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

信学技報
TECHNICAL REPORT OF IEICE.

制約付き項書換え系の定理自動証明における等式の方向付けのための簡約化順序

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

†, †† 名古屋大学大学院情報科学研究科
〒 463–8603 名古屋市千種区不老町
E-mail: †{nishida,sakai,kusakari,sakabe}@is.nagoya-u.ac.jp, ††sakata@trs.cm.is.nagoya-u.ac.jp

あらまし　定理自動証明では手続きが停止せずに暴走してしまうことがある．その暴走を避けるために，制約付き項書換えに対する定理自動証明の手続きが潜在帰納法と書換え帰納法それぞれに対して提案された．一方，書換えが停止性を持つにもかかわらず，計算が進むにつれて項に含まれる書換え可能な項の大きさが増大していく場合がある．このような書換え系での定理自動証明において，書換え系の停止性を保存するように等式を方向付けして追加することは，経路順序を用いる手法では困難である．本稿では，引数切り落とし法を導入しながら，制約付き書換え系の停止性を保存する等式の方向付けに適した簡約化順序の条件を示す．制約付き TRS の潜在帰納法と書換え帰納法における定理証明での等式の方向付けにおいて，その条件を満たす簡約化順序が有効であることを示す．

キーワード　停止性，簡約化対，引数切り落とし法，潜在帰納法，書換え帰納法

A Reduction Order for Orienting Equations in Theorem Proving of Constrained TRSs

Naoki NISHIDA†, Tsubasa SAKATA††, Masahiko SAKAI†,

Keiichirou KUSAKARI†, and Toshiki SAKABE†

†, †† Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya 463-8603 Japan
E-mail: †{nishida,sakai,kusakari,sakabe}@is.nagoya-u.ac.jp, ††sakata@trs.cm.is.nagoya-u.ac.jp

Abstract　In automated theorem proving, we often face cases where the procedure keeps running. To avoid such cases, the theorem proving methods based on implicit and rewriting inductions have been proposed for constrained rewriting systems. On the other hand, in terminating term rewriting, there are some cases where computation is reducing while the size of reducible terms is increasing. In (automated) theorem proving for rewriting systems with such cases, it is very difficult (or near-impossible) to orient equations by path-based reduction orders in order to add the equations into the rewriting systems, preserving termination of the rewriting systems. In this paper, introducing the argument filtering technique, we show a condition of reduction orders that can be used in orienting equations with preserving termination of constrained systems. We also show that reduction orders satisfying the condition are effective for orientation phases in theorem proving methods based on implicit induction or rewriting induction for constrained term rewriting systems.

Key words　termination, reduction pair, argument filtering, implicit induction, rewriting induction

1. はじめに

手続き型プログラムと関数型プログラムでは，それぞれにプログラム検証手法の研究がされている．手続き型プログラムに対しては，モデル検査 [5], [10], [15] やホーア論理に基づく検証手法 [6], [8], [10] が代表的である．しかし，ループ不変式の発見は一般的に不可能であることや，事前条件・事後条件の付与などのヒューリスティックな作業が必要であることから，検証の完全な自動化は困難である．一方，関数型プログラムに対しては，帰納的定理の自動証明手法である潜在帰納法 [9], [13] や

書換え帰納法 [3], [16] などが項書換え系（TRS）の分野で広く研究されている．帰納的定理とは任意の基底項上で成立する等式である [4]．関数の等価性は帰納的定理として定式化できるので，等価性検証に帰納的定理の自動証明法を利用できる．しかし，整数の比較演算を表す項を含む等式が帰納的定理であることを示す場合，ほとんどの場合で潜在帰納法と書換え帰納法に基づいた証明手続きは暴走してしまい，証明に成功しない．

そこで，そのような暴走を軽減するために，プレスブルガー文のような制約を書換え規則が持つことを許した制約付き TRS の枠組で定理証明を行う手法が提案された [17], [18]．文献 [18] では，制約付き TRS に対する潜在帰納法に基づく証明手続きが提案され，手続き型関数から等価な制約付き TRS への変換を与えることで，手続き型プログラムの検証に潜在帰納法を利用する枠組が示された．文献 [17] では，文献 [18] の手法の課題であった制約の分解を解決することをめざし，書換え帰納法に基づいた証明手続きを制約付き TRS に拡張した．これらの証明手続きは比較演算によって暴走してしまう手続きを収束させることには有効である．一方で，手続きを実装し証明を自動化するにはいくつかの課題が残されている．

文献 [17], [18] の証明手続きは，制約付き TRS R と R の帰納的定理であることを証明したい等式の集合 E，さらに R の停止性を保証する簡約化順序を入力にとり，等式を順序付けして得られる書換え規則の集合 H を加えた対 (E, R, H) もしくは $(E, R \cup H)$ にある操作を繰り返し施していき，E が空集合になれば証明に成功して手続きを終了する．繰り返し行われる操作は主に次の作業である．E から $R \cup H$ の停止性を保存する，すなわち，簡約化順序に従って方向付けできる等式を H に追加し，その追加から派生する計算の分岐に相当する等式を E に加える．そして，E の等式の両辺を $R \cup H$ で書き換え，両辺が等しくなった等式は E から除去する．これらの操作を自動化するためには，等式を簡約化順序に従って自動的に方向付けしなければならない．このための順序として，辞書式経路順序などのような，関数記号の順序からあらかじめ静的に定められる経路順序がよく用いられる．このような順序は TRS だけでなく制約付き TRS にも適用可能ではあるが，以下で説明するように，while ループに対応するような計算を持つ場合に対しては有効ではない．

while ループを変換して得られる制約付き TRS では，例え停止性を持っていても，計算が進むにつれて項に含まれる書換え可能な項の大きさは増大していく場合が多い．例として，文献 [18] で示された次の制約付き TRS を考える．

$$
R_{\mathtt{sum}} = \left\{ \begin{array}{rcll}
\mathsf{sum1}(n) & \to & \mathrm{u}_2(n, \mathsf{s}(\mathsf{0}), \mathsf{0}) & (1) \\
\mathrm{u}_2(n, i, z) & \to & \mathrm{u}_2(n, \mathsf{s}(i), i + z) \Leftarrow i \mathsf{<=} n & (2) \\
\mathrm{u}_2(n, i, z) & \to & z \Leftarrow \neg(i \mathsf{<=} n) & (3) \\
\mathsf{sum}(\mathsf{0}) & \to & \mathsf{0} & (4) \\
\mathsf{sum}(\mathsf{s}(n)) & \to & \mathsf{sum}(n) + \mathsf{s}(n) & (5) \\
\mathsf{0} + y & \to & y & (6) \\
\mathsf{s}(x) + y & \to & \mathsf{s}(x + y) & (7)
\end{array} \right.
$$

$\mathsf{sum1}(n)$ と $\mathsf{sum}(n)$ は自然数 0 から n の総和を求める関数であり，$\mathtt{sum1}$ は手続き型プログラムから変換して得られた関数である．u_2 は while ループに対応した動作をする関数記号である．項 $\mathrm{u}_2(t_n, t_i, t_z)$ はループの中での記憶の状態を表しているとみなせる．(2) の規則が適用されて 1 回のループに相当する書換えが起こると，項 $\mathrm{u}_2(t_n, t_i, t_z)$ は $\mathrm{u}_2(t_n, \mathsf{s}(t_i), \mathsf{s}(t_i) + t_z)$ に書き換えられる．このとき，項の構造は大きくなるが，ループの終了条件を満たす状態 $\mathrm{u}_2(t_n, \mathsf{s}(t_n), t'_z)$ へは近づいている．このとき，$R_{\mathtt{sum}}$ が停止性を持つにもかかわらず，$R_{\mathtt{sum}}$ の停止性を保証する経路順序は存在しない．さらに，手続きの中では，その停止性を保存するように等式を方向付けする作業を繰り返し行わなければならない．なお，手続きでは最初に一度，与えられた制約付き TRS の停止性を示す必要がある．しかし，ほとんどの場合でその制約付き TRS はプログラムに相当するので，書換え規則を人間が見て停止性を判定することや帰納法などで証明できる場合も多い．近年，プレスブルガー文を制約に持つ制約付き TRS の停止性証明の手法も提案されている [7]．その一方で，プログラムの動作に対応しない等式が手続きの中で制約付き TRS に追加されていく毎に，その停止性を確認することは非常に困難な作業である．

TRS でも同様の状況が起こるが，TRS の場合には既存の停止性自動証明ツールを利用することでこの問題を解決できる場合もある．近年，TRS の停止性自動証明ツールは非常に性能が向上している．しかし，while ループに相当する計算を持つ TRS の停止性証明，および制約付き TRS の停止性証明に対してはまだ十分に開発されていない．また，停止性証明ツールを用いる手法では，簡約化順序を用いる場合に比べて効率が大きく低下する．

本稿では，引数切り落とし法を導入しながら，制約付き書換え系の停止性を保存する等式の方向付けのための簡約化順序の条件を示す．具体的には，順序の対である簡約化対 $(\gtrsim, \succ)$ と引数切り落とし関数 π から簡約化対 $(\gtrsim_\pi, \succ_\pi)$ を構成し，停止性を持つ制約付き TRS の書換え規則に順序付けを施す．その際には，擬順序 $\gtrsim_\pi$ を用い，等式の方向付けには狭義の半順序 $\succ_\pi$ を用い，左辺が右辺のどの部分項よりも大きくなるようにする．元の制約付き TRS の停止性と簡約化対 $(\gtrsim_\pi, \succ_\pi)$ から新たに定義する簡約化対と依存対法 [1] の原理から，この順序付けに従って得られた制約付き TRS が停止性を持つことを示す．また，制約付き TRS の潜在帰納法と書換え帰納法における定理証明での等式の方向付けにおいて，その条件を満たす簡約化順序が有効であることを示す．

本稿は次のように構成される．2. 節では，項書換えおよび制約付き TRS に関する記法および定義を示す．3. 節では，制約付き TRS の潜在帰納法と書換え帰納法に用いる簡約化順序を示す．4. 節では，今後の課題を述べる．

2. 準　　備

本稿では，項書換えの一般的な記法に従う [2], [14].

2.1 抽象書換え系

抽象書換え系 S は，対象となる集合 A と A 上の**簡約化関係**と呼ばれる二項関係 $\to$ の組 $(A, \to)$ である．$a \xrightarrow{*} c \xleftarrow{*} b$ とな

るような $c \in A$ が存在するとき，$b \overset{*}{\leftarrow} a \overset{*}{\rightarrow} c$ ならば $b \downarrow c$ のとき，S は**合流性**を持つという．任意の $a \in A$ に対して a から始まる $\rightarrow$ の無限系列が存在しないとき，S は**停止性**（または**強正規性**）を持つという．すべての $a \in A$ が正規形を持つとき，

2.2 項と代入

関数記号の集合 $\mathcal{F}$，**変数**の可算無限集合 $\mathcal{V}$，から生成されるすべての**項**の集合を $\mathcal{T}(\mathcal{F},\mathcal{V})$ とする．関数記号 f の引数個数を $\mathtt{arity}(f)$ で表す．項 t に現れるすべての変数の集合を $\mathcal{V}ar(t)$ で表す．項 s と t が同一であるときは $s \equiv t$ と記述する．項 t への 1 引数関数記号 f の n 回の適用は $f^n(t)$ と略記する．項 t における**位置**の集合を $\mathcal{O}(t)$ とする．位置 $p,q \in \mathcal{O}(t)$ に対して，$pp' = q$ を満たす p' が存在するとき，$p \leqq q$ と書く．特に $p' \neq \varepsilon$ とき，$p < q$ と記す．$\mathtt{root}(s)$ は項 s の先頭（位置 ε）の記号を表す．**ホール** $\Box$ を特別な関数記号とし，引数は持たないとする．**文脈**とは，$\Box$ を一つだけ含む項である．ホール自身も文脈であり，このような文脈を空の文脈という．文脈 $C[\,]$ において位置 p に出現するホール $\Box$ を項 t で置き換えることによって得られる項を $C[t]_p$ と記す．なお，p を省略してもよい．$\mathcal{F},\mathcal{V}$ 上のすべての文脈の集合を $\mathcal{T}_\Box(\mathcal{F},\mathcal{V})$ とする．項 t,u に対して $t \equiv C[u]_p$ となるような文脈 $C[\,]$ が存在するとき，u を t の**部分項**と呼び，特に $\varepsilon < p$ のとき，u を**真部分項**という．このとき，それぞれ $t \trianglerighteq u$，$t \triangleright u$ と記す．

代入 σ の定義域と値域をそれぞれ $\mathcal{D}om(\sigma)$（$= \{x \mid x \neq \sigma(x)\}$）と $\mathcal{R}an(\sigma)$（$= \{\sigma(x) \mid x \in \mathcal{D}om(\sigma)\}$）で表す．$\mathcal{D}om(\sigma) = \{x_1,\ldots,x_n\}$ であり，かつ $\sigma(x_i) \equiv t_i$ のとき，σ を $\{x_1 \mapsto t_1,\cdots,x_n \mapsto t_n\}$ と記す．項 t に対して，$\sigma(t)$ を t の**インスタンス**と呼び，$t\sigma$ と略記する．代入 σ,σ' について，$\mathcal{D}om(\sigma) = \mathcal{D}om(\sigma')$ かつすべての $x \in \mathcal{D}om(\sigma)$ において $\sigma(x) \equiv \sigma'(x)$ のとき，$\sigma = \sigma'$ と記述する．σ の定義域を $X \subseteqq \mathcal{V}$ に制限した代入 $\sigma|_X$ を $\{x \mapsto x\sigma \mid x \in \mathcal{D}om(\sigma) \cap X\}$ と定義する．

2.3 簡約化順序と簡約化対

A を集合とする．A 上の**擬順序** $\succsim$ は反射性と推移性を持つ二項関係である．A 上の**狭義の半順序** $\succ$ は推移性と非反射性を持つ二項関係である．無限列 $a_1 \rightarrow a_2 \rightarrow \cdots$ が存在しないとき，二項関係 $\rightarrow$ は**整礎**であるという．擬順序 $\succsim$ で定義される同値関係 $\succsim \cap \precsim$ を $\sim$ で表す．擬順序 $\succsim$ で定義される狭義の半順序 $\succsim \setminus \precsim$ を $\gtrsim$ で表す．

$\rightarrow$ を項上の二項関係とする．任意の項 s と t について，$s \rightarrow t$ ならば任意の文脈 $C[\,]$ について $C[s] \rightarrow C[t]$ であるとき，$\rightarrow$ は**文脈に閉じている**という．任意の項 s と t について，$s \rightarrow t$ ならば任意の代入 θ について $s\theta \rightarrow t\theta$ であるとき，$\rightarrow$ は**代入に閉じている**という．$\triangleright \subseteqq \rightarrow$ のとき，$\rightarrow$ は**部分項性**を持つという．項上の擬順序 $\succsim$ が文脈と代入に閉じているとき，$\succsim$ を**書換え順序**（rewrite order）と呼ぶ．項上の狭義の半順序 $\succ$ が整礎であり，文脈と代入に閉じているとき，$\succ$ を**簡約化順序**（reduction order）と呼ぶ．

$\mathcal{F}$ を関数記号の集合，$\succsim$ を $\mathcal{F}$ 上の擬順序とする．$\succsim$ で定義される**辞書式経路順序** $\succsim_{\mathsf{lpo}}$ は以下のように定義される．

- $s \succsim_{\mathsf{lpo}} x$，ただし，$x \in \mathcal{V}ar(s)$.
- 以下のいずれかが成立するとき，$f(s_1,\ldots,s_m) \succsim_{\mathsf{lpo}} g(t_1,\ldots,t_n)$.
  - $f \sim g$，かつ，ある i が存在して $s_j \sim_{\mathsf{lpo}} t_j$（$1 \leqq j < i$）かつ $s_i \succsim_{\mathsf{lpo}} t_i$，かつ，各 j について $f(s_1,\ldots,s_m) \gtrsim_{\mathsf{lpo}} t_j$.
  - $f \gtrsim g$ かつ，各 j について $f(s_1,\ldots,s_m) \gtrsim_{\mathsf{lpo}} t_j$.
  - ある i が存在して，$s_i \succsim_{\mathsf{lpo}} g(t_1,\ldots,t_n)$.

書換え順序 $\succsim$ と狭義の半順序 $\succ$ が以下を満たすとき，これらの対 $(\succsim,\succ)$ を**簡約化対**と呼ぶ．

- $\succ$ は整礎であり，代入に閉じている．
- $(\succsim \circ \succ) \subseteqq \succ$，または $(\succ \circ \succsim) \subseteqq \succ$.

2.4 引数切り落とし

本稿では，文献 [1], [11], [12] の引数切り落としの定義を制限した切り落とし関数を用いる．

$\mathcal{F}$ を関数記号の集合とする．$\mathcal{F}$ 上の**引数切り落とし関数** π は各関数記号 f に対して次のどちらかで定義される写像である．

- $\pi(f) = \{i_1,\ldots,i_k\}$，ただし，$i_j \in \{1,\ldots,\mathtt{arity}(f)\}$.
- $\pi(f) = i$，ただし，$1 \leqq i \leqq \mathtt{arity}(f)$.

f に対して $\pi(f)$ が明示されていないときは，$\pi(f) = \{1,\ldots,\mathtt{arity}(f)\}$ とする．π の項 t への適用は次のように定義される．

- $\pi(x) = x$，ただし，$x \in \mathcal{V}$.
- $\pi(f(t_1,\ldots,t_n)) = f(\pi(t_{i_1}),\ldots,\pi(t_{i_k}))$，ただし，$\pi(f) = \{i_1,\ldots,i_k\}$ かつ各 j について $i_j < i_{j+1}$.
- $\pi(f(t_1,\ldots,t_n)) = \pi(t_i)$，ただし，$\pi(f) = i$.

2.5 制約付き項書換え系

$\mathcal{P}$ は関数記号の集合 $\mathcal{F}_\mathcal{P}$，述語記号の集合 $\mathcal{H}_\mathcal{P}$，変数集合 $\mathcal{V}$ によって定められる一階述語論理式の集合であり，解釈 $\mathcal{I}_\mathcal{P}$ が付随しているものとする．すなわち，$\mathcal{P}$ は，アトム $p(t_1,\ldots,t_n)$（ここに，$p \in \mathcal{H}_\mathcal{P}$ は n 引数述語記号，$t_i \in \mathcal{T}(\mathcal{F}_\mathcal{P},\mathcal{V})$），$\mathtt{true}$，$\mathtt{false}$，$\wedge$，$\vee$，$\neg$，$\forall$，$\exists$ から構成される論理式の集合であり，付随する $\mathcal{I}_\mathcal{P}$ により自由変数を持たない論理式（**閉じた論理式**という）に真偽値が割り当てられているとする．以降では，$\mathcal{P}$ の論理式を**制約**と呼ぶ．制約 c の自由変数の集合を $\mathrm{fv}(c)$ で表す．閉じた制約 c が $\mathcal{P}$ の下で真であることを単に c **は真である**という．本稿では，$\mathcal{I}_\mathcal{P}$ の下では，閉じた制約の真偽判定問題が決定可能であるとする．よって，$\mathcal{P}$ の任意の論理式の $\mathcal{I}_\mathcal{P}$ の下での真偽値と充足可能性は決定可能である．限量子を含まない論理式 c への項の代入 θ の適用は，項の代入の自然な拡張とする．すなわち，$\theta(p(t_1,\ldots,t_n)) = p(\theta(t_1),\ldots,\theta(t_n))$（ただし，$p \in \mathcal{H}_\mathcal{P} \cup \{\mathtt{true},\mathtt{false}\}$），$\theta(\neg c) = \neg\theta(c)$，$\theta(c_1 \mathrm{\ op\ } c_2) = \theta(c_1) \mathrm{\ op\ } \theta(c_2)$（ただし，op は $\vee$ または $\wedge$）である．

$\mathcal{F}$ を関数記号の集合とし，$\mathcal{F} \supseteqq \mathcal{F}_\mathcal{P}$ であるとする．σ を $\mathcal{R}an(\sigma) \subseteqq \mathcal{T}(\mathcal{F},\mathcal{V})$ を満たす代入とする．$\mathcal{R}an(\sigma|_{\mathrm{fv}(c)}) \subseteqq \mathcal{T}(\mathcal{F}_\mathcal{P},\mathcal{V})$ であるときのみ σ を c へ適用することを許し，c へ σ を適用（入力）して得られる論理式を $c\sigma$ と表記する（$c\sigma \in \mathcal{P}$ である）．

$(\mathcal{F},\mathcal{P})$ 上の**制約付き書換え規則** (l,r,c) は，$l \notin \mathcal{V}$，$l,r \in \mathcal{T}(\mathcal{F},\mathcal{V})$，$\mathcal{V}ar(l) \supseteqq \mathcal{V}ar(r)$ を満たす**左辺**項 l，**右辺**項 r と，$\mathcal{V}ar(l) \supseteqq \mathrm{fv}(c)$ と $c \in \mathcal{P}$ を満たす**制約部** c の組であり，$l \to r \Leftarrow c$ と記す．本稿では c には限量子 $\forall, \exists$ が含まれないこととする．c が `true` のときは制約部を省略して $l \to r$ と書くこともある．R の**被定義記号**の集合と**構成子**の集合をそれぞれ $\mathcal{D}_R$，$\mathcal{C}_R$ とする：$\mathcal{D}_R = \{\mathtt{root}(l) \mid l \to r \Leftarrow c \in R\}$，$\mathcal{C}_R = \mathcal{F} \setminus \mathcal{D}_R$．本稿では $\mathcal{C}_R \subseteqq \mathcal{F}_\mathcal{P}$ を仮定する．

R を $(\mathcal{F},\mathcal{P})$ 上の制約付き書換え規則の有限集合とする．R で定められる書換え関係 $\to_R$ を，$\to_R = \{(C[l\sigma]_p, C[r\sigma]_p) \mid l \to r \Leftarrow c \in R,\ C[\,] \in \mathcal{T}_\Box(\mathcal{F},\mathcal{V}),\ c\sigma \text{ が真}\}$ と定義する(注1)．p を明記する場合には $\to_R^p$ と記述する．$(\mathcal{F},\mathcal{P})$ 上の**制約付き項書換え系**（constrained TRS）は項の集合 $\mathcal{T}(\mathcal{F},\mathcal{V})$ と書換え関係 $\to_R$ で定められる抽象書換え系 $(\mathcal{T}(\mathcal{F},\mathcal{V}), \to_R)$ であり，書換え規則の集合 R で表す．制約部が `true` である規則のみからなる制約付き項書換え系は**項書換え系**（TRS）である．

c を充足可能な論理式としたとき，任意の代入 θ に対して $c\theta$ が真ならば $d\theta$ も真（すなわち，$\vec{x}$ を c,d のすべての自由変数の並びとすると，“$\forall\vec{x}.(\neg c \vee d)$” が真(注2)）であることを $c \models d$ と書く．**充足可能な制約** c **の下での書換え関係** $\xrightarrow[c]{}_R$ を $\{(C[l\sigma],C[r\sigma]) \mid l \to r \Leftarrow d \in R,\ C[\,] \in \mathcal{T}_\Box(\mathcal{F},\mathcal{V}),\ c \models d\sigma\}$ と定義する(注3)．

命題 2.1（[18]） R を制約付き TRS とする．このとき，以下の 2 つは等価である．

- 任意の充足可能な制約 c について，$\xrightarrow[c]{}_R$ は停止性を持つ．
- $\to_R$ が停止性を持つ． □

ゆえに，R が停止性を持つとき，$\xrightarrow[c]{}_R$ も停止性を持つ．

3. 等式の方向付けに適した簡約化順序

本節では，簡約化対と引数切り落とし法を用いて，停止性を持つ制約付き TRS の書換え規則に対する順序付けとその制約付き TRS に追加される書換え規則に対する順序付けの手法を示す．さらに，その順序付けに従う制約付き TRS が停止性を持つことを示し，その制約付き TRS の書換え関係が潜在帰納法と書換え帰納法に必要な簡約化順序の役割を果たすことを示す．

本節では，停止性を持つ既存の制約付き TRS として R を用い，等式集合から方向付けされて R に追加される書換え規則の集合を H を用いて表現する．文献 [17], [18] の手続きでは，H の被定義記号は R の被定義記号であるように等式を方向付ける．すなわち，$\mathcal{D}_{R\cup H} = \mathcal{D}_R$ である．

3.1 制約付き TRS の停止性を保存する簡約化順序

まずは，簡約化対 $(\gtrsim,\succ)$ と引数切り落とし関数 π から定められる順序を定義する．

定義 3.1 $(\gtrsim,\succ)$ を簡約化対，π を引数切り落とし関数とする．このとき，関係 $\gtrsim_\pi$ と $\succ_\pi$ を次のように定義する．

- $\pi(s) \gtrsim \pi(t)$ のとき，$s \gtrsim_\pi t$.
- 以下のいずれかのとき，$s \succ_\pi t$.
  - $\pi(s) \rhd \pi(t)$.
  - $\pi(s) \succ C[\pi(t)]$ を満たす文脈 $C[\,]$ が存在する． □

文献 [11], [12] では簡約化順序から $\gtrsim_\pi$ と $\succ_\pi$ を定義し，$\gtrsim_\pi$ と $\succ_\pi$ は推移性を持たない．一方，本稿では簡約化対から $\gtrsim_\pi$ と $\succ_\pi$ を定義し，$\gtrsim_\pi$ と $\succ_\pi$ が推移性を持つための条件を示すことで，$(\gtrsim_\pi,\succ_\pi)$ を簡約化対として用いる．

次に，$(\gtrsim_\pi,\succ_\pi)$ が簡約化順序であるための条件を示す．

補題 3.2 以下のすべてが成り立つ．

- $\gtrsim_\pi$ は書換え順序である．
- $\rhd \subseteqq \succ$ かつ $\succ$ が文脈に閉じるならば，$\succ_\pi$ は推移性を持つ．
- $\succ_\pi$ は代入に閉じる．
- $\succ$ が部分項性を持つまたは $\succ$ が文脈に閉じるならば，$\succ_\pi$ は整礎である．
- $(\succ \circ \gtrsim) \subseteqq \succ$ ならば，$(\succ_\pi \circ \gtrsim_\pi) \subseteqq \succ_\pi$.
- $(\gtrsim \circ \succ) \subseteqq \succ$ ならば，$(\gtrsim_\pi \circ \succ_\pi) \subseteqq \succ_\pi$.

［証明］ 簡約化対と $\gtrsim_\pi$ と $\succ_\pi$ の定義より明らか． □

整礎な関係は非反射性を持つので，次の系が得られる．

系 3.3 π を引数切り落とし関数，$(\gtrsim,\succ)$ を簡約化対とする．$\succ$ が部分項性を持ちかつ $\succ$ が文脈に閉じるならば，$(\gtrsim_\pi,\succ_\pi)$ は簡約化対である． □

命題 3.4 $\gtrsim$ を書換え順序とする．このとき，$\gtrapprox$ は簡約化順序である．さらに，$\gtrsim$ が部分項性を持つならば，$\gtrapprox$ は部分項性を持つ． □

系 3.5 π を引数切り落とし関数，$\gtrsim$ を書換え順序とする．このとき，$\gtrsim$ が部分項性を持つならば，$(\gtrsim_\pi,\gtrapprox_\pi)$ は簡約化対である． □

次に，書換え規則の順序付けに用いる順序を定義する．

定義 3.6 R を $(\mathcal{F},\mathcal{P})$ 上の制約付き TRS，π を引数切り落とし関数，$(\gtrsim,\succ)$ を簡約化対とする．このとき，関係 $\gtrsim^{\mathcal{D}_R}$ と $\succ^{\mathcal{D}_R}$ を次のように定義する．

- 以下のすべてが成り立つとき，$l \gtrsim^{\mathcal{D}_R} r$.
  - $\mathcal{V}ar(l) \supseteqq \mathcal{V}ar(r)$.
  - r の任意の部分項 r'（$\trianglelefteq r$）について，$\mathtt{root}(r') \in \mathcal{D}_R$ ならば $l \gtrsim r'$.
- 以下のすべてが成り立つとき，$s \succ^{\mathcal{D}_R} t$.
  - $\mathcal{V}ar(s) \supseteqq \mathcal{V}ar(t)$.
  - $s \gtrsim t$.
  - t の任意の部分項 t'（$\trianglelefteq t$）について，$\mathtt{root}(t') \in \mathcal{D}_R$ ならば $s \succ t'$. □

簡約化対 $(\gtrsim_\pi,\succ_\pi)$ から構成される $\gtrsim_\pi^{\mathcal{D}_R}$ と $\succ_\pi^{\mathcal{D}_R}$ が R の停止性から $R \cup H$ の停止性を導く．

定理 3.7 $R \cup H$ を $(\mathcal{F},\mathcal{P})$ 上の制約付き TRS，$\gtrsim$ を部分項性を持つ書換え順序，π を引数切り落とし関数とする．$R \cup H$ が

(注1)：基底項上の述語を扱うので，$c\sigma$ が真と解釈されるときは $\mathrm{fv}(c\sigma) = \emptyset$ である．よって，$\mathcal{R}an(\sigma|_{\mathrm{fv}(c)}) \subseteqq \mathcal{T}(\mathcal{F}_\mathcal{P})$.

(注2)：“$\neg c \vee d$” は c が満たされるときには必ず d も満たされることを意味する．c が充足不能のときには，R が停止性を持つにもかかわらず制約付き書換えが停止しないことがある．

(注3)：$\to_R$ の定義と異なり，$\mathcal{R}an(\sigma|_{\mathrm{fv}(d)}) \subseteqq \mathcal{T}(\mathcal{F}_\mathcal{P})$ は満たされなくてよい．

以下のすべてを満たすとき，$R \cup H$ は停止性を持つ．

- R は停止性を持つ．
- R の任意の規則 $l \to r \Leftarrow c$ について，$l\ (\gtrsim_\pi^{\mathcal{D}_R} \cup \succ_\pi^{\mathcal{D}_R})\ r$．
- H の任意の規則 $s \to t \Leftarrow d$ について，$s \succ_\pi^{\mathcal{D}_R} t$．　□

定理 3.7 の証明は 3.2 小節で述べる．

例 3.8　1. 節の R_{sum} を考える．文献 [18] では以下の等式が右向きに方向付けされ，R_{sum} に追加されている．

$$H_{\mathsf{sum}} = \begin{cases} \mathsf{u}_2(\mathsf{s}(n), i, z) \approx \mathsf{u}_2(n, i, z) + \mathsf{s}(n) \Leftarrow i \mathsf{<=} n + \mathsf{1} & (8) \\ \mathsf{sum}(n) \approx \mathsf{u}_2(n, \mathsf{s}^2(\mathsf{0}), \mathsf{s}(\mathsf{0})) \Leftarrow \mathsf{1} \mathsf{<=} n & (9) \\ \mathsf{sum}(n) \approx \mathsf{0} \Leftarrow \neg(\mathsf{1} \mathsf{<=} n) & (10) \end{cases}$$

以下のような関数記号の擬順序 $\gtrsim$ と引数切り落とし関数 π から定義される簡約化対 $((\gtrsim_{\mathsf{lpo}})_\pi, (\succ_{\mathsf{lpo}})_\pi)$ を考える．

$$\mathsf{sum1} \gtrapprox \mathsf{sum} \gtrapprox \mathsf{u}_2 \gtrapprox + \gtrapprox \mathsf{s} \gtrapprox \mathsf{0}$$

$$\pi(\mathsf{u}_2) = \{1, 3\} \quad \pi(+) = 2$$

このとき，以下の関係が成り立つことから，R_{sum} の規則と H_{sum} の等式が定理 3.7 の意味で右向きに方向付けられる（それぞれの左辺と右辺に $(\gtrsim_{\mathsf{lpo}})_\pi^{\mathcal{D}_R}$，$(\succ_{\mathsf{lpo}})_\pi^{\mathcal{D}_R}$ の関係がある）．なお，斜線は π により切り落とされた項や文脈であることを表す．

(1)　$\mathsf{sum1}(n) \gtrsim_{\mathsf{lpo}} \mathsf{u}_2(n, \cancel{\mathsf{s}(\mathsf{0})}, \mathsf{0})$．
(2)　$\mathsf{u}_2(n, \cancel{l}, z) \gtrsim_{\mathsf{lpo}} \mathsf{u}_2(n, \cancel{\mathsf{s}(i)}, \cancel{l} \not\succ z) \gtrsim_{\mathsf{lpo}} \cancel{l} \not\succ z$．
(5)　$\mathsf{sum}(\mathsf{s}(n)) \gtrsim_{\mathsf{lpo}} \cancel{\mathsf{sum}(n)} \not\succ \mathsf{s}(n)$，$\mathsf{sum}(\mathsf{s}(n)) \gtrsim_{\mathsf{lpo}} \mathsf{sum}(n)$．
(7)　$\cancel{\mathsf{s}(x)} \not\succ y \gtrsim_{\mathsf{lpo}} \cancel{x} \not\succ y$．
(8)　$\mathsf{u}_2(\mathsf{s}(n), \cancel{l}, z)\ (\gtrsim_{\mathsf{lpo}} \cap \succ_{\mathsf{lpo}})\ \cancel{\mathsf{u}_2(n, i, z)} \not\succ \mathsf{s}(n)$，
$\mathsf{u}_2(\mathsf{s}(n), \cancel{l}, z) \succ_{\mathsf{lpo}} \mathsf{u}_2(n, \cancel{l}, z)$．
(9)　$\mathsf{sum}(n)\ (\gtrsim_{\mathsf{lpo}} \cap \succ_{\mathsf{lpo}})\ \mathsf{u}_2(n, \cancel{\mathsf{s}^2(\mathsf{0})}, \mathsf{s}(\mathsf{0}))$．　□

3.2　定理 3.7 の証明

本小節では，定理 3.7 の証明を与える．

$\mathcal{F}$ を関数記号の集合とする．このとき，$\mathcal{F}$ の関数記号にマーク $\sharp$ を付けた関数記号 $f^\sharp$ の集合を $\mathcal{F}^\sharp$ とする：$\mathcal{F}^\sharp = \{f^\sharp \mid f \in \mathcal{F}\}$．さらに，$\mathcal{T}(\mathcal{F}, \mathcal{V})$ の項 s の先頭の記号のみに $\sharp$ を付けて得られる項を $s^\sharp$ で表す：$f(t_1, \ldots, t_n)^\sharp = f^\sharp(t_1, \ldots, t_n)$．よって，$s^\sharp$ を用いた場合は s は変数ではないこととする．なお，$\sharp$ を付けずに書かれる項 s や t は $\mathcal{T}(\mathcal{F}, \mathcal{V})$ の項を表すこととする．

制約付き TRS$R \cup H$ の位置 ε での書換え関係から制約無し書換え規則を構成することで，制約付き TRS の停止性を制約無し TRS の停止性に帰着させる．制約無し書換え規則の集合 $(R \cup H)_u$ を次のように定義する．

$$(R \cup H)_u = \{\, (s, t) \mid s \to^{\varepsilon}_{R \cup H} t \,\}$$

$(R \cup H)_u$ は一般には制約無し書換え規則の無限集合である．

命題 3.9　$R \cup H$ を制約付き TRS とする．このとき，$\to_{R \cup H} = \to_{(R \cup H)_u}$．

［証明］ 書換え関係の定義と $(R \cup H)_u$ の定義より明らか．　□

よって，次の命題が成り立つ．

命題 3.10　制約付き TRS $R \cup H$ が停止性を持つとき，かつそのときに限り，$(R \cup H)_u$ は停止性を持つ．　□

このことから，$R \cup H$ の停止性を示すには，無限集合に拡張された TRS $(R \cup H)_u$ の停止性を示せばよい．

$(R \cup H)_u$ の停止性を証明するために依存対法 [1] を用いる．

定義 3.11（依存対 [1]）　$l \to C[t]$ を TRS S の書換え規則，$\mathtt{root}(t) \in \mathcal{D}_S$ とする．このとき，項の対 $(l^\sharp, t^\sharp)$ を $l \to C[t]$ の**依存対**と呼ぶ．S のすべての依存対の集合を $\mathcal{DP}(S)$ とする：$\mathcal{DP}(S) = \bigcup_{l \to r \in S} \{(l^\sharp, t^\sharp) \mid r \equiv C[t], \mathtt{root}(t) \in \mathcal{D}_S\}$．　□

定理 3.12（依存対法 [1]）　S を TRS，$(\gtrsim, \succ)$ を簡約化対とする．以下のすべてが成り立つとき，S は停止性を持つ．

- すべての書換え規則 $l \to r \in S$ について，$l \gtrsim r$．
- すべての危険対 $(s^\sharp, t^\sharp) \in \mathcal{DP}(S)$ について，$s^\sharp \succ t^\sharp$．　□

以降では，$(R \cup H)_u$ の停止性を依存対法で証明するための簡約化対を示す．

定義 3.13　R を $(\mathcal{F}, \mathcal{P})$ 上の制約付き TRS，$(\gtrsim, \succ)$ を簡約化対とする．このとき，関係 $\gtrsim^R$ と $\succ^R$ を以下のように定義する．

- 以下のすべてが成り立つとき，$s \gtrsim^R t$．
  - $s\ (\gtrsim \cup \to_R)^*\ t$．
  - $\mathtt{root}(t) \in \mathcal{D}_R$ ならば，$\mathtt{root}(s) \in \mathcal{D}_R$ かつ $s \gtrsim t$．
  - t の任意の部分項 t'（$\trianglelefteq t$）について，$\mathtt{root}(t') \in \mathcal{D}_R$ ならば，$\mathtt{root}(s') \in \mathcal{D}_R$ かつ $s' \gtrsim t'$ を満たす s の部分項 s'（$\trianglelefteq s$）が存在する．
- $\mathtt{root}(s) \in \mathcal{D}_R$ かつ $\mathtt{root}(t) \in \mathcal{D}_R$ かつ以下のいずれかのとき，$s \succ^R t$．
  - $s \succ t$．
  - $s \gtrsim t$ かつ $s\ (\xrightarrow{+}_R \circ \trianglerighteq)\ t$．
- $s \succ^R t$ のとき，$s^\sharp \succ^R t^\sharp$．　□

補題 3.14　R を $(\mathcal{F}, \mathcal{P})$ 上の TRS，$(\gtrsim, \succ)$ を簡約化対とする．このとき，以下のすべてが成り立つ．

- $\gtrsim^R$ は擬順序であり，代入と文脈に閉じる．
- $\succ^R$ は推移性を持ち，$\succ^R$ は代入に閉じる．
- R が停止性を持つならば，$\succ^R$ は整礎である．
- $(\gtrsim \circ \succ) \subseteqq \succ$ ならば，$(\gtrsim^R \circ \succ^R) \subseteqq \succ^R$．

［証明］ 簡約化対，$\gtrsim^R$，$\succ^R$ の定義より明らか．　□

整礎な関係は非反射性を持つので，次の系が得られる．

系 3.15　R を $(\mathcal{F}, \emptyset)$ 上の TRS，$(\gtrsim, \succ)$ を $(\gtrsim \circ \succ) \subseteqq \succ$ を満たす簡約化対とする．R が停止性を持つならば，$(\gtrsim^R, \succ^R)$ は簡約化対である．　□

系 3.16　$R \cup H$ を $(\mathcal{F}, \mathcal{P})$ 上の制約付き TRS，$\gtrsim$ を部分項性を持つ書換え順序，π を引数切り落とし関数とする．R が停止性を持つならば，$(\gtrsim_\pi^R, \succ_\pi^R)$ は簡約化対である．　□

定理 3.17　$R \cup H$ を $(\mathcal{F}, \mathcal{P})$ 上の制約付き TRS，$\gtrsim$ を部分項性を持つ書換え順序，π を引数切り落とし関数とする．さらに，$R \cup H$ が以下を満たすとする．

- R は停止性を持つ．
- R の任意の規則 $l \to r \Leftarrow c$ について，$l\ (\gtrsim_\pi^{\mathcal{D}_R} \cup \succ_\pi^{\mathcal{D}_R})\ r$．
- H の任意の規則 $s \to t \Leftarrow d$ について，$s \succ_\pi^{\mathcal{D}_R} t$．

このとき，以下のすべてが成り立つ．

- $(R \cup H)_u$ の任意の規則 $l \to r$ について，$l \gtrsim_\pi^R r$.
- $\mathcal{DP}((R \cup H)_u)$ の任意の危険対 $(s^\sharp, t^\sharp)$ について，$s^\sharp \succ_\pi^R t^\sharp$.

すなわち，$(R \cup H)_u$ は停止性を持つ． □

以上より，定理 3.7 は証明された．

3.3 潜在帰納法・書換え帰納法に用いる簡約化順序

潜在帰納法では元の制約付き TRS の規則および追加された規則の右辺が書換えられることもある．そのような場合にも $(\gtrsim_\pi^{\mathcal{D}_R}, \succ_\pi^{\mathcal{D}_R})$ による方向付けで，手続きを通して有効である簡約化順序を R と π と $(\gtrsim, \succ)$ のみから構成できることを示す

H を R と $\succ_\pi^{\mathcal{D}_R}$ で順序付けられるすべての書換え規則の集合とする：$H = \{s \to t \Leftarrow d \mid \mathtt{root}(s) \in \mathcal{D}_R, s \succ_\pi^{\mathcal{D}_R} t\}$．次に，$R$ と H の書換え規則の右辺が書換えられた集合 R^+ と H^+ をそれぞれ以下のように定義する．

$$R^+ = \{\, l \to r' \Leftarrow c \mid l \to r \Leftarrow c \in R, r \xrightarrow[c]{*}_{R \cup H} r' \,\}$$

$$H^+ = \{\, s \to t' \Leftarrow d \mid s \to t \Leftarrow d \in H, t \xrightarrow[d]{*}_{R \cup H} t' \,\}$$

命題 3.18 $R \cup H$ を $(\mathcal{F}, \mathcal{P})$ 上の制約付き TRS，$R' \subseteqq R^+$，$H' \subseteqq H^+$ とする．このとき，$\to_{R' \cup H'} \subseteqq \to_{R^+ \cup H^+} \subseteqq \xrightarrow{+}_{R \cup H}$. ゆえに，$\xrightarrow{+}_{R' \cup H'} \subseteqq \xrightarrow{+}_{R^+ \cup H^+} = \xrightarrow{+}_{R \cup H}$. □

定理 3.7 と命題 3.18 より，$\xrightarrow{+}_{R \cup H}$ は潜在帰納法 [18] と書換え帰納法 [17] に有効な簡約化順序であることが導かれる．

定理 3.19 $R \cup H$ を $(\mathcal{F}, \mathcal{P})$ 上の制約付き TRS，$\gtrsim$ を部分項性を持つ書換え規則，π を引数切り落とし関数とする．さらに，$R \cup H$ が以下を満たすとする．

- R は停止性を持つ．
- R の任意の規則 $l \to r \Leftarrow c$ について，$l \ (\gtrsim_\pi^{\mathcal{D}_R} \cup \succ_\pi^{\mathcal{D}_R})\ r$.
- H の任意の規則 $s \to t \Leftarrow d$ について，$s \succ_\pi^{\mathcal{D}_R} t$.

このとき，$\xrightarrow{+}_{R \cup H}$ は簡約化順序であり，任意の $R' \subseteqq R^+$ と $H' \subseteqq H^+$ に対して，$\to_{R' \cup H'} \subseteqq \xrightarrow{+}_{R \cup H}$ を満たす． □

簡約化順序 $\xrightarrow{+}_{R \cup H}$ は停止性を持つ制約付き TRS R と部分項性を持つ書換え順序 $\gtrsim$ と引数切り落とし関数 π から定められる．よって，文献 [17], [18] の手続きでは，あらかじめ R の停止性を示し，適切な $\gtrsim$ と π を与えることで，等式の方向付けを自動化できる．

4. おわりに

本稿では，与えられた制約付き TRS が停止性を持つ場合に，制約付き TRS の潜在帰納法と書換え帰納法における等式の方向付けを自動化する簡約化順序の条件を示した．本手法は，制約付き TRS に対してだけでなく制約無し TRS に対しても有効である．制約付き TRS での定理自動証明は，人間が手で行なうには非常に煩雑な作業で，正確に行なうことは困難であった．しかし，等式の方向付けを自動化できたことで，文献 [17], [18] の証明手続きの自動化が実現できる．今後は，多くの例に対して，制約付き TRS の定理自動証明の実験を行なえる．文献 [17], [18] の証明手続きを自動化したシステムを実装し，多くの実験を行なうことが今後の課題である．

謝辞 本研究は一部，文部科学省科学研究費#18500011, #20300010，#20500008，および栢森情報科学振興財団の補助を受けている．

文　献

[1] Arts, T. and Giesl, J.: Termination of Term Rewriting Using Dependency Pairs, *Theoretical Computer Science*, Vol. 236, No. 1-2, pp. 133–178 (2000).
[2] Baader, F. and Nipkow, T.: *Term Rewriting and All That*, Cambridge University Press (1998).
[3] Bouhoula, A.: Automated Theorem Proving by Test Set Induction, *Journal of Symbolic Computation*, Vol. 23, No. 1, pp. 47–77 (1997).
[4] Boyer, R. S. and Moore, J. S.: *A Computational Logic*, Academic Press (1979).
[5] Clarke, E. M. and Emerson, E. A.: Design and Synthesis of Synchronization Skeletons Using Branching Time Temporal Logic, *Proceedings of Logic and Programs Workshop*, Lecture Notes in Computer Science, Vol. 131, Springer, pp. 52–71 (1981).
[6] Dijkstra, E. W.: A Discipline of Programming, Prentice-Hall (1976).
[7] Falke, S. and Kapur, D.: Dependency Pairs for Rewriting with Built-in Numbers and Semantic Data Structures, *Proceedings of the 19th International Conference on Rewriting Techniques and Applications* (to appear) (2008).
[8] Hoare, C. A. R.: An Axiomatic Basis for Computer Programming, *Communications of the ACM*, Vol. 12, No. 10, pp. 576–580 (1969).
[9] Huet, G. P. and Hullot, J.-M.: Proofs by Induction in Equational Theories with Constructors, *Journal of Computer and System Sciences*, Vol. 25, No. 2, pp. 239–266 (1982).
[10] Huth, M. and Ryan, M.: *Logic in Computer Science: Modelling and Reasoning about Systems*, Cambridge University Press (2000).
[11] Kusakari, K., Nakamura, M. and Toyama, Y.: Argument Filtering Transformation, *Proceedings of the International Conference on Principles and Practice of Declarative Programming*, Lecture Notes in Computer Science, Vol. 1702, Springer, pp. 47–61 (1999).
[12] Kusakari, K., Nakamura, M. and Toyama, Y.: limination Transformations for Associative-Commutative Rewriting Systems, *Journal of Automated Reasoning*, Vol. 37, No. 3,, pp. 205–229 (2006).
[13] Musser, D. R.: On Proving Inductive Properties of Abstract Data Types, *Conference Record of the Seventh Annual ACM Symposium on Principles of Programming Languages*, pp. 154–162 (1980).
[14] Ohlebusch, E.: *Advanced Topics in Term Rewriting*, Springer-Verlag (2002).
[15] Queille, J.-P. and Sifakis, J.: Specification and Verification of Concurrent Systems in CESAR, *Proceedings of the 5th International Symposium on Programming*, Lecture Notes in Computer Science, Vol. 137, Springer, pp. 337–351 (1982).
[16] Reddy, U. S.: Term Rewriting Induction, *Proceedings of the 10th International Conference on Automated Deduction*, Lecture Notes in Computer Science, Vol. 449, Springer, pp. 162–177 (1990).
[17] 坂田 翼，西田 直樹，坂部 俊樹，酒井 正彦，草刈 圭一朗：プレスブルガー文付き項書換え系における書換え帰納法について，信学技報 SS2008-1，Vol. 108，No. 64，pp. 1–6 （2008）．
[18] 古市 祐樹，西田 直樹，酒井 正彦，草刈 圭一朗，坂部 俊樹：制約付き項書換え系の潜在帰納法を利用した手続き型プログラム検証の試み，情報処理学会論文誌 プログラミング（掲載予定）（2008）．