mathcal{B}(s) \text{ の場合}) \\ \{(\Gamma \setminus \{v \approx w\})\sigma_i \cup \{C[r_i]\sigma_i \approx w\sigma_i\} \Rightarrow s\sigma_i \approx t\sigma_i \mid v = C[u], \sigma_i = mgu(l_i, u), l_i \to r_i \in R\} \\ \qquad\qquad (u \in \mathcal{B}(v), v \approx w \in \Gamma \text{ の場合}) \end{cases}$$

**定義 4.1 (ホーン節帰納的定理のための書き換え帰納法)** ホーン節帰納的定理に対する書き換え帰納法の導出規則を以下のように与える．

- *Expand*
$$\frac{\langle E \uplus \{\Gamma \Rightarrow s \mathrel{\dot{\approx}} t\}, H\rangle}{\langle E \cup Expd_u(\Gamma, s, t), H \cup \{s \to t \Leftarrow \Gamma\}\rangle} u \in \mathcal{B}(s), s > t$$
- *Simplify*
$$\frac{\langle E \uplus \{\Gamma \Rightarrow s \mathrel{\dot{\approx}} t\}, H\rangle}{\langle E \cup \{\Gamma \Rightarrow s' \approx t\}, H\rangle} s \to_R s' \vee \Gamma \vdash s \to_H s'$$
- *Delete*
$$\frac{\langle E \uplus \{\Gamma \Rightarrow s \approx t\}, H\rangle}{\langle E, H\rangle} s \approx_\Gamma t$$
- *C-Expand*
$$\frac{\langle E \uplus \{\Gamma \Rightarrow s \approx t\}, H\rangle}{\langle E \cup Expd_u(\Gamma, s, t), H\rangle} u \in \mathcal{B}(v), v \approx w \in \Gamma$$
- *C-Simplify*
$$\frac{\langle E \uplus \{\Gamma, u \approx v \Rightarrow s \approx t\}, H\rangle}{\langle E \cup \{\Gamma, u' \approx v \Rightarrow s \approx t\}, H\rangle} u \to_R u'$$
- *C-Delete*
$$\frac{\langle E \uplus \{\Gamma, u \approx v \Rightarrow s \approx t\}, H\rangle}{\langle E, H\rangle} root(u), root(v) \in \mathcal{C}, root(u) \neq root(v)$$

**例 4.2 (ホーン節帰納的定理に対する書き換え帰納法の例)** 入力する CTRS $R$ と証明する等式集合 $E$ が以下のように与えられたときの，書き換え帰納法による導出の例を示す．

$$R = \left\{ \begin{array}{llll} ge(0, y) & \to & true & ge(s(x), 0) \to false \\ ge(s(x), s(y)) & \to & ge(x, y) & \end{array} \right\}$$

$$E = \left\{ \ \rho \ : \ ge(x,y) \approx true, ge(y,z) \approx true \Rightarrow ge(x,z) \approx true \ \right\}$$

$$\begin{array}{ll}
 & \left\langle \left\{ \rho \ : \ ge(x,y) \approx true, ge(y,z) \approx true \Rightarrow \underline{ge(x,z)} \approx true \right\}, \emptyset \right\rangle \\
\rightsquigarrow_{Expand} & \left\langle \left\{ \begin{array}{l} \underline{ge(0,y) \approx true, ge(y,z) \approx true \Rightarrow true \approx true} \\ ge(s(x),y) \approx true, ge(y,0) \approx true \Rightarrow false \approx true \\ ge(s(x),y) \approx true, ge(y,s(z)) \approx true \Rightarrow ge(x,z) \approx true \end{array} \right\}, \{\rho\} \right\rangle \\
\rightsquigarrow_{Delete} & \left\langle \left\{ \begin{array}{l} ge(s(x),\underline{y}) \approx true, \underline{ge(y,0)} \approx true \Rightarrow false \approx true \\ ge(s(x),y) \approx true, ge(y,s(z)) \approx true \Rightarrow ge(x,z) \approx true \end{array} \right\}, \{\rho\} \right\rangle \\
\rightsquigarrow_{C\text{-}Expand} & \left\langle \left\{ \begin{array}{l} \underline{ge(s(x),0)} \approx true, true \approx true \Rightarrow false \approx true \\ \underline{ge(s(x),s(y))} \approx true, false \approx true \Rightarrow false \approx true \\ ge(s(x),y) \approx true, ge(y,s(z)) \approx true \Rightarrow ge(x,z) \approx true \end{array} \right\}, \{\rho\} \right\rangle \\
\rightsquigarrow^*_{C\text{-}Simplify} & \left\langle \left\{ \begin{array}{l} \underline{false \approx true, true \approx true \Rightarrow false \approx true} \\ \underline{ge(x,y) \approx true, false \approx true \Rightarrow false \approx true} \\ ge(s(x),y) \approx true, ge(y,s(z)) \approx true \Rightarrow ge(x,z) \approx true \end{array} \right\}, \{\rho\} \right\rangle \\
\rightsquigarrow^*_{C\text{-}Delete} & \left\langle \left\{ \ ge(s(x),y) \approx true, \underline{ge(y,s(z))} \approx true \Rightarrow ge(x,z) \approx true \ \right\}, \{\rho\} \right\rangle \\
\rightsquigarrow_{C\text{-}Expand} & \left\langle \left\{ \begin{array}{l} \underline{ge(s(x),0)} \approx true, true \approx true \Rightarrow ge(x,z) \approx true \\ \underline{ge(s(x),s(y))} \approx true, ge(y,z) \approx true \Rightarrow ge(x,z) \approx true \end{array} \right\}, \{\rho\} \right\rangle \\
\rightsquigarrow_{C\text{-}Simplify} & \left\langle \left\{ \begin{array}{l} \underline{false \approx true, true \approx true \Rightarrow ge(x,z) \approx true} \\ ge(x,y) \approx true, ge(y,z) \approx true \Rightarrow ge(x,z) \approx true \end{array} \right\}, \{\rho\} \right\rangle \\
\rightsquigarrow_{C\text{-}Delete} & \left\langle \left\{ \ ge(x,y) \approx true, ge(y,z) \approx true \Rightarrow \underline{ge(x,z)} \approx true \ \right\}, \{\rho\} \right\rangle \\
\rightsquigarrow_{Simplify} & \left\langle \left\{ \ \underline{ge(x,y) \approx true, ge(y,z) \approx true \Rightarrow true \approx true} \ \right\}, \{\rho\} \right\rangle \\
\rightsquigarrow_{Delete} & \langle \emptyset, \{\rho\} \rangle
\end{array}$$

このように定義 4.1 の導出規則を適切な順番で用いることにより，書き換え帰納法の導出が成功する．また，導出規則を適用する箇所に下線を引き，用いた導出規則を $\rightsquigarrow$ の添字として記した．例中では，最初の導出規則として *Expand* を用いているが，例えば *C-Expand* を用いることもできる．

以下では，拡張した書き換え帰納法が，ホーン節帰納的定理の証明法として正しいことを証明する．

**定理 4.3 (書き換え帰納法の健全性)** $R$ を擬簡約かつ完備な項書き換えシステムとし，$>$ を $R \subseteq >$ なる単純化順序とする．$E$ を条件付き等式の集合とするとき，$\langle E, \emptyset \rangle \rightsquigarrow^*_{>,R} \langle \emptyset, H \rangle$ ならば，$R \models_{ind} E$ である．

**(証明)** 以下では，$R \not\models_{ind} E$ を仮定して矛盾を導く．$R \not\models_{ind} E$ と仮定すると，$R \not\models_{ind} \rho$ を満たすような $\rho \in E$ が存在する．特に，$E^\infty = \bigcup_i E_i$ とおくと，$R \not\models_{ind} \rho$ を満たすような $\rho \in E^\infty$ が存在する．よって，ある $\rho \ : \ u_1 \approx v_1, \ldots, u_n \approx v_n \Rightarrow s \approx t \in E$ とある基底代入 $\sigma_g$ で，各 $1 \leq i \leq n$ について $u_i\sigma_g \overset{*}{\leftrightarrow}_R v_i\sigma_g$ かつ $s\sigma_g \not\overset{*}{\leftrightarrow}_R t\sigma_g$ となるものが存在する．

条件付き等式 $\rho \ : \ u_1 \approx v_1, \ldots, u_n \approx v_n \Rightarrow s \approx t$ に対して，その重みを

$$w(\rho) = \langle \{s,t\}_m, \{\{u_1, v_1\}_m, \ldots, \{\{u_n, v_n\}_m\}_m \rangle$$

により定義する．また，$w(E) = \{w(p) \mid \rho \in E\}$ とする．ここで，$\{\ldots\}_m$ は多重集合を表わす．また，重み上の順序 $\gg_{lex}$ を，$>$ の多重集合拡張 $\gg$(の多重集合拡張) の辞書式拡張を用いて定める．

順序 $>$ の整礎性よりこの順序は整礎であるから，各 $1 \leq i \leq n$ について $u_i\sigma_g \overset{*}{\leftrightarrow}_R v_i\sigma_g$ が成立し，$s\sigma_g \overset{*}{\not\to}_R t\sigma_g$ となるような $\rho : u_1 \approx v_1, \ldots, u_n \approx v_n \Rightarrow s \approx t \in E^\infty$ および基底代入 $\sigma_g$ で，$w(\rho\sigma_g)$ が極小となるもの (極小反例) が存在する．また，このとき，$R$ が擬簡約で停止性をもつ TRS であることから，$\sigma_g$ は基底構成子代入となる．

いま，$\rho \in E_k$ とおく．このとき，$\rho'\sigma_g$ が極小反例となるような $\rho' \in E_{k+1}$ が存在すること (※) を示す．

$\langle E_k, H_k\rangle$ から $\langle E_{k+1}, H_{k+1}\rangle$ の導出で用いられる導出規則が，$\rho$ 以外の条件付き等式に適用されている場合には，$\rho' := \rho$ ととることで (※) は明らかに成立する．したがって，$\langle E_k, H_k\rangle$ から $\langle E_{k+1}, H_{k+1}\rangle$ の導出で，$\rho$ が変形されている場合を考えればよい．以下では，導出でどの導出規則が適用されているかによって場合分けを行う．また，$\Gamma = \{u_1 \approx v_1, \ldots, u_n \approx v_n\}$ とおく．

- (*Expand* 規則)
  このとき，$R$ の擬簡約性より，ある $u'_1 \approx v'_1, \ldots, u'_n \approx v'_n \Rightarrow s' \approx t' \in Expd(\Gamma, s, t)$ およびある $\sigma'_g$ が存在して，$s\sigma_g \to_R s'\sigma'_g$，$t\sigma_g = t'\sigma'_g$，かつ，任意の $1 \leq i \leq n$ について，$u'_i\sigma'_g = u_i\sigma_g \leftrightarrow^*_R v_i\sigma_g = v'_i\sigma'_g$．よって，$\rho' := u'_1 \approx v'_1, \ldots, u'_n \approx v'_n \Rightarrow s' \approx t'$ ととると，$\rho'\sigma'_g$ は反例となる．しかし，$w(\rho\sigma_g) \gg_{lex} w(\rho'\sigma'_g)$ より，これは $\rho\sigma_g$ の極小性に矛盾．よって，このような場合はあり得ない．
- (*Simplify* 規則) $\rho$ に導出規則を適用して得られた条件付き等式を $\rho'$ とすると，$\rho : \Gamma \Rightarrow s \approx t$, $\rho' : \Gamma \Rightarrow s' \approx t$ とおける．このとき，次の 2 通りに場合分けして示す．
  1. $s \to_R s'$ の場合．このとき，$\rho' \in E_{k+1}$ かつ $w(\rho\sigma_g) \gg_{lex} w(\rho'\sigma_g)$．よって，このような場合はあり得ない．
  2. $\Gamma \vdash s \to_H s'$ の場合．このとき，ある $\delta : l \to r \Leftarrow c \in H$ と代入 $\theta$ が存在して，$s = C[l\theta]$，$s' = C[r\theta]$，かつ，任意の $u \approx v \in c$ について $u\theta \approx_\Gamma v\theta$．また，$\rho\sigma_g$ が反例であることから，$\Gamma\sigma_g \subseteq \leftrightarrow^*_R$ かつ $s\sigma_g \overset{*}{\not\to}_R t\sigma_g$ が成立する．よって，任意の $u \approx v \in c$ について，$u\theta\sigma_g \leftrightarrow^*_R v\theta\sigma_g$．いま，基底構成子代入 $\theta_g$ を $\theta_g(x) = (\sigma_g \circ \theta)(x)\downarrow_R$ により定める．すると，$c\theta\sigma_g \subseteq \leftrightarrow^*_R$ より，任意の $u \approx v \in c$ について，$u\theta_g \leftrightarrow^*_R v\theta_g$．また，$s\sigma_g = C\sigma_g[l\theta\sigma_g] \trianglerighteq l\theta\sigma_g \to^*_R l\theta_g$ かつ $t\sigma_g = C\sigma_g[r\theta\sigma_g] \trianglerighteq r\theta\sigma_g \to^*_R r\theta_g$ となるので，$>$ が単純化順序であることから，$\{s\sigma_g, t\sigma_g\}_m \gg\!\!= \{l\theta_g, r\theta_g\}_m$ が成立する．さらに，$l \to r \Leftarrow c \in H$ であることから，$c \Rightarrow l \approx r \in E_\infty$ に対して *Expand* 規則が適用されている．よって，$l > r$，かつ，ある $\delta' : c' \Rightarrow l' \approx r' \in Expd(c, l, r)$ と $\theta'_g$ が存在して，$l\theta_g \to_R l'\theta'_g$，$r\theta_g = r'\theta'_g$，かつ，$c'\theta'_g = c\theta_g$．このとき，任意の $u \approx v \in c'$ について，$u\theta'_g = u\theta_g \leftrightarrow^*_R v\theta_g = v\theta'_g$．また，$\{l\theta_g, r\theta_g\}_m \gg \{l'\theta'_g, r'\theta'_g\}_m$ であるから，$w(\rho\sigma_g) \gg_{lex} w(\delta'\theta'_g)$．よって，$\rho\sigma_g$ の極小性より，$l'\theta'_g \leftrightarrow^*_R r'\theta'_g$ が成立する．したがって，$s\sigma_g = C\sigma_g[l\theta\sigma_g] \to^*_R C\sigma_g[l\theta_g] \to_R C\sigma_g[l'\theta'_g] \leftrightarrow^*_R C\sigma_g[r'\theta'_g] = C\sigma_g[r\theta_g] = s'\sigma_g$ となり，$s'\sigma_g \overset{*}{\not\to}_R t\sigma_g$ が得られるので，$\rho'\sigma_g$ が反例となる．一方，$>$ は簡約順序であるので $l > r$ より $s = C[l\theta] > C[r\theta] = s'$．よって，$w(\rho\sigma_g) \gg_{lex} w(\rho'\sigma_g)$ となるが，これは $\rho\sigma_g$ の極小性に矛盾．よって，このような場合はあり得ない．
- (*Delete* 規則) このとき，$\rho\sigma_g$ が反例ではなく，このような場合はあり得ない．
- (*C-Expand* 規則) このとき，$\rho' \in Expd(\Gamma, s, t)$ で，$w(\rho\sigma_g) \gg_{lex} w(\rho'\sigma'_g)$ かつ $\rho'\sigma'_g$ が反例となるものが存在する．よって，$\rho\sigma_g$ の極小性から，このような場合はあり得ない．
- (*C-Simplify* 規則) このとき，$w(\rho\sigma_g) \gg_{lex} w(\rho'\sigma_g)$ かつ $\rho'\sigma_g$ が反例となる．しかし，$\rho\sigma_g$ の極小性から，このような場合はあり得ない．
- (*C-Delete* 規則) このとき，$u\sigma_g \overset{*}{\leftrightarrow}_R v\sigma_g$．よって，$R$ が合流性をもつことから，$u\sigma_g \to^*_R \circ \leftarrow^*_R v\sigma_g$ となり，したがって，$root(u), root(v) \in \mathcal{C}$ より $root(u) = root(v)$．これは，$root(u) \neq$

$root(v)$ に矛盾するので，このような場合はあり得ない．

以上で，(※) が成立することが示された．一方，(※) は，導出が $\langle E, \emptyset\rangle \overset{*}{\rightsquigarrow} \langle \emptyset, H\rangle$ となったことに矛盾する．よって，反例は存在しない．つまり，$R \models_{ind} E$ が成立する． □

定理 3.2 と定理 4.3 を組み合わせると以下が得られる．

**系 4.4** $R$ をシグニチャ $\mathcal{F}$ 上の弱左線形かつ強簡約不能な 3DCTRS とする．また，$U(R)$ は完備かつ擬簡約な左線形 TRS であり，$>$ を $U(R) \subseteq >$ なる単純化順序とする．$E$ をシグニチャ $\mathcal{F}$ 上の条件付き等式の集合とするとき，ある $H$ に対して，$\langle E, \emptyset\rangle \rightsquigarrow^*_{>,U(R)} \langle \emptyset, H\rangle$ なるならば，$R \models_{ind} E$.

## 5 帰納的定理証明システムの実装と実験

アンラベリング変換およびホーン節帰納的定理の自動証明のための書き換え帰納法を実装した．実装には，関数型言語 SML/NJ[1] を用い，プログラムの行数は約 8500 行である．実装した自動証明システムは，CTRS $R$ および等式のホーン節集合 $E$ を入力し，$R \models_{ind} E$ かを判定する．なお，現在のところ，$R \models_{ind} E$ かの判定は完全自動ではなく，証明戦略および簡約順序に用いる $\mathcal{F} \cup \mathcal{U}$ 上の順序の入力が必要である．簡約順序には (半順序 $\leq$ に基づく) 辞書式経路順序 [**?**] を用いた．また，$R$ の弱左線形性や強簡約不能性，$U(R)$ の完備性や擬簡約性，左線形性は，ユーザが保証するものとして，判定していない．

また，*Delete* 規則と *Simplify* 規則には，等号付き第一階述語論理の定理証明器である E Theorem Prover[2] を用いた．まず，*Delete* 規則の条件である $s \approx_\Gamma t$ を判定するには，$\mathcal{V}(\Gamma, s, t)$ の変数をそれぞれ新たな定数に対応させて，$\Gamma, s, t$ の変数を定数化 (スコーレム化) したものを $\widehat{\Gamma}, \hat{s}, \hat{t}$ とおくとき，仮定 $\widehat{\Gamma}$ のもとで，$\hat{s} \approx \hat{t}$ が成立するかを判定すればよい．この判定に E Theorem Prover を用いた．次に，*Simplify* 規則については，$s \to_R s'$ または $\Gamma \vdash s \to_H s'$ なる $s'$ を $s$ から構成する必要がある．$s \to_R s'$ の方は通常の書き換えであるので問題ない．$\Gamma \vdash s \to_H s'$ については，$\Gamma, s$ の変数を定数化したものを $\widehat{\Gamma}, \hat{s}$ とおいた後，それぞれの $c \Rightarrow l \to r \in H$ について，(1) $\hat{s} = C[l\sigma]$ なる $C$，$\sigma$ を求め，(2) $\mathcal{V}(c\sigma) = \{x_1, \ldots, x_n\}$ として，仮定 $\widehat{\Gamma}$ のもとで， $c\sigma$ が成立するような $x_1, \ldots, x_n$ の具体化を求める必要がある．これは，$\exists x_1, ..., x_n.c\sigma$ を `question` モードで E Theorem Prover に入力することで得ることが出来る．最後に，得られた $x_1, \ldots, x_n$ の具体化 $\rho$ をもちいて，$\hat{s}' = C[(r\sigma)\rho]$ を構成し，$\hat{s}'$ のスコーレム定数を変数に戻すことで $s'$ が得られる．

次に，今回の実験に用いた証明戦略のヒューリスティックスについて説明する．導出規則の適用に際して 2 つのサイクルパターンを用いた．それらを以下に示す．

- サイクル 1

$$\textit{Expand} \Rightarrow \textit{C-Simplify} \Rightarrow \textit{C-Delete} \Rightarrow \textit{Simplify} \Rightarrow \textit{Delete}\,(\Rightarrow \textit{Expand} \text{ or } \textit{C-Expand})$$

- サイクル 2

$$\textit{C-Expand} \Rightarrow \textit{C-Simplify} \Rightarrow \textit{C-Delete} \Rightarrow \textit{Simplify} \Rightarrow \textit{Delete}\,(\Rightarrow \textit{Expand} \text{ or } \textit{C-Expand})$$

この 2 つのサイクルのうち，どちらをどの順番で選択して書き換え帰納法を行うかというヒューリスティックの指定を行った．具体的には，ステップ数 $n$ について，$n \mod k$ の値に応じて，サイクル 1 の方を選択するようにした．

表 1 に，実験結果のまとめを示す．全部で 7 例に対して実験をおこなった．conjectures は，証明する帰納的定理を指す．それぞれの例で，CTRS として，conjecture に用いられる定義記号に対す

[1] `http://www.smlnj.org/`
[2] `http://wwwlehre.dhbw-stuttgart.de/~sschulz/E/E.html`

**表 1.** 提案手法を用いた帰納的定理証明の実験結果

| No. | conjectures | 結果 | 戦略 | | SPIKE |
|---|---|---|---|---|---|
| | | | h1 | h2 | |
| 1 | $plus(x,y) \approx 0 \Rightarrow x \approx 0$ | ○ | ○ | ○ | ○ |
| 2 | $ge(x,y) \approx true, ge(y,z) \approx true \Rightarrow ge(x,z) \approx true$ | ○ | × | ○ | $\infty$ |
| 3 | $even(x) \approx true \Rightarrow odd(x) \approx false$ | ○ | ○ | × | ○ |
| 4 | $even(plus(x,y)) \approx false, even(x) \approx true \Rightarrow even(y) \approx false$ | × | - | - | ○ |
| 5 | $len(xs) \approx x, len(ys) \approx y \Rightarrow len(app(xs,ys)) \approx plus(x,y)$ | × | - | - | $\infty$ |
| 6 | $app(ys,zs) \approx xs \Rightarrow len(xs) \approx plus(len(ys), len(zs))$ | × | - | - | $\infty$ |
| 7 | $split(x,ys) \approx pair(us,vs) \Rightarrow len(ys) \approx plus(len(us), len(vs))$ | × | - | - | × |

る自然な定義を用いたが，その詳細は付録に記載する．“結果” は，実装システムによる帰納的定理証明の結果を，“SPIKE” は，帰納的定理証明システム SPIKE[3][**?**, **?**, **?**] による帰納的定理証明の結果であり，○は証明成功を，×は証明失敗を，$\infty$ は発散を示す．“戦略” は，実験に用いた証明戦略を表わす．

実験において用いた証明戦略のヒューリスティックスは以下の通りである：

h1: $n \mod 2 = 1$ のときにサイクル 1 を選択

h2: $n \mod 3 = 1$ のときにサイクル 1 を選択

今回の実験では，実際にどのように規則の適用が行われているかを確認し，失敗の原因を確認したあと，各サイクルの頻度を決めた．

実際，異なるヒューリスティクのもとでは証明は成功しなかった．例えば実験例 2 について，以下のような等式に導出規則を適用することを考える．

$$ge(s(x_1), y) \approx true, ge(y, s(y_1)) \approx true \Rightarrow ge(x_1, y_1) \approx true$$

この等式にまずは *Expand* 規則を適用すると以下の等式を得る．

$$ge(s(0), y) \approx true, ge(y, s(y_2)) \approx true \Rightarrow true \approx true$$
$$ge(s(s(x_2)), y) \approx true, ge(y, s(y_2)) \approx true \Rightarrow false \approx true$$

一方，*C-Expand* 規則を適用すると以下の等式を得る．

$$false \approx true, ge(0, s(y_1)) \approx true \Rightarrow ge(x_2, y_1) \approx true$$
$$ge(x_2, y_2) \approx true, ge(s(y_2), s(y_1)) \approx true \Rightarrow ge(x_2, y_1) \approx true$$

ここで，どちらの規則を適用したほうがいいのかという問題が発生するが，上記の場合はこの後の *C-Delete* 規則の適用を考え，*C-Expand* 規則を適用するべきである．なお *Expand* 規則を適用した際は，等式が増えていき，証明が失敗する．このように，サイクル 1 とサイクル 2 のどちらを何回目に利用するかで，証明が成功するかどうかが決まってしまう．

実験例 5–7 では，*Expand* 規則における順序条件 ($s > t$) が満たされないため，*Expand* 規則の適用ができない．他の規則だけでは，帰納的定理証明に成功しないため，本論文で提案した書き換え帰納法では扱うことが出来なかった．結果として，それぞれの例について適切な証明戦略をとることで，7 例中 3 例の帰納的定理証明に成功した．また，成功例における実行時間はすべて 4msec 以下であった．SPIKE システムでは，7 例中 3 例の帰納的定理証明に成功し，4 例で失敗した (うち，3 例は発散し，1 例は適切な順序をとることが出来なかった)．本手法では証明できず，SPIKE では発散の結果を得た実験例 5，6 については，本手法を理論面から拡張する必要があると考えられる．

[3] `https://github.com/sorinica/spike-prover/wiki`

また本手法では証明できず，SPIKE では証明できる実験例 4 についても，理論的に拡張する必要がある．提案手法を用いた帰納的定理証明システムの実現には，より多くの例について実験を行い，一般的に有効な証明戦略を考案することが必須であると考えられるが，適切な証明戦略を用いれば，提案手法がホーン節帰納的定理の自動証明に有効であることがわかった．なお，本実験では 7 例しか実験をおこなっていないが，これは 7 例の実験をおこなった時点で，いくつかの改善点を発見したためで，これ以上の実験には注力していない．

*Expand* 規則における順序条件による制約については，文献 [?] と同様な方法を用いて制約をなくすことが出来ると考えられるが，そのような拡張は，今後の課題の 1 つである．また，本論文で提案した書き換え帰納法では，帰納的定理でないことの証明 (反証) は行えない．反証のための導出規則についても，[?] と同様の手法を用いることで可能と考えられるが，それも今後の課題の 1 つである．また，項書き換えシステムにおける等式帰納的定理におけるさまざまな補題生成法や書き換え規則変換法，決定手続きの利用などの拡張も，強力な帰納的定理自動証明システムの実現のためには必要と考えられる．

## 6 おわりに

本論文では，条件付き項書き換えシステムにおけるホーン節帰納的定理の自動証明法を提案した．このため，項書き換えシステムにおける等式帰納的定理の自動証明手法である書き換え帰納法を，ホーン節帰納的定理を扱える書き換え帰納法へ拡張するとともに，その正当性を示した．また，条件付き項書き換えシステム $R$ におけるホーン節帰納的定理証明については，アンラベリング変換によって得られた $U(R)$ を用いて，ホーン節帰納的定理のための書き換え帰納法を実行することで実現するとともに，この手法が適用可能な条件付き項書き換えシステムの条件を明らかにした．

また，提案手法にもとづく，条件付き項書き換えシステムにおけるホーン節帰納的定理の自動証明システムを実装するとともに，実装システムを用いて実験を行った．そして，適切な証明戦略を用いれば，提案手法がホーン節帰納的定理の証明に有効であることを明らかにした．より一般的に有効な証明戦略を考案することや，ホーン節帰納的定理の自動証明に適した，より強力な書き換え帰納法を提案すること，また，本手法の完全性を明らかにすることは今後の課題である．

## 謝辞

本論文を丁寧に査読していただいた査読者に感謝いたします．なお，本研究は一部日本学術振興会科学研究費 15K00003 の補助を受けて行われた．

## A 実験の詳細

以下に実験例として用いた書き換え規則，条件付き等式を記載する．

- 実験例 1
  - 書き換え規則

$$R_1 = \left\{ \begin{array}{ll} plus(0, y) & \to y \\ plus(s(x), y) & \to s(plus(x, y)) \end{array} \right\}$$

  - 条件付き等式

$$plus(x, y) \approx 0 \Rightarrow x \approx 0$$

- 実験例 2

- 書き換え規則

$$R_2 = \left\{ \begin{array}{ll} ge(0,y) & \to true \\ ge(s(x),0) & \to false \\ ge(s(x),s(y)) & \to ge(x,y) \end{array} \right\}$$

- 条件付き等式

$$ge(x,y) \approx true, ge(y,z) \approx true \Rightarrow ge(x,z) \approx true$$

- 実験例 3
  - 書き換え規則

$$R_3 = \left\{ \begin{array}{ll} even(0) & \to true \\ odd(0) & \to false \\ even(s(x)) & \to odd(x) \\ odd(s(x)) & \to even(x) \end{array} \right\}$$

  - 条件付き等式

$$even(x) \approx true \Rightarrow odd(x) \approx false$$

- 実験例 4
  - 書き換え規則

$$R_4 = \left\{ \begin{array}{ll} even(0) & \to true \\ even(s(x)) & \to true \Leftarrow even(x) \approx false \\ even(s(x)) & \to false \Leftarrow even(x) \approx true \\ plus(x,0) & \to x \\ plus(x,s(y)) & \to s(plus(x,y)) \end{array} \right\}$$

  - 条件付き等式

$$even(plus(x,y)) \approx false, even(x) \approx true \Rightarrow even(y) \approx false$$

- 実験例 5
  - 書き換え規則

$$R_5 = \left\{ \begin{array}{ll} len([]) & \to 0 \\ len(:(x,xs)) & \to s(y) \Leftarrow len(xs) \approx y \\ plus(0,y) & \to y \\ plus(s(x),y) & \to s(plus(x,y)) \\ app([\,],xs) & \to xs \\ app(:(x,xs),ys) & \to :(x,app(xs,ys)) \end{array} \right\}$$

  - 条件付き等式

$$len(xs) \approx x, len(ys) \approx y \Rightarrow len(app(xs,ys)) \approx plus(x,y)$$

- 実験例 6
  - 書き換え規則

$$R_6 = R_5$$

  - 条件付き等式

$$app(ys,zs) \approx xs \Rightarrow len(xs) \approx plus(len(ys), len(zs))$$

- 実験例 7

– 書き換え規則

$$R_7 = \left\{ \begin{array}{ll} gt(0, x) & \to true \\ gt(s(x), 0) & \to false \\ gt(s(x), s(y)) & \to gt(x, y) \\ split(x, [\,]) & \to pair([\,], [\,]) \\ split(x, : (y, ys)) & \to pair(xs, : (y, zs)) \Leftarrow split(x, ys) \approx pair(xs, zs), gt(x, y) \approx true \\ split(x, : (y, ys)) & \to pair(: (y, xs), zs) \Leftarrow split(x, ys) \approx pair(xs, zs), gt(x, y) \approx false \\ len([\,]) & \to 0 \\ len(: (x, xs)) & \to s(y) \Leftarrow len(xs) \approx y \\ plus(0, y) & \to y \\ plus(s(x), y) & \to s(plus(x, y)) \end{array} \right\}$$

– 条件付き等式

$$split(x, ys) \approx pair(us, vs) \Rightarrow len(ys) \approx plus(len(us), len(vs))$$