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.

関数プログラムの停止性証明のための辞書式経路順序

星野 由美† 草刈 圭一朗†† 酒井 正彦†† 坂部 俊樹†† 西田 直樹††

† 名古屋大学 工学部 電気電子・情報工学科 〒 464-8603 名古屋市 千種区 不老町
†† 名古屋大学大学院 情報科学研究科 〒 464-8603 名古屋市 千種区 不老町
E-mail: †yhoshino@sakabe.i.is.nagoya-u.ac.jp, ††{kusakari,sakai,sakabe,nishida}@is.nagoya-u.ac.jp

あらまし 項書き換え系は関数プログラムの計算モデルであり，停止性はその重要な性質の一つである．停止性証明法の一つに簡約化順序によって順序付ける方法がある．代表的な簡約化順序としては辞書式経路順序が知られており，その整礎性は単純化順序の概念を用いて保証される．本稿では，高階関数を扱うことができるように拡張した辞書式経路順序を 3 種類提案する．そして，様々な項の集合上でそれらが持つ性質を明らかにする．
キーワード 辞書式経路順序，簡約化順序，単純化順序，停止性，高階関数

Lexicographic Path Ordering for Proving Termination of Functional Programs

Yumi HOSHINO†, Keiichirou KUSAKARI††, Masahiko SAKAI††,
Toshiki SAKABE††, and Naoki NISHIDA††

† Department of Information Engineering, School of Engineering, Nagoya University Furo-cho, Chikusa-ku, Nagoya, 464-8603
†† Graduate School of Information Science, Nagoya University Furo-cho, Chikusa-ku, Nagoya, 464-8603
E-mail: †yhoshino@sakabe.i.is.nagoya-u.ac.jp, ††{kusakari,sakai,sakabe,nishida}@is.nagoya-u.ac.jp

Abstract Term rewriting system is a computational model of functional programs, and termination is one of the important properties. Reduction order is an important tool for proving termination. As typical reduction orders, we know lexicographic path orderings whose well-foundedness are ensured by the notion of simplification order. In this paper, we propose three extended lexicographic path orderings in order to deal with higher-order functions. Additionally, we analyze their properties under several constraints.
Key words Lexicographic Path Ordering, Reduction Order, Simplification Order, Termination, Higher-Order Function

1. はじめに

項書き換え系（TRS）は，関数プログラムの計算モデルであり，定理自動証明や代数的仕様記述，プログラム検証などに利用されている [4]．一方，高階関数は関数型言語における最大の特徴の一つであるが，TRS では高階関数を直接扱うことができないため，型無し項書き換え系（UTRS）や，それに基づく単純型項書き換え系（STRS）といった高階項書き換え系も提案されている [2]．

TRS の重要な性質の一つに停止性がある．停止性を持つ TRS では，どのような項に対しても計算が終了することが保証される．TRS の停止性は決定不能であるが，応用上欠かせない性質であるので，これまでに様々な停止性証明法が提案されている．

TRS の停止性証明法として，簡約化順序または準簡約化順序によって書き換え規則の両辺を順序付ける方法がある．すべての書き換え規則を順序付けることができれば，その TRS の停止性を示すことができる．簡約化順序の代表的なものに，経路順序と呼ばれる関数記号上の順序を項上に拡張したものがあり，特に再帰経路順序や辞書式経路順序はよく知られている．また，経路順序の整礎性は単純化順序の概念を用いて保証される [1]．

上記 2 つの経路順序のうち，再帰経路順序は UTRS 上に拡張されているので高階関数を扱うことができるが，数学的に重要な結合律 $f(f(x,y),z) = f(x,f(y,z))$ を順序付けることができない．これに対して辞書式経路順序は，結合律の順序付けが可能なことに加えて，実装が比較的容易であるという利点を持つ．

本稿では，高階関数を扱うことができるように拡張した辞書

式経路順序を 3 種類提案する．また，それぞれの順序が持つ性質を解析し，引数個数を定めた項の集合上では簡約化順序，引数個数の上限を定めた項の集合と単純型項の集合上では準簡約化順序であることを明らかにする．

2. 諸 定 義

本稿で必要となる基本定義を与える．なお，型無し項書き換え系（UTRS）と単純型項書き換え系（STRS）は文献 [2] で提案された．本稿では文献 [3] の定式化に従う．

関数記号の集合 Σ と変数記号の集合 $\mathcal{V}$ から生成される**項**の全体からなる集合 $\mathcal{T}(\Sigma,\mathcal{V})$ は，$a \in \Sigma \cup \mathcal{V}$ かつ $t_1,...,t_n \in \mathcal{T}(\Sigma,\mathcal{V})$ ならば $a[t_1,...,t_n] \in \mathcal{T}(\Sigma,\mathcal{V})$ のように帰納的に定義される．項 $a[\,]$ は単に a と記す．$s \equiv a[s_1,...,s_n]$ とするとき，$s[t_1,...,t_m]$ と書いて項 $a[s_1,...,s_n,t_1,...,t_m]$ を表す．また，2 つの項 s と t が構文的に等しいことを $s \equiv t$ で表す．項 t に出現する全ての変数の集合を $Var(t)$ で表す．本稿では，a を $\Sigma \cup \mathcal{V}$ の要素として用いる．

項 t における**位置**の集合 $Pos(t)$ を正整数の列（空列を ε とする）を用いて以下のように定義する．$t \in \mathcal{V}$ のときは $Pos(t)=\{\varepsilon\}$，$t \equiv a[t_1,...,t_n]$ のときは $Pos(t)=\{\varepsilon\}\cup\{i\cdot p' \mid 1 \leqq i \leqq n, p' \in Pos(t_i)\}$．また，位置 p における記号を $(t)_p$ で記す．また，親を持たない位置を**根**，子を持たない位置を**葉**と呼ぶ．特に，項 t の根の位置の記号を $root(t)$ で記す．

代入は，変数から項への関数である．代入 θ の項上への拡張を $\hat{\theta}$ として，$\hat{\theta}(a[t_1,...,t_n])$ を $a \in \Sigma$ のときは $a[\hat{\theta}(t_1),...,\hat{\theta}(t_n)]$，$a \in \mathcal{V}$ のときは $\theta(a)[\hat{\theta}(t_1),...,\hat{\theta}(t_n)]$ で定義する．以下では，簡便のため θ と $\hat{\theta}$ を同一視する．また，$\theta(t)$ を $t\theta$ で略記する．

文脈とは，穴と呼ばれる特別な関数記号 $\square$ を 1 つだけ持つ項である．文脈 $C[\,]$ 中の $\square$ を項 t で置き換えることによって得られる項を $C[t]$ で表す．また，根の位置に $\square$ を持つ文脈を**根文脈**，葉の位置に $\square$ を持つ文脈を**葉文脈**と呼び，それぞれ $C_R[\,]$, $C_L[\,]$ で記す．

空でない**基本型**（basic type）の集合を $\mathcal{B}$ で表す．基本型 $\mathcal{B}$ から生成される**単純型**（simple type）の集合 $\mathcal{S}$ は，型構成子 $\to, \times$ を用いて $\mathcal{S} ::= \mathcal{B} \mid (\mathcal{S} \to \mathcal{S}) \mid (\mathcal{S} \times \cdots \times \mathcal{S})$ のように定義される．$\alpha \to \beta$ の形の単純型を**関数型**，それ以外の単純型を**非関数型**と呼び，それぞれの集合を $\mathcal{S}^{fun}$，$\mathcal{S}^{nfun}$ で表す．**組化構成子**と呼ぶ特別な関数記号 $tp \in \Sigma$ の存在を仮定する．項 $tp[t_1,...,t_n]$ を $(t_1,...,t_n)$ と記すこともある．**型関数** τ は，$\Sigma \cup \mathcal{V}$ から $\mathcal{S}$ への関数である．ただし，tp に対しては $\tau(tp)$ は未定義であるとする．項 $t \equiv a[t_1,...,t_n] \in \mathcal{T}(\Sigma,\mathcal{V})$ が単純型 α を持つとは，各 $t_i (i=1,...,n)$ が単純型 α_i を持ち，$a = tp$ のときは $\alpha = \alpha_1 \times \cdots \times \alpha_n (n \geqq 2)$，そうでないときは $\tau(a) = \alpha_1 \to \cdots \to \alpha_n \to \alpha$ となることである．単純型を持つ項を**単純型項**（simply-typed term）と呼び，その全体から成る集合を $\mathcal{T}_\tau(\Sigma,\mathcal{V})$ と記す．

書き換え規則とは，$root(l) \in \Sigma, Var(l) \supseteq Var(r)$ の条件を満たす項 l, r の対 (l,r) であり，$l \to r$ と記す．**型無し項書き換え系**（untyped term rewriting system；UTRS）とは，書き換え規則の集合である．UTRS R が**単純型項書き換え系**（simply-typed term rewriting system；STRS）であるとは，すべての書き換え規則 $l \to r \in R$ の両辺が同じ単純型を持つことである．

ある規則 $l \to r \in R$ と代入 θ と文脈 $C[\,]$ が存在して $s \equiv C[l\theta]$ かつ $t \equiv C[r\theta]$ となることを，$s \to_R t$ または単に $s \to t$ で表す．ここで，$\to_R$ を**書き換え関係**と呼ぶ．2 項関係 Υ が**整礎**であるとは $t_1 \Upsilon t_2 \Upsilon t_3 \Upsilon \cdots$ となる無限列が存在しないことである．$\to_R$ が整礎であるとき，R は**停止性**を持つと言う．

反射的かつ推移的かつ反対称的な 2 項関係 $\geqq$ を半順序と呼び，推移的かつ非反射的な 2 項関係 $>$ を狭義の半順序，反射的かつ推移的な 2 項関係 $\gtrsim$ を擬順序と呼ぶ．$>$ を集合 A 上の狭義の半順序とする．このとき，$>$ を A のリスト上へ辞書式拡張したものを**辞書式順序**と呼び，$>^{lex}$ で記す．なお，擬順序を基にした $\gtrsim^{lex}$ も同様に定義する．また，$>$ が T 上で**文脈に閉じている**とは，$s, t, C[s], C[t] \in T$ のとき $s > t \Rightarrow C[s] > C[t]$ であることを言う．葉文脈と根文脈に対しても同様に定義する．さらに，$>$ が T 上で**代入に閉じている**とは，$s, t, s\theta, t\theta \in T$ のとき $s > t \Rightarrow s\theta > t\theta$ であることを，**部分項性を持つ**とは，$t, C_L[t] \in T$ のとき $C_L[t] \geqq t$ であることを，**削除性を持つ**とは，$a[...,t,...], a[...,...] \in T$ のとき $a[...,t,...] > a[...,...]$ であることをそれぞれ言う．

定義 2.1 (（準）簡約化順序)　T を $T \subseteqq \mathcal{T}(\Sigma,\mathcal{V})$ となる任意の項の集合とする．このとき，ある順序 $>$ が T 上で**準簡約化順序**であるとは，$>$ が以下の性質を満たすことである．

- $>$ は T 上で整礎である．
- $>$ は T 上で狭義の半順序である．
- $>$ は T 上で葉文脈に閉じている．
- $>$ は T 上で代入に閉じている．

また，葉文脈のみならず，あらゆる文脈に閉じている準簡約化順序を**簡約化順序**と呼ぶ．

定理 2.2　R を TRS とし，$>$ を $\mathcal{T}(\Sigma,\mathcal{V})$ 上の簡約化順序とする．もし全ての書き換え規則 $l \to r \in R$ に対して $l > r$ ならば，R は停止性を持つ．

定義 2.3 (拡張関数)　R を STRS とし，$l \to r \in R$ に対して $\tau(l) = \alpha_1 \to \cdots \to \alpha_n \to \alpha$ かつ $\alpha \in \mathcal{S}^{nfun}$ とする．このとき，η 拡張関数 $\overline{\eta}$ を次のように定義する．

- $\overline{\eta}(l \to r) = \{l \to r, l[z_1] \to r[z_1], ..., l[z_1,...,z_n] \to r[z_1,...,z_n]\}$ ただし，$\forall i. \tau(z_i) = \alpha_i$ かつ $z_i \notin Var(l)$．
- $\overline{\eta}(R) = \bigcup_{l \to r \in R} \overline{\eta}(l \to r)$

定理 2.4　R を STRS とし，$>$ を $\mathcal{T}_\tau(\Sigma,\mathcal{V})$ 上の準簡約化順序とする．もし全ての書き換え規則 $l \to r \in \overline{\eta}(R)$ に対して $l > r$ ならば，R は停止性を持つ．

定義 2.1 において整礎性を直接証明するのは困難であるが，単純化順序の概念を利用すると比較的容易である．

定義 2.5 (単純化順序)　T を $T \subseteqq \mathcal{T}(\Sigma,\mathcal{V})$ となる任意の項の集

合とする．このとき，ある順序 $>$ が T 上で**単純化順序**であるとは，$>$ が以下の性質を満たすことである．

- $>$ は T 上で狭義の半順序である．
- $>$ は T 上で葉文脈に閉じている．
- $>$ は T 上で部分項性を持つ．
- $>$ は T 上で削除性を持つ．

定理 2.6 [1] 単純化順序は整礎である．

3. 高階関数を扱える辞書式経路順序の提案

本節では，高階関数を扱うことができる辞書式経路順序を 3 種類提案し，4. 節以降で解析するそれらの性質について結論を先に述べる．はじめに，それぞれの定義を与える．

定義 3.1 (辞書式経路順序 $>_{lpoL1}$) $\trianglerighteq$ を Σ 上の擬順序とし，$\sim$ を $\trianglerighteq$ から生成される同値関係とする．$s \equiv a[s_1,...,s_n]$, $t \equiv a'[t_1,...,t_m]$ に対して，以下のいずれかを満たすとき $s{>_{lpoL1}}t$ とする．

(1) $a \triangleright a'$ かつ $\forall j.s{>_{lpoL1}}t_j$

(2) $a \sim a' \in \Sigma$ かつ $[s_1,...,s_n] >^{lex}_{lpoL1} [t_1,...,t_m]$，かつ $\forall j.s{>_{lpoL1}}t_j$

(3) $\exists i.s_i \geqq_{lpoL1} t$

(4) $a = a' \in \mathcal{V}$ かつ $[s_1,...,s_n] >^{lex}_{lpoL1} [t_1,...,t_m]$，かつ $\forall j.s{>_{lpoL1}}t_j$

(5) $a' \in Var(s_1)$ かつ a は $\trianglerighteq$ に関して最大かつ $[s_1,...,s_n] >^{lex}_{lpoL1} [t_1,...,t_m]$，かつ $\forall j.s{>_{lpoL1}}t_j$

定義 3.2 (辞書式経路順序 $>_{lpoL2}$) $\trianglerighteq$ を Σ 上の擬順序とし，$\sim$ を $\trianglerighteq$ から生成される同値関係とする．$s \equiv a[s_1,...,s_n]$, $t \equiv a'[t_1,...,t_m]$ に対して，以下のいずれかを満たすとき $s{>_{lpoL2}}t$ とする．

(1) $a \triangleright a'$ かつ $\forall j.s{>_{lpoL2}}t_j$

(2) $a \sim a' \in \Sigma$ かつ $[s_1,...,s_n] >^{lex}_{lpoL2} [t_1,...,t_m]$，かつ $\forall j.s{>_{lpoL2}}t_j$

(3) $\exists i.s_i \geqq_{lpoL2} t$

(4) $a = a' \in \mathcal{V}$ かつ $[s_1,...,s_n] >^{lex}_{lpoL2} [t_1,...,t_m]$，かつ $\forall j.s{>_{lpoL2}}t_j$

(5) $a' \in Var(s_1)$ かつ a は $\trianglerighteq$ に関して最大かつ $\forall j.s{>_{lpoL2}}t_j$，かつ $\trianglerighteq$ に関して最大となるあらゆる $F \in \Sigma$ に対して $ar(a') > ar(F)$

なお，$>_{lpoL2}$ は引数関数 ar（定義 4.1）によって引数個数を定めているため，$\mathcal{T}(\Sigma,\mathcal{V})$ 上では定義されない．

定義 3.3 (辞書式経路順序 $>_{lpoR}$) $\trianglerighteq$ を Σ 上の擬順序とし，$\sim$ を $\trianglerighteq$ から生成される同値関係とする．$s \equiv a[s_1,...,s_n]$, $t \equiv a'[t_1,...,t_m]$ に対して，以下のいずれかを満たすとき $s \gtrsim_{lpoR} t$ とする．

(1) $a \triangleright a'$ かつ $\forall j.s{>_{lpoR}}t_j$

(2) $a \sim a' \in \Sigma$ かつ $[s_n,...,s_1] >^{lex}_{lpoR} [t_m,...,t_1]$，かつ $\forall j.s{>_{lpoR}}t_j$

(3) $\exists i.s_i \geqq_{lpoR} t$

(4) $a = a' \in \mathcal{V}$ かつ $[s_n,...,s_1] >^{lex}_{lpoR} [t_m,...,t_1]$，かつ $\forall j.s{>_{lpoR}}t_j$

(5) $a' \in Var(s)$ かつ a は $\trianglerighteq$ に関して最大かつ $[s_n,...,s_1] \geqq^{lex}_{lpoR} [t_m,...,t_1,a']$，かつ $\forall j.s{>_{lpoR}}t_j$

TRS R で扱う項の集合を $T(\subseteq\mathcal{T}(\Sigma,\mathcal{V}))$ とする．このとき，R の停止性を証明するためには，定理 2.2 と 2.4 より，3 種類の lpo（総じて $>_{lpo}$ と記す）に関して次の条件が必要である．

- **$>_{lpo}$ は T 上で簡約化順序または準簡約化順序である．**

そもそも，本来は型無し項の集合 $\mathcal{T}(\Sigma,\mathcal{V})$ 上で $>_{lpo}$ が簡約化順序であることが期待される．しかし，$a[1]{>_{lpo}}a[0,1]{>_{lpo}}a[0,0,1]$ のような無限列が存在するので整礎性を持たず，簡約化順序ではない．そこで，項の引数個数に上限を与えたクラス $\mathcal{T}^{\leqq}_{ar}(\Sigma,\mathcal{V})$（定義 4.7）を用意する．なお，仮に整礎であったとしても，残念ながらこのクラスでは根文脈に閉じていないので準簡約化順序となる．

ここで，準簡約化順序の性質の一つである整礎性を示すために以下の条件が望まれる．

- **$>_{lpo}$ は T 上で単純化順序である．**

ところが，$\mathcal{T}^{\leqq}_{ar}(\Sigma,\mathcal{V})$ 上では削除性が成立しないので単純化順序とならず，整礎性を直接証明することができない．そこで，項の引数個数を固定したクラス $\mathcal{T}_{ar}(\Sigma,\mathcal{V})$（定義 4.2）を用意する．このクラスでは削除性が保証されるので単純化順序となり，整礎性を示すことができる．さらに簡約化順序であることが示せる．この結果を用いると，$\mathcal{T}^{\leqq}_{ar}(\Sigma,\mathcal{V})$ 上で準簡約化順序であることが証明でき，さらに $\mathcal{T}_{\tau}(\Sigma,\mathcal{V})$ 上でも準簡約化順序となることが証明できる．

以上を踏まえて，4. 節以降では，3 種類の $>_{lpo}$ それぞれについて以下の順に証明を行う．

ステップ I $>_{lpo}$ は $\mathcal{T}_{ar}(\Sigma,\mathcal{V})$ 上で簡約化順序である．

ステップ II $>_{lpo}$ は $\mathcal{T}^{\leqq}_{ar}(\Sigma,\mathcal{V})$ 上で準簡約化順序である．

ステップ III $>_{lpo}$ は $\mathcal{T}_{\tau}(\Sigma,\mathcal{V})$ 上で準簡約化順序である．

なお，本稿で明らかとなる $>_{lpo}$ の性質をまとめると表 1 のようになる．

表 1 各項の集合上で $>_{lpo}$ が満たす性質

	単純化順序	準簡約化順序	簡約化順序
$\mathcal{T}(\Sigma,\mathcal{V})$	×	×	×
$\mathcal{T}_{\tau}(\Sigma,\mathcal{V})$	×	○	×
$\mathcal{T}^{\leqq}_{ar}(\Sigma,\mathcal{V})$	×	○	×
$\mathcal{T}_{ar}(\Sigma,\mathcal{V})$	○	○	○

4. 辞書式経路順序 $>_{lpoL1}$ に関する証明

本節では，$>_{lpoL1}$ に関してステップ I から III を順に示す．

4.1 ステップ I

定義 4.1 (引数関数) 引数関数 ar とは $\Sigma \cup \mathcal{V}$ から自然数の集合 N 上への関数である．本稿では，ar を τ によって以下のように定義する．なお，組化構成子 tp を便宜上 $tp_i (i \in N)$ で表す．すなわち，後に ar によって定義する $\mathcal{T}_{ar}(\Sigma, \mathcal{V})$ と $\mathcal{T}_{ar}^{\leqq}(\Sigma, \mathcal{V})$ においては，$\mathcal{T}_{\tau}(\Sigma, \mathcal{V})$ 上の $tp[t_1, ..., t_n]$ を $tp_n[t_1, ..., t_n]$ と解釈する．

$$ar(a) = \begin{cases} n & \text{if } \tau(a) = \alpha_1 \to \cdots \to \alpha_n \to \alpha \text{ かつ} \\ & \quad \alpha \in \mathcal{S}^{nfun} \text{ かつ } a \not\equiv tp_i \\ i & \text{if } a = tp_i \end{cases}$$

定義 4.2 ($\mathcal{T}_{ar}(\Sigma, \mathcal{V})$) ar によって引数個数を定めた項全体からなる集合 $\mathcal{T}_{ar}(\Sigma, \mathcal{V})$ を次のように帰納的に定義する．

- $n = ar(a)$ かつ $t_1, ..., t_n \in \mathcal{T}_{ar}(\Sigma, \mathcal{V})$ ならば $a[t_1, ..., t_n] \in \mathcal{T}_{ar}(\Sigma, \mathcal{V})$

補題 4.3 $s, t \in \mathcal{T}_{ar}(\Sigma, \mathcal{V})$ かつ $s >_{lpoL1} t$ ならば $Var(s) \supseteqq Var(t)$.

証明 $|s| + |t|$ に関する帰納法で証明できる． □

定理 4.4 $\mathcal{T}_{ar}(\Sigma, \mathcal{V})$ 上で，$>_{lpoL1}$ は単純化順序である．

証明 部分項性を持つことは (3) から明らかであり，部分項性と推移性を用いると，非反射性を示すことができる．また，葉文脈に閉じていることは (2) と (4) から，削除性を持つことは a[..., t, ...] と a[..., ...] が $\mathcal{T}_{ar}(\Sigma, \mathcal{V})$ で共存しないことから無条件に成立する．以上のことから，$>_{lpoL1}$ が単純化順序であるためには，推移性を示せば十分である．

$t_1 >_{lpoL1} t_2 >_{lpoL1} t_3$ として $|t_1| + |t_2| + |t_3|$ に関する帰納法で $t_1 >_{lpoL1} t_3$ を示す．ただし本稿では，$t_1 >_{lpoL1} t_2$ が (2)，$t_2 >_{lpoL1} t_3$ が (5) で順序付けられた場合のみを示す．このとき，a_2 は $\trianglerighteq$ に関して最大であり，$a_1 \sim a_2$ だから a_1 も $\trianglerighteq$ に関して最大である．また，$[t_{11}, ..., t_{1k_1}] >_{lpoL1}^{lex} [t_{21}, ..., t_{2k_2}] \geqq_{lpoL1}^{lex} [t_{31}, ..., t_{3k_3}]$ と帰納法の仮定より，$[t_{11}, ..., t_{1k_1}] >_{lpoL1}^{lex} [t_{31}, ..., t_{3k_3}]$．さらに前提より，$t_1 >_{lpoL1} t_2$ かつ $\forall j. t_2 >_{lpoL1} t_{3j}$．帰納法の仮定より，$\forall j. t_1 >_{lpoL1} t_{3j}$．また，$[t_{11}, ..., t_{1k_1}] >_{lpoL1}^{lex} [t_{21}, ..., t_{2k_2}]$ より，$t_{11} \sim t_{21}$ または $t_{11} >_{lpoL1} t_{21}$．$t_{11} \sim t_{21}$ のときは，前提より $a_3 \in Var(t_{21})$ なので $a_3 \in Var(t_{11})$．$t_{11} >_{lpoL1} t_{21}$ のときは，補題 4.3 より $Var(t_{11}) \supseteqq Var(t_{21})$ なので $a_3 \in Var(t_{21}) \subseteqq Var(t_{11})$．従って，いずれの場合も $a_3 \in Var(t_{11})$．(5) より，$t_1 >_{lpoL1} t_3$． □

補題 4.5 $v, t \in \mathcal{T}_{ar}(\Sigma, \mathcal{V})$ かつ $v \in Var(t)$ ならば $v \equiv root(t)$ または $t >_{lpoL1} v$.

証明 $|t|$ に関する帰納法で証明できる． □

定理 4.6 $\mathcal{T}_{ar}(\Sigma, \mathcal{V})$ 上で，$>_{lpoL1}$ は簡約化順序である．

証明 代入に閉じていることを示せば十分である．

$s \equiv a[s_1, ..., s_n] >_{lpoL1} a'[t_1, ..., t_m] \equiv t$ として，$|s| + |t|$ に関する帰納法で $s\theta >_{lpoL1} t\theta$ を示す．ただし本稿では，$s >_{lpoL1} t$ が (5) で順序付けられた場合のみを示す．$a'\theta \equiv a''[u_1, ..., u_k]$ とおき，次の 5 つに場合分けする．

- $a \triangleright a''$ となる場合を考える．このとき，$Var(s_1) \subseteqq Var(s)$ なので $a' \in Var(s)$．補題 4.5 と $root(s) \equiv a (\in \Sigma) \not\equiv a' (\in V)$ より，$s >_{lpoL1} a'$．帰納法の仮定と部分項性より $\forall j. s\theta >_{lpoL1} a'\theta >_{lpoL1} u_j$ であり，推移性より $\forall j. s\theta >_{lpoL1} u_j$．また，帰納法の仮定より $\forall j. s\theta >_{lpoL1} t_j\theta$．(1) より $s\theta >_{lpoL1} t\theta$．
- $a \sim a''$ かつ $k = 0$ となる場合を考える．帰納法の仮定より，明らかに $[s_1\theta, ..., s_n\theta] >_{lpoL1}^{lex} [t_1\theta, ..., t_m\theta]$．(2) より $s\theta >_{lpoL1} t\theta$．
- $a \sim a''$ かつ $k > 0$ となる場合を考える．$a' \in Var(s_1)$ と補題 4.5 より $a' \equiv root(s_1)$ または $s_1 >_{lpoL1} a'$．$a' \equiv root(s_1)$ のときは，部分項性より $s_1\theta >_{lpoL1} u_1$．$s_1 >_{lpoL1} a'$ のときは，帰納法の仮定と部分項性より $s_1\theta >_{lpoL1} a'\theta >_{lpoL1} u_1$ であり，推移性より $s_1\theta >_{lpoL1} u_1$．従って，いずれの場合も $[s_1\theta, ..., s_n\theta] >_{lpoL1}^{lex} [u_1, ..., u_k, t_1\theta, ..., t_m\theta]$．ゆえに，(2) より $s\theta >_{lpoL1} t\theta$．
- $a'' \in \mathcal{V}$ かつ $k = 0$ となる場合を考える．$a' \in Var(s_1)$ より，$a'' \in Var(s_1\theta)$．帰納法の仮定より，$[s_1\theta, ..., s_n\theta] \geqq_{lpoL1}^{lex} [t_1\theta, ..., t_m\theta]$ かつ $\forall j. s\theta >_{lpoL1} t_j\theta$．(5) より $s\theta >_{lpoL1} t\theta$．
- $a'' \in \mathcal{V}$ かつ $k > 0$ となる場合を考える．$a' \in Var(s_1)$ より，$a'' \in Var(s_1\theta)$．また，$a' \in Var(s_1)$ と補題 4.5 より $a' \equiv root(s_1)$ または $s_1 >_{lpoL1} a'$．$a' \equiv root(s_1)$ のときは，部分項性より $\forall j. s\theta >_{lpoL1} s_1\theta >_{lpoL1} u_j$．$s_1 >_{lpoL1} a'$ のときは，帰納法の仮定と部分項性より $\forall j. s\theta >_{lpoL1} s_1\theta >_{lpoL1} a'\theta >_{lpoL1} u_j$．従って，いずれの場合も推移性より $s_1\theta >_{lpoL1} u_1$ なので $[s_1\theta, ..., s_n\theta] >_{lpoL1}^{lex} [u_1, ..., u_k, t_1\theta, ..., t_m\theta]$．同じく推移性より，$\forall j. s\theta >_{lpoL1} u_j$．帰納法の仮定より，$\forall j. s\theta >_{lpoL1} t_j\theta$．(5) より $s\theta >_{lpoL1} t\theta$． □

4.2 ステップ II

本節の証明に 4.1 節の結果を使うために，特別な関数記号 $\perp$ と，$\mathcal{T}_{ar}^{\leqq}(\Sigma, \mathcal{V})$ から $\mathcal{T}_{ar}(\Sigma, \mathcal{V})$ への写像関数 δ を用意する．

定義 4.7 ($\mathcal{T}_{ar}^{\leqq}(\Sigma, \mathcal{V})$) ar によって引数個数の上限を定めた項全体からなる集合 $\mathcal{T}_{ar}^{\leqq}(\Sigma, \mathcal{V})$ を次のように帰納的に定義する．

- $n \leqq ar(a)$ かつ $t_1, ..., t_n \in \mathcal{T}_{ar}^{\leqq}(\Sigma, \mathcal{V})$ ならば $a[t_1, ..., t_n] \in \mathcal{T}_{ar}^{\leqq}(\Sigma, \mathcal{V})$

定義 4.8 (写像関数) $\perp$ を特別な関数記号とする．このとき，$\mathcal{T}_{ar}^{\leqq}(\Sigma, \mathcal{V})$ から $\mathcal{T}_{ar}(\Sigma, \mathcal{V})$ への写像関数 δ を次のように定義する．

- $\delta(a[t_1, ..., t_k]) = a[\overbrace{\delta(t_1), ..., \delta(t_k), \perp, ..., \perp}^{ar(a)}]$

補題 4.9 $t \in \mathcal{T}_{ar}^{\leqq}(\Sigma, \mathcal{V})$ ならば $\delta(t) \in \mathcal{T}_{ar}(\Sigma, \mathcal{V})$.

証明 $|t|$ に関する帰納法で証明できる． □

補題 4.10 $s, t \in \mathcal{T}_{ar}^{\leqq}(\Sigma, \mathcal{V})$ に対し $s >_{lpoL1} t \Leftrightarrow \delta(s) >_{lpoL1} \delta(t)$.

証明 $|s| + |t|$ に関する帰納法で証明できる． □

補題 4.11　代入 θ_δ を $a\theta \equiv a'[u_1, ..., u_m]$ に対して $\theta_\delta(a) = a'[\delta(u_1), ..., \delta(u_m)]$ で定義する．このとき，任意の項 t に対して $\delta(t\theta) = \delta(t)\theta_\delta$.

証明　$|t|$ に関する帰納法で証明できる．□

補題 4.12　$\mathcal{T}_{ar}^{\leqq}(\Sigma, \mathcal{V})$ 上で，$>_{lpoL1}$ は代入に閉じている．

証明　$s>_{lpoL1}t$ ならば $s\theta>_{lpoL1}t\theta$ であることを示す．補題 4.10 より，$\delta(s)>_{lpoL1}\delta(t)$．$\mathcal{T}_{ar}(\Sigma, \mathcal{V})$ 上で代入に閉じているので，$\delta(s)\theta_\delta>_{lpoL1}\delta(t)\theta_\delta$．補題 4.11 より，$\delta(s\theta)>_{lpoL1}\delta(t\theta)$．再度補題 4.10 より，$s\theta>_{lpoL1}t\theta$．□

定理 4.13　$\mathcal{T}_{ar}^{\leqq}(\Sigma, \mathcal{V})$ 上で，$>_{lpoL1}$ は準簡約化順序である．

証明　補題 4.12 より，代入に閉じている．部分項性を持つことは (3) から明らかであり，部分項性と推移性を用いると，非反射性を示すことができる．また，葉文脈に閉じていることは (2) と (4) から明らかである．以上のことから，$>_{lpoL1}$ が準簡約化順序であるためには，推移性と整礎性を示せば十分である．

- 推移性を示す．$t_1>_{lpoL1}t_2>_{lpoL1}t_3$ として $t_1>_{lpoL1}t_3$ を示す．補題 4.10 より，$\delta(t_1)>_{lpoL1}\delta(t_2)>_{lpoL1}\delta(t_3)$．$\mathcal{T}_{ar}(\Sigma, \mathcal{V})$ 上で推移性を持つので，$\delta(t_1)>_{lpoL1}\delta(t_3)$．再度補題 4.10 より，$t_1>_{lpoL1}t_3$.
- 整礎性を示す．$>_{lpoL1}$ が整礎でないと仮定する．このとき，$t_i (i \in N) \in \mathcal{T}_{ar}^{\leqq}(\Sigma, \mathcal{V})$ に対して $t_1>_{lpoL1}t_2>_{lpoL1}t_3>_{lpoL1}...$ となる無限列が存在する．すると，補題 4.10 より $\delta(t_1)>_{lpoL1}\delta(t_2)>_{lpoL1}\delta(t_3)>_{lpoL1}....$ これは，$\mathcal{T}_{ar}(\Sigma, \mathcal{V})$ 上で $>_{lpoL1}$ が整礎であることに矛盾．□

4.3　ステップ III

本節の証明に 4.1 節の結果を使うために，$\mathcal{T}_\tau(\Sigma, \mathcal{V})$ から $\mathcal{T}_{ar}^{\leqq}(\Sigma, \mathcal{V})$ への写像関数 ϕ を用意する．

定義 4.14 (写像関数)　$\mathcal{T}_\tau(\Sigma, \mathcal{V})$ から $\mathcal{T}_{ar}^{\leqq}(\Sigma, \mathcal{V})$ への写像関数 ϕ は以下のように定義できる．

$$\phi(a[t_1, ..., t_n]) = \begin{cases} a[\phi(t_1), ..., \phi(t_n)] & \text{if } a \neq tp \\ tp_n[\phi(t_1), ..., \phi(t_n)] & \text{if } a = tp \end{cases}$$

なお，各 tp_i は Σ 上で同一優先順位とする．

補題 4.15　$t \in \mathcal{T}_\tau(\Sigma, \mathcal{V})$ ならば $\phi(t) \in \mathcal{T}_{ar}^{\leqq}(\Sigma, \mathcal{V})$.

証明　$|t|$ に関する帰納法で証明できる．□

補題 4.16　$s, t \in \mathcal{T}_\tau(\Sigma, \mathcal{V})$ に対し $s>_{lpoL1}t \Leftrightarrow \phi(s)>_{lpoL1}\phi(t)$.

証明　$|s| + |t|$ に関する帰納法で証明できる．□

補題 4.17　代入 θ_ϕ を $\theta_\phi(a) = \phi(\theta(a))$ で定義する．このとき，任意の項 t に対して $\phi(t\theta) = \phi(t)\theta_\phi$.

証明　$|t|$ に関する帰納法で証明できる．□

定理 4.18　$\mathcal{T}_\tau(\Sigma, \mathcal{V})$ 上で，$>_{lpoL1}$ は準簡約化順序である．

証明　補題 4.10 の代わりに補題 4.16，補題 4.11 の代わりに補題 4.17 を用いると，定理 4.13 と同様に示すことができる．□

5.　辞書式経路順序 $>_{lpoL2}$ に関する証明

本節では，$>_{lpoL2}$ に関してステップ I から III を順に示す．

5.1　ステップ I

補題 5.1　$s, t \in \mathcal{T}_{ar}(\Sigma, \mathcal{V})$ かつ $s>_{lpoL2}t$ ならば $Var(s) \supseteqq Var(t)$.

証明　$|s| + |t|$ に関する帰納法で証明できる．□

定理 5.2　$\mathcal{T}_{ar}(\Sigma, \mathcal{V})$ 上で，$>_{lpoL2}$ は単純化順序である．

証明　定理 4.4 と同様に証明できる．□

補題 5.3　$v, t \in \mathcal{T}_{ar}(\Sigma, \mathcal{V})$ かつ $v \in Var(t)$ ならば $v \equiv root(t)$ または $t>_{lpoL2}v$.

証明　$|t|$ に関する帰納法で証明できる．□

定理 5.4　$\mathcal{T}_{ar}(\Sigma, \mathcal{V})$ 上で，$>_{lpoL2}$ は簡約化順序である．

証明　代入に閉じていることを示せば十分である．

$s \equiv a[s_1, ..., s_n]>_{lpoL2}a'[t_1, ..., t_m] \equiv t$ として，$|s| + |t|$ に関する帰納法で $s\theta>_{lpoL2}t\theta$ を示す．ただし本稿では，$s>_{lpoL2}t$ が (5) で順序付けられた場合のみを示す．$a'\theta \equiv a''[u_1, ..., u_k]$ とおく．ここで，$a \in \Sigma$ のときは定理 4.6 と同様に証明できるので，次の 2 通りを考えれば十分である．

- $a'' \in \mathcal{V}$ かつ $k = 0$ となる場合を考える．$a' \in Var(s_1)$ より，$a'' \in Var(s_1\theta)$．帰納法の仮定より，$\forall j.s\theta>_{lpoL2}t_j\theta$．また，k=0 より $ar(a') = ar(a'')$ だから，$\trianglerighteq$ に関して最大となるあらゆる $F \in \Sigma$ に対して $ar(a'') < ar(F)$．(5) より $s\theta>_{lpoL2}t\theta$.
- $a'' \in \mathcal{V}$ かつ $k > 0$ となる場合を考える．$a' \in Var(s_1)$ より，$a'' \in Var(s_1\theta)$．また，$a' \in Var(s_1)$ と補題 5.3 より $a' \equiv root(s_1)$ または $s_1>_{lpoL2}a'$．$a' \equiv root(s_1)$ のときは，部分項性より $\forall j.s\theta>_{lpoL2}s_1\theta>_{lpoL2}u_j$．$s_1>_{lpoL2}a'$ のときは，帰納法の仮定と部分項性より $\forall j.s\theta>_{lpoL2}s_1\theta>_{lpoL2}a'\theta>_{lpoL2}u_j$．従って，いずれの場合も推移性より $\forall j.s\theta>_{lpoL2}u_j$．帰納法の仮定より，$\forall j.s\theta>_{lpoL2}t_j\theta$．(5) より $s\theta>_{lpoL2}t\theta$．□

5.2　ステップ II

補題 5.5　$s, t \in \mathcal{T}_{ar}^{\leqq}(\Sigma, \mathcal{V})$ に対し $s>_{lpoL2}t \Leftrightarrow \delta(s)>_{lpoL2}\delta(t)$.

証明　$|s| + |t|$ に関する帰納法で証明できる．□

定理 5.6　$\mathcal{T}_{ar}^{\leqq}(\Sigma, \mathcal{V})$ 上で，$>_{lpoL2}$ は準簡約化順序である．

証明　補題 4.10 の代わりに補題 5.5 を用いると，定理 4.13 と同様に示すことができる．□

5.3　ステップ III

補題 5.7　$s, t \in \mathcal{T}_\tau(\Sigma, \mathcal{V})$ に対し $s >_{lpoL2} t \Leftrightarrow \phi(s) >_{lpoL2} \phi(t)$.

証明　$|s| + |t|$ に関する帰納法で証明できる．□

定理 5.8　$\mathcal{T}_\tau(\Sigma, \mathcal{V})$ 上で，$>_{lpoL2}$ は準簡約化順序である．

証明　補題 4.16 の代わりに補題 5.7 を用いると，定理 4.18 と同様に示すことができる．□

6.　辞書式経路順序 $>_{lpoR}$ に関する証明

本節では，$>_{lpoR}$ に関してステップ I から III を順に示す．

6.1　ステップ I

補題 6.1　$s, t \in \mathcal{T}_{ar}(\Sigma, \mathcal{V})$ かつ $s >_{lpoR} t$ ならば $Var(s) \supseteqq Var(t)$.

証明　$|s| + |t|$ に関する帰納法で証明できる．□

定理 6.2　$\mathcal{T}_{ar}(\Sigma, \mathcal{V})$ 上で，$>_{lpoR}$ は単純化順序である．

証明　推移性を示せば十分である．他の性質は，定理 4.4 と同様に証明できる．

$t_1 >_{lpoR} t_2 >_{lpoR} t_3$ として，$|t_1| + |t_2| + |t_3|$ に関する帰納法で $t_1 >_{lpoR} t_3$ を示す．ただし本稿では，$t_1 >_{lpoR} t_2$ が (2), $t_2 >_{lpoR} t_3$ が (5) で順序付けられた場合のみを示す．このとき，$\forall j. t_1 >_{lpoR} t_{3j}$ かつ a_1 が $\trianglerighteq$ に関して最大であることは定理 4.4 と同様に証明できる．また，$[t_{1k_1}, ..., t_{11}] >^{lex}_{lpoR} [t_{2k_2}, ..., t_{22}] >^{lex}_{lpoR} [t_{3k_3}, ..., t_{33}, a_3]$ と帰納法の仮定より，$[t_{1k_1}, ..., t_{11}] >^{lex}_{lpoR} [t_{3k_1}, ..., t_{33}, a_3]$．前提より，$t_1 >_{lpoR} t_2$ なので，補題 6.1 より $Var(t_1) \supseteqq Var(t_2)$．これと $a_3 \in Var(t_2)$ より，$a_3 \in Var(t_1)$．(5) より，$t_1 >_{lpoR} t_3$．□

補題 6.3　$v, t \in \mathcal{T}_{ar}(\Sigma, \mathcal{V})$ かつ $v \in Var(t)$ ならば $v \equiv root(t)$ または $t >_{lpoR} v$.

証明　$|t|$ に関する帰納法で証明できる．□

定理 6.4　$\mathcal{T}_{ar}(\Sigma, \mathcal{V})$ 上で，$>_{lpoR}$ は簡約化順序である．

証明　代入に閉じていることを示せば十分である．

$s \equiv a[s_1, ..., s_n] >_{lpoR} a'[t_1, ..., t_m] \equiv t$ として，$|s| + |t|$ に関する帰納法で $s\theta >_{lpoR} t\theta$ を示す．ただし本稿では，$s >_{lpoR} t$ が (5) で順序付けられた場合のみを示す．$a'\theta \equiv a''[u_1, ..., u_k]$ とおく．ここで，$a \in \Sigma$ のときは定理 4.6 と同様に証明できるので，次の 2 通りを考えれば十分である．

- $a'' \in \mathcal{V}$ かつ $k = 0$ となる場合を考える．$a' \in Var(s)$ より，$a'' \in Var(s\theta)$．帰納法の仮定より，$\forall j. s\theta >_{lpoR} t_j\theta$ かつ $[s_n\theta, ..., s_1\theta] >^{lex}_{lpoR} [t_m\theta, ..., t_1\theta, a'']$．(5) より $s\theta >_{lpoR} t\theta$.
- $a'' \in \mathcal{V}$ かつ $k > 0$ となる場合を考える．$a' \in Var(s)$ より，$a'' \in Var(s\theta)$．また，$a' \in Var(s)$ と補題 6.3 より $a' \equiv root(s)$ または $s >_{lpoR} a'$．$a' \equiv root(s)$ のときは，部分項性より $\forall j. s\theta >_{lpoR} u_j$．$s >_{lpoR} a'$ のときは，帰納法の仮定と部分項性より $\forall j. s\theta >_{lpoR} a'\theta >_{lpoR} u_j$．従って，いずれの場合も推移性より $\forall j. s\theta >_{lpoR} u_j$．帰納法の仮定より，$\forall j. s\theta >_{lpoR} t_j\theta$ かつ $[s_n\theta, ..., s_1\theta] >^{lex}_{lpoR} [t_m\theta, ..., t_1\theta, a'\theta]$．また，部分項性より $a'\theta >_{lpoR} u_k$ なので，$[t_m\theta, ..., t_1\theta, a'\theta] >^{lex}_{lpoR} [t_m\theta, ..., t_1\theta, u_k, ..., u_1, a'']$ となり，推移性より $[s_n\theta, ..., s_1\theta] >^{lex}_{lpoR} [t_m\theta, ..., t_1\theta, u_k, ..., u_1, a'']$．(5) より $s\theta >_{lpoR} t\theta$. □

6.2　ステップ II

補題 6.5　$s, t \in \mathcal{T}^{\leqq}_{ar}(\Sigma, \mathcal{V})$ に対し $s >_{lpoR} t \Leftrightarrow \delta(s) >_{lpoR} \delta(t)$.

証明　$|s| + |t|$ に関する帰納法で証明できる．□

定理 6.6　$\mathcal{T}^{\leqq}_{ar}(\Sigma, \mathcal{V})$ 上で，$>_{lpoR}$ は準簡約化順序である．

証明　補題 4.10 の代わりに補題 6.5 を用いると，定理 4.13 と同様に示すことができる．□

6.3　ステップ III

補題 6.7　$s, t \in \mathcal{T}_\tau(\Sigma, \mathcal{V})$ に対し $s >_{lpoR} t \Leftrightarrow \phi(s) >_{lpoR} \phi(t)$.

証明　$|s| + |t|$ に関する帰納法で証明できる．□

定理 6.8　$\mathcal{T}_\tau(\Sigma, \mathcal{V})$ 上で，$>_{lpoR}$ は準簡約化順序である．

証明　補題 4.16 の代わりに補題 6.7 を用いると，定理 4.18 と同様に示すことができる．□

7.　今後の課題

本稿で提案した辞書式経路順序は，高階関数の扱いが可能であることに加えて，項の型情報に依存しないという利点もある．しかし，これらの順序を単独で用いた場合は実用的に不十分である．

一方，STRS の停止性証明法に依存対法がある [2]．依存対法では，切り落とし法と呼ばれる手法を用いて簡約化対 $(\gtrsim, >)$ を生成する際に経路順序が必要である．そこで，この依存対法に本稿の辞書式経路順序を組み合わせることで，実用的な効果が期待できる．

謝辞　本研究は一部，科研費#15500007，#16650005，#17700009 ならびに名古屋大学 21 世紀 COE プログラム（社会情報基盤のための音声・映像の知的統合）の補助を受けている．

文　献

[1] N.Dershowitz, Orderings for Term-rewriting Systems, Theoretical Computer Science, Vol.17, pp.279–301, 1982.
[2] K.Kusakari, On Proving Termination of Term Rewriting Systems with Higher-Oeder Variables, IPSJ Transactions on Programming, Vol.42, No.SIG 7(PRO 11), pp.35–45, 2001.
[3] K.Kusakari, M.Sakai, Enhancing Dependency Pair Method by Strong Computability in Simply-Typed Term Rewriting Systems, 電気情報通信学会技術研究報告 (SS2005-65), Vol.105, No.491, pp.13–18, 2005.
[4] Terese, Term Rewriting Systems, Cambridge Tracts in Theoretical Computer Science, Vol.55, Cambridge University Press, 2003.