Vol. 36 No. 12 情報処理学会論文誌 Dec. 1995

テクニカルノート

拡張ステータスによる項書換え系の停止性検証

能登 正人† 栗原 正仁† 大内 東†

項書換え系の停止性の検証は一般には決定不能であるが，単純化順序を用いることにより，一部の（しかし，実用上重要な多くの）系に対して停止性の機械的検証が可能である．例えば，（ステータス付き）再帰経路順序（RPOS）は，関数記号の集合上の優先順位と各関数記号のステータスに基づいて項の集合上の単純化順序を定義する．本論文では，このステータスの概念を拡張し，従来の順序よりも検証に成功しやすい拡張ステータス付き再帰経路順序（RPOES）を提案し，それが単純化順序であることを証明し，その有効性について述べる．

Termination Verification of Term Rewriting Systems by Extended Status

MASATO NOTO,† MASAHITO KURIHARA† and AZUMA OHUCHI†

Termination of term rewriting systems is undecidable in general, but simplification orderings allow us to verify the termination of a part of (but a practically important part of) rewrite systems. For example, recursive path orderings with status (RPOS) define a simplification ordering on a set of terms, based on precedence on and status of function symbols. This paper extends the notion of status, and defines recursive path orderings with extended status (RPOES). It is shown that RPOES are simplification orderings and more effective than RPOS.

1. はじめに

等式系は，定理自動証明，代数的仕様記述，関数型言語などの理論的基礎として，ソフトウェアのメカニズムや性質を明らかにするために用いられ非常に重要である．等式系に基づく計算モデルとして，等式を向きの付いた書換え規則とみなし，項を書換えることによって計算を行う項書換え系（term rewriting system: TRS）[3),5),6)]がある．TRS の重要な性質の一つとして停止性がある．TRS の停止性は無限に書換えが続かないことを意味する．停止性の検証は一般には決定不能な問題であるが，単純化順序を用いることにより，一部の（しかし，実用上重要な多くの）TRS に対して停止性の機械的検証が可能である．単純化順序として経路順序とよばれるクラスのものがいくつか提案されている．例えば，（ステータス付き）再帰経路順序（RPOS）[7)]が代表的なものであり，それらは，関数記号の集合上に導入した優先順位[3)]とよばれる半順序と各関数記号ごとに割り当てた *mult*, *left*, *right* のいずれかのステータスに基づいて項の集合上の単純化順序を定義する．本論文では，このステータスの概念を拡張し，従来の順序よりも検証に成功しやすい拡張ステータス付き再帰経路順序（RPOES）を提案し，それが単純化順序であることを証明し，その有効性について述べる．

† 北海道大学工学部システム情報工学専攻
Institute of Systems and Information Engineering, Faculty of Engineering, Hokkaido University

2. 項書換え系の停止性

2.1 項書換え系

関数記号の集合を $\mathcal{F}$，変数の集合を $\mathcal{V}$ で表す．各関数記号 $f \in \mathcal{F}$ には，引数の個数を示す項数とよばれる非負整数が割り当てられている．$\mathcal{F}, \mathcal{V}$ から構成される項の集合を $\mathcal{T}(\mathcal{F}, \mathcal{V})$ とする．

代入 θ は，$\mathcal{V}$ から $\mathcal{T}(\mathcal{F}, \mathcal{V})$ への写像である．θ は $\mathcal{T}(\mathcal{F}, \mathcal{V})$ から $\mathcal{T}(\mathcal{F}, \mathcal{V})$ への写像に拡張される．すなわち，$t \equiv f(t_1, \ldots, t_n)$ とすると，$\theta(t) \equiv f(\theta(t_1), \ldots, \theta(t_n))$ である．以後，$\theta(t)$ を $t\theta$ と表す．

特別な定数記号 □ を一つ含む項を文脈といい，$C[\,]$ で表す．$C[\,]$ の □ を項 s で置き換えて得られる項を $C[s]$ と表す．ここで，項 s を $C[s]$ の部分項という．

特に $C \not\equiv \square$ のとき項 s は $C[s]$ の真部分項である．

書換え規則は，$\ell \to r$ の形をした $\mathcal{T}(\mathcal{F},\mathcal{V})$ 中の項の順序対である．ただし，ℓ は変数でなく，r に含まれる変数は ℓ にも含まれる．

項書換え系（TRS）$\mathcal{R}$ は書換え規則の集合である．$\ell \to r$ が $\mathcal{R}$ の書換え規則であり，$s \equiv C[\ell\theta]$，$t \equiv C[r\theta]$ となるような文脈 C と代入 θ があるとき，$s \to_{\mathcal{R}} t$ と表し，項 s は項 t に書換えられるという．

2.2 停止性

TRS $\mathcal{R}$ は，もし $t_0 \to_{\mathcal{R}} t_1 \to_{\mathcal{R}} \cdots$ のような無限簡約列が存在しなければ停止するといわれる．TRS の停止性を検証する一般的な方法は存在しないが，本論文では，Dershowitz [1),2)] によって提案された単純化順序を用いた停止性検証法を考える．

定義 2.1（単純化順序） 項を $s = f(s_1,\ldots,s_m)$，$t = g(t_1,\ldots,t_n)$ とする．$\mathcal{T}(\mathcal{F},\mathcal{V})$ 上の半順序（推移性と非反射性を満たす二項関係）$\succ$ は，次の二つの性質（置換，部分項）を満たすならば単純化順序である．

(1) $s \succ t \quad \Rightarrow \quad f(\ldots,s,\ldots) \succ f(\ldots,t,\ldots)$

(2) $f(\ldots,t,\ldots) \succ t$

単純化順序と以下の定理[1),2)]を用いて TRS の停止性を検証できる．

定理 2.1（停止定理） 有限の TRS：$\mathcal{R} = \{\ell_i \to r_i | 1 \le i \le n\}$ は任意の代入 θ に対して，

$$\ell_i\theta \succ r_i\theta, \quad \text{for all } i \in \{1,\ldots,n\}$$

となる単純化順序 $\succ$ が $\mathcal{T}(\mathcal{F},\mathcal{V})$ 上に存在するならば停止する．

2.3 多重集合順序と辞書式順序

S 上での多重集合（multiset）は，S の要素を重複を許して集めた集合である．S 上のすべての有限な多重集合の集合を $\mathcal{M}(S)$ とする．任意の（厳格）半順序集合 $(S,\succ)$ が与えられたとき，$\mathcal{M}(S)$ 上の多重集合順序 $(\mathcal{M}(S),\succ\!\succ)$ は以下に定義される．

定義 2.2（多重集合順序） 集合 S 上の多重集合を $M, N \in \mathcal{M}(S)$ とする．以下のすべての条件を満たす有限な多重集合 $X, Y \in \mathcal{M}(S)$ が存在するとき（またそのときに限り）$M \succ\!\succ N$ である．

- $\{\,\} \ne X \subseteq M$
- $N = (M - X) \cup Y$
- $(\forall y \in Y)(\exists x \in X)\ x \succ y$

Dershowitz と Manna による以下の重要な定理[4)]がある．

定理 2.2（Dershowitz & Manna） $(S,\succ)$ が半順序ならば $(\mathcal{M}(S),\succ\!\succ)$ も半順序である．

S の要素 $a_i \in S$ $(i = 1,2,\ldots,m)$ の列を $(a_1,a_2,\ldots,a_m)$ と書く．S の要素の有限列の集合を S^* とする．半順序 $(S,\succ)$ が与えられたとき，これを S^* 上の辞書式順序 $(S^*,\succ_{lex})$ に以下のように拡張できる．

定義 2.3（辞書式順序） $(a_1,\ a_2,\ \ldots,\ a_m) \succ_{lex} (b_1,\ b_2,\ \ldots,\ b_n)$ であることは以下の条件のいずれかを満たすときに限る．

(1) $m > 0 \quad \wedge \quad n = 0$

(2) $a_1 \succ b_1$

(3) $a_1 = b_1 \quad \wedge \quad (a_2,\ldots,a_m) \succ_{lex} (b_2,\ldots,b_n)$

2.4 再帰経路順序

単純化順序として種々の経路順序[9)]が提案されている．経路順序の中でも定義の簡潔さや効率の面から再帰経路順序（recursive path ordering: RPO）[3),6)]がよく使用される．これは，優先順位（precedence）とよばれる $\mathcal{F}$ 上の半順序 $>$ を用いて，$\mathcal{T}(\mathcal{F},\mathcal{V})$ 上の単純化順序 $\succ$ を定義するものである．しかしながら，検証できる範囲が狭いといったマイナス面もある．そこで，ステータスとよばれるものによって，元の経路順序を拡張することができる．

ステータスは各関数記号ごとにもっていて，比較すべき二つの項の最外（最左）の関数記号が同一のときの引数の比較方法を示している．ステータスの種類としては，多重集合順序，左辞書式順序，右辞書式順序の 3 種類あり，慣習的にそれぞれ $mult$，$left$，$right$ の記号を用いる．RPO のステータス付き経路順序として RPOS があり，以下に示す $\succ$ として定義される．

定義 2.4（RPOS） 項を $s = f(s_1,\ldots,s_m)$，$t = g(t_1,\ldots,t_n)$ とする．このとき $s \succ t$ であることは以下の条件のいずれかを満たすときに限る．

(1) $s_i \succeq t \quad \text{for some } i \in \{1,\ldots,m\}$

(2) $f > g \quad \wedge \quad s \succ t_j \quad \text{for all } j \in \{1,\ldots,n\}$

(3) $f = g \quad \wedge \quad f^\sigma = mult \quad \wedge$
$\qquad \{s_1,\ldots,s_m\} \succ\!\succ \{t_1,\ldots,t_n\}$

(4) $f = g \quad \wedge \quad f^\sigma = left \quad \wedge$
$\qquad (s_1,\ldots,s_m) \succ_{lex} (t_1,\ldots,t_n) \quad \wedge$
$\qquad s \succ t_j \quad \text{for all } j \in \{1,\ldots,n\}$

(5) $f = g \quad \wedge \quad f^\sigma = right \quad \wedge$
$\qquad (s_m,\ldots,s_1) \succ_{lex} (t_n,\ldots,t_1) \quad \wedge$
$\qquad s \succ t_j \quad \text{for all } j \in \{1,\ldots,n\}$

ただし，$>$ は $\mathcal{F}$ 上の優先順位，f^σ は f のステータス，$\succeq$ は $\succ$ と $=$ の和，$\succ\!\succ$ は $\succ$ の多重集合順序への拡張，$\succ_{lex}$ は $\succ$ の辞書式順序への拡張である．

3. 拡張ステータス

3.1 拡張ステータス

従来の $left, right$ ステータスは，引数の比較する

順序を辞書式順序で1引数ずつ比較していたが，本論文では，いくつかの引数をまとめて多重集合順序で比較できるように，ステータスの概念を拡張する．

定義 3.1（拡張ステータス）

(1) 関数記号 f の拡張ステータス f^σ はある列 $(I_1,\ldots,I_c)$ である．ただし，f の項数が m のとき，$I_1,\ldots,I_c$ は集合 $\{1,\ldots,m\}$ の被覆（cover）であるとする．すなわち

- $I_i \subseteq \{1,\ldots,m\}$ for all $i \in \{1,\ldots,c\}$
- $I_1 \cup \cdots \cup I_c = \{1,\ldots,m\}$

(2) f の引数 $\{s_1,\ldots,s_m\}$ に対して，$f^\sigma(s_1,\ldots,s_m)$ は，f のステータス f^σ 中の整数 i を s_i で置き換えて得られる多重集合の列である．すなわち

$f^\sigma(s_1,\ldots,s_m) = (S_1,\ldots,S_c)$

ただし

$f^\sigma = (I_1,\ldots,I_c)$,

$S_j = \{s_{i_1},\ldots,s_{i_k}\}$

if $I_j = \{i_1,\ldots,i_k\}$ for all $j \in \{1,\ldots,c\}$.

例 3.1 項を $s = f(x,x,g(x))$ とし，f のステータスを $f^\sigma = (\{1,3\},\{2\})$ とする．このとき，

$$f^\sigma(x,x,g(x)) = (\{x,g(x)\},\{x\})$$

である．

拡張ステータスを用いると，従来のステータスもこの表記法により表すことができる．その例を以下に示す．

例 3.2 5引数の場合において，従来の *mult*, *left*, *right* ステータスは，それぞれ次のように書ける．

mult	:	$(\{1,2,3,4,5\})$
left	:	$(\{1\},\{2\},\{3\},\{4\},\{5\})$
right	:	$(\{5\},\{4\},\{3\},\{2\},\{1\})$

3.2 拡張ステータス付き再帰経路順序

定義 3.2（汎関数） 項の集合を $\mathcal{T}$ とするとき，汎関数 $\rhd$ は，$2^{(\mathcal{T}^2)}$ から $2^{(\mathcal{T}^2)}$ への写像である．

すなわち，$\rhd$ は $\mathcal{T}$ 上の二項関係 $R \subseteq \mathcal{T}^2$ を $\mathcal{T}$ 上の二項関係 $R' = \rhd(R) \subseteq \mathcal{T}^2$ に変換する．特に，我々が興味があるのは $\mathcal{T}$ 上の半順序 $\succ$ であり，$\rhd(\succ)$ のことを $\rhd\!\!\succ$ と書く．

定義 3.3 項を $s=f(s_1,\ldots,s_m)$, $t=g(t_1,\ldots,t_n)$ とする．このとき $s \succ t$ であることは以下の条件のいずれかを満たすときに限る．

(1) $s_i \succeq t$ for some $i \in \{1,\ldots,m\}$

(2) $f > g \quad \wedge \quad s \succ t_j$ for all $j \in \{1,\ldots,n\}$

(3) $f = g \quad \wedge \quad s \rhd\!\!\succ t \quad \wedge$

$\qquad s \succ t_j$ for all $j \in \{1,\ldots,n\}$

定義 2.4の RPOS は定義 3.3において $\rhd$ を f のステータスに依存して適切に定義することによって得られる．

補題 3.1（Kamin & Lévy[7]） 汎関数 $\rhd$ が以下の $(a) \sim (c)$ を満たすならば，$\succ$ は単純化順序である．

(a) $\succ$ が半順序ならば $\rhd\!\!\succ$ も半順序である．

(b) $\rhd$ は連続的である．

(c) $s \succ t \quad \Rightarrow \quad f(\ldots,s,\ldots) \rhd\!\!\succ f(\ldots,t,\ldots)$

なお，$\rhd$ が「連続的である」とは，$s \rhd\!\!\succ t$ ならば $\succ$ の「有限」部分集合 $\succ_f$ が存在して $s \rhd\!\!\succ_f t$ であることを意味する．すなわち，$s \rhd\!\!\succ t$ であることをチェックするには，有限個の対 $s' \succ t'$ のチェックで十分であることを意味する．これは通常，非常に弱い条件であり，ほぼ自明に成り立つものである．

定義 3.4（RPOES） 拡張ステータス付き再帰経路順序 RPOES（RPO with extended status）は，定義 3.3において $\rhd$ を以下のように定義したものである．

$s = f(s_1,\ldots,s_m) \rhd\!\!\succ g(t_1,\ldots,t_n) = t \qquad \Leftrightarrow$

$f = g \quad \wedge \quad S_1 = T_1,\ldots,S_{i-1} = T_{i-1}, S_i \succ\!\!\succ T_i$

$\qquad$ for some $i \in \{1,\ldots,c\}$

ただし

$f^\sigma(s_1,\ldots,s_m) = (S_1,\ldots,S_c)$,

$f^\sigma(t_1,\ldots,t_n) = (T_1,\ldots,T_c)$

定理 3.1 RPOES は単純化順序である．

証明 我々の定義した $\rhd$ が補題 3.1の条件 $(a) \sim (c)$ を満たすことを示せばよい．

(a) $\succ$ が半順序であると仮定し，定理 2.2より $\succ\!\!\succ$ は半順序であることに注意して，$\rhd\!\!\succ$ の推移性と非反射性を示す．

推移性：$s \rhd\!\!\succ t$, $t \rhd\!\!\succ u$ ならば $s \rhd\!\!\succ u$ であることを示せばよい．

$s = f(s_1,\ldots,s_m)$, $t = g(t_1,\ldots,t_n)$, $u = h(u_1,\ldots,u_p)$, $f^\sigma(s_1,\ldots,s_m) = (S_1,\ldots,S_c)$, $g^\sigma(t_1,\ldots,t_n) = (T_1,\ldots,T_d)$, $h^\sigma(u_1,\ldots,u_p) = (U_1,\ldots,U_e)$ とする．$s \rhd\!\!\succ t$ より $f = g$（よって $m = n$, $f^\sigma = g^\sigma$, $c = d$）であり，ある $i \in \{1,\ldots,c\}$ に対し，$S_1 = T_1,\ldots,S_{i-1} = T_{i-1}, S_i \succ\!\!\succ T_i$．同様に，$t \rhd\!\!\succ u$ より $g = h$（よって $n = p$, $g^\sigma = h^\sigma$, $d = e$）で，ある $j \in \{1,\ldots,c\}$ に対し，$T_1 = U_1,\ldots,T_{j-1} = U_{j-1}, T_j \succ\!\!\succ U_j$．よって，$k = \min(i,j)$ とおくと，$\succ\!\!\succ$ の推移性より，$f = h, S_1 = U_1,\ldots,S_{k-1} = U_{k-1}, S_k \succ\!\!\succ U_k$．よって，$s \rhd\!\!\succ u$.

非反射性：$s \rhd\!\!\succ s$ と仮定すると，ある i に対し，$S_1 = S_1,\ldots,S_{i-1} = S_{i-1}, S_i \succ\!\!\succ S_i$．これは $\succ\!\!\succ$ の非反射性に矛盾する．

(b) $s \rhd\!\!\succ t$ のチェックは高々 s と t の真部分項ど

うしの有限回の比較で十分である．すなわち，$\succ_f = \{(s',t') | s',t'$ はそれぞれ s，t の真部分項．$s' \succ t'\} \subseteq \succ$ なる有限の $\succ_f$ が存在する．

(c) $f(\ldots,s,\ldots)$ および $f(\ldots,t,\ldots)$ 中の s，t は第 k 番目の引数であるとし，$f^\sigma = (I_1,\ldots,I_c)$ において k は I_i $(i \in \{1,\ldots,c\})$ ではじめて現れるものとする．($I_1,\ldots,I_c$ の被覆条件よりそのような i の存在が保証される．) すなわち，$k \notin I_1 \cup \cdots \cup I_{i-1}, k \in I_i$．$f^\sigma(\ldots,s,\ldots) = (S_1,\ldots,S_c)$，$f^\sigma(\ldots,t,\ldots) = (T_1,\ldots,T_c)$ とおくと，$s \succ t$ ならば $S_1 = T_1,\ldots,S_{i-1} = T_{i-1}, S_i \succ\!\!\succ T_i$．(ここで，$S_i \succ\!\!\succ T_i$ であるのは，$T_i = (S_i - \{s\}) \cup \{t\}, s \succ t$ より．) よって，$f(\ldots,s,\ldots) \rhd\!\!\succ f(\ldots,t,\ldots)$．

□

この証明からわかるように，被覆条件が満たされないとき，$\succ$ は単純化順序にはならない．例えば，$f^\sigma = (\{1\})$ とすると，$1 \succ 0$ であっても $f(0,1) \not\succ f(0,0)$ であり，置換の性質が成り立たない．

4. 例　　題

RPOS では検証不可能だが，今回提案した RPOES で検証可能である例を以下に示す．

$$\mathcal{R} = \begin{cases} f(0,0,2) & \to & f(1,0,0) \\ f(1,2,0) & \to & f(0,1,2) \\ f(2,1,0) & \to & f(0,2,1) \end{cases}$$

この系においては，優先順位と従来のステータスをどのように選んでも停止性を示すことができない．しかしながら，拡張ステータスを $(\{1,2,3\},\{1\})$ とし，優先順位を $2 > 1 > 0$ とすることによって，いずれの書換え規則においても，左辺が右辺よりも大きくなり停止性を示すことができる．

$$\mathcal{R} =$$
$$\begin{cases} f(cons(x,y),z,w) & \to & f(z,y,cons(x,w)) \\ f(nil,cons(x,y),z) & \to & f(nil,y,cons(x,z)) \end{cases}$$

この系は，f の最初の二つの引数に与えられたリストの要素を交互に第3引数のリストの先頭に移動させるもので，拡張ステータスを $(\{1,2\},\{3\})$ とすることによって停止性を示すことができる．

5. おわりに

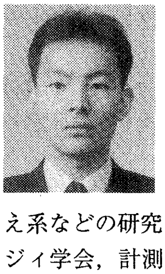
本論文では，従来のステータスを拡張することによって新しいステータスを提案した．また，拡張ステータスを適用した経路順序として RPOES を提案し，それが単純化順序であることを証明し，その有効性について述べた．明らかに，RPOES は RPOS を包含しており，より強力である．

今後の課題としては，優先順位や拡張ステータスを自動的に決定[8]すること，RPOS よりも強力な順序 (KNSS, RDOS など)[9]に拡張ステータスを適用し，より強力な拡張ステータス付き経路順序を提案することなどが挙げられる．

参　考　文　献

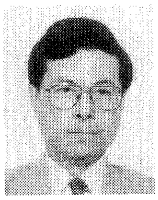
1) Dershowitz, N.: Orderings for Term-rewriting Systems, *Theoretical Computer Science*, Vol.17, pp.279–301 (1982).
2) Dershowitz, N.: Termination of Rewriting, *J. Symbolic Computation*, Vol.3, pp.69–116 (1987).
3) Dershowitz, N. and Jouannaud, J.-P.: 書換え系，コンピュータ基礎理論ハンドブック II, pp.243–321, 丸善 (1994).
4) Dershowitz, N. and Manna, Z.: Proving Termination with Multiset Orderings, *Comm. ACM*, Vol.22, No.8, pp.465–476 (1979).
5) 二木厚吉，外山芳人：項書き換え型計算モデルとその応用，情報処理，Vol.24, No.2, pp.133–146 (1983).
6) 井田哲雄：計算モデルの基礎理論，pp.223–296, 岩波書店 (1991).
7) Kamin, S. and Lévy, J.-J.: Attempts for Generalizing the Recursive Path Orderings, Report, Dept. of Computer Science, Univ. of Illinois (1980).
8) 近藤　久，栗原正仁，大内　東：Reason Maintenance System による項書き換えシステム停止性検証の効率化，情報処理学会論文誌，Vol.32, No.9, pp.1046–1056 (1991).
9) Steinbach, J.: Simplification Orderings: Putting Them to the Test, *Journal of Automated Reasoning*, Vol.10, pp.389–397 (1993).

(平成7年　8月14日受付)
(平成7年10月　5日採録)



能登　正人（学生会員）

1967年生．1991年北海学園大学工学部電子情報工学科卒業．1993年北海道大学大学院工学研究科情報工学専攻修士課程修了．現在，同博士課程在学中．システム工学，項書換え系などの研究に従事．電子情報通信学会，日本ファジィ学会，計測自動制御学会各会員．



栗原　正仁（正会員）

1955年生．1978年北海道大学工学部電気工学科卒業．1980年同大学院工学研究科情報工学専攻修士課程修了．現在，北海道大学工学部システム情報工学専攻助教授．工学博士．人工知能，項書換え系などの研究に従事．電子情報通信学会，日本ソフトウェア科学会，人工知能学会，EATCS（欧州理論計算機科学会）各会員．

大内　　東（正会員）

1945年生．1974年北海道大学大学院工学研究科博士課程修了．現在，北海道大学工学部システム情報工学専攻教授．工学博士．システム情報工学，応用人工知能システム，医療システムなどの研究に従事．電子情報通信学会，日本ファジィ学会，計測自動制御学会，人工知能学会，電気学会，日本OR学会，医療情報学会，病院管理学会，IEEE-SMC各会員．