

無限項書き換えシステムにおける性質に関する考察*

島根大学 総合理工学部 岩見 宗弘 (Munehiro Iwami)

Interdisciplinary Faculty of Science and Engineering, Shimane University

東北大学 電気通信研究所 青戸 等人 (Takahito Aoto)

RIEC, Tohoku University

概要

遅延リストやストリームといった仮想的に無限長とみなされるデータを扱う関数型プログラムの計算モデルとして、無限項書き換えシステムが提案されている。項書き換えシステムにおける停止性に対応する基本的な性質として、無限項書き換えシステムにおける強頭部正規化可能性があり、その証明法が Zantema (2008) や Endrullis ら (2009) によって報告されている。本論文では、強頭部正規化可能性に対する反証手続きを提案する。提案する手続きの基本的なアイデアは、有限表現を持つ無限項である正則項の反例を構成する点にある。反証手続きの正しさを示すとともに、手続きの実装を報告する。実験の結果、自動反証法が従来知られていない例について自動反証に成功することを確認した。

1 はじめに

遅延評価に基づくデータ構造やストリームといったデータ構造を用いたプログラムでは、データを仮想的に無限長と見做すことによって見通しのよい議論が可能になる場合が少なくない。このような無限長のデータを扱う計算モデルとして、無限項書き換えシステムが提案されている [4, 9, 10, 11, 14, 15, 16]。無限項書き換えシステムは、関数型プログラムの計算モデルとして知られている項書き換えシステム [1, 6, 14] の拡張である。通常の項書き換えシステムとの違いは、通常の項書き換えシステムが有限の大きさの項しか対象にしないのとは対照的に、無限項書き換えシステムでは大きさが無限であるような項 (無限項) も対象とする点である。

項書き換えシステムに基づく様々な検証法において停止性は非常に有用な性質として知られており、様々な停止性の証明法や反証法が提案されている [1, 6, 14]。一方、通常の項書き換えシステムとは対照的に、無限項書き換えシステムでは停止性を考える意味があまりない。

このため、停止性に替わる無限項書き換えシステムにおける基本的な性質として、強頭部正規化可能性¹が考えられている [4, 11, 15]。強頭部正規化可能性は、項のどの固定位置も有限回しか書き換えが可能でないという性質であり、これによって、ストリームのプレフィックスのように観測可能な有限部分のデータについては、書き換えに対して安定な部分が求まることが保証される。

無限項書き換えシステムが左線形 (どの書き換え規則の左辺にも同じ変数は 2 回以上出現しない) の場合、強頭部正規化可能性は ω -強頭部正規化可能性と同等であることが知られている [10, 14]。 ω -強頭部正規化可能性は、長さ ω (最小の極限順序数) の書き換え列に限定して強頭部正規化可能性を保証する性質である。

非 ω -強頭部正規化可能性については、縮退規則 (右辺が変数となっている書き換え規則) を持つ無限項書き換えシステムが ω -強頭部正規化可能性を持たないことが知られているが、それ以外の十分条件や証明法は従来あまり知られていない。その一方で、よく知られた組合せ子の書き換え規則の多くは、非縮退であるにもかかわらず、 ω -強頭部正規化可能性を持たないことが岩見 [7] によって報告されている。このため、縮退規則を持たない無限項書き換えシステムに対しても適用可能であるような ω -強頭部正規化可能性の自動反証法は、より強力な ω -強頭部正規化可能性判定法を実現するためには不可欠と考えられる。

*This paper is an extended abstract and the detailed version will be published elsewhere.

¹無限停止性あるいは強収束性とよばれることもある。

本論文では、縮退規則を持たない無限項書き換えシステムに対しても適用可能であるような ω -強頭部正規化可能性の反証手続きを提案する。この反証手続きの正しさ（健全性）を示すとともに、手続きの実装および実験を報告する。提案する手続きの基本的なアイデアは、有限表現を持つ無限項である正則項の反例を構成する点にある。

2 準備

本節では、本論文で使用する無限項書き換えシステムに関する定義や記法を与える。詳細は文献 [4, 11, 14] 等を参考にして頂きたい。

関数記号の集合を \mathcal{F} 、変数の可算無限集合を \mathcal{V} と記す ($\mathcal{F} \cap \mathcal{V} = \emptyset$ とする)。0 引数の関数記号を定数とよび、 n 引数の関数記号集合を \mathcal{F}_n と記す。 \mathbb{N}^+ を正整数集合とし、正整数の有限列を \mathbb{N}^{+*} と記す。有限列 $p \in \mathbb{N}^{+*}$ の長さを $|p|$ 、有限列 $p, q \in \mathbb{N}^{+*}$ の連結を $p.q$ と記す。部分関数 $t: \mathbb{N}^{+*} \rightarrow \mathcal{F} \cup \mathcal{V}$ のうち、以下の条件を満たすものを \mathcal{F}, \mathcal{V} 上の項とよぶ: (1) $t(\epsilon) \in \mathcal{F} \cup \mathcal{V}$; (2) 任意の $p \in \mathbb{N}^{+*}$ について、 $t(pi) \in \mathcal{F} \cup \mathcal{V} \iff t(p) \in \mathcal{F}_n$ かつ $1 \leq i \leq n$ 。ここで、 ϵ は空列を表す。 \mathcal{F}, \mathcal{V} 上の項の集合を $T_{inf}(\mathcal{F}, \mathcal{V})$ と表す。項 $t \in T_{inf}(\mathcal{F}, \mathcal{V})$ の定義域 $\text{Pos}(t) = \{p \in \mathbb{N}^{+*} \mid t(p) \in \mathcal{F} \cup \mathcal{V}\}$ の要素を t における位置とよぶ。特に、位置 ϵ を根位置とよぶ。

関数 $\sigma: \mathcal{V} \rightarrow T_{inf}(\mathcal{F}, \mathcal{V})$ を代入とよぶ。代入 σ の定義域 $\{x \in \mathcal{V} \mid \sigma(x) \neq x\}$ を $\text{dom}(\sigma)$ と記す。 $\text{dom}(\sigma) = \{x_1, \dots, x_n\}$ 、 $\sigma(x_i) = u_i$ なる代入を $\{x_1 := u_1, \dots, x_n := u_n\}$ とも記す。

$\square \notin \mathcal{F} \cup \mathcal{V}$ なる定数 \square (ホールとよぶ) を考える。 $T_{inf}(\mathcal{F} \cup \{\square\}, \mathcal{V})$ の要素 C のうち、 $\{p \in \mathbb{N}^{+*} \mid C(p) = \square\}$ が有限集合となるものを文脈とよぶ。

等式を $s \approx t$ と記す。ただし、ここで $s, t \in T_{inf}(\mathcal{F}, \mathcal{V})$ とする。等式 $s \approx t$ の左辺は s 、右辺は t をさし、等式の右辺と左辺を区別する。代入 σ を等式集合 $\{x \approx \sigma(x) \mid x \in \text{dom}(\sigma)\}$ と同一視することがある。等式 $l \approx r$ が以下の条件を満たすとき、これを書き換え規則とよび、 $l \rightarrow r$ と記す: (1) l, r は有限項²、(2) $l(\epsilon) \notin \mathcal{V}$ 、(3) $\mathcal{V}(r) \subseteq \mathcal{V}(l)$ 。書き換え規則の集合 \mathcal{R} を無限項書き換えシステムとよぶ。ある書き換え規則の右辺が変数であるとき \mathcal{R} は縮退性を持つという。

項 $s, t \in T_{inf}(\mathcal{F}, \mathcal{V})$ に対して、書き換え規則 $l \rightarrow r \in \mathcal{R}$ 、代入 σ 、文脈 $C[\]_p$ が存在して、 $s = C[l\sigma]_p$ かつ $t = C[r\sigma]_p$ となるとき、 $s \rightarrow_{p, \mathcal{R}} t$ と記す。これを位置 p における項 s から項 t への簡約または書き換えステップとよぶ。

等式集合 E の単一化子とは、任意の $s \approx t \in E$ について $s\sigma = t\sigma$ となる代入 σ をいう。等式集合 E の単一化子の集合を $\text{Unif}_{inf}(E)$ と記す。 $\text{Unif}_{inf}(E) \neq \emptyset$ であるとき、等式集合 E は単一化可能であるという。等式集合 $\{s \approx t\}$ が単一化可能であるとき、 s と t が単一化可能であるという。単一化可能性を判定する問題を、単一化問題という。等式集合 E の最汎単一化子を $\text{mgu}_{inf}(E)$ と記す。特に、 $E = \{s \approx t\}$ のとき、 $\text{mgu}_{inf}(E)$ を $\text{mgu}_{inf}(s, t)$ と記す。

項 $s \in T_{inf}(\mathcal{F}, \mathcal{V})$ に対して、ある書き換え規則 $l \rightarrow r \in \mathcal{R}$ について s の非変数部分項 $s|_p$ と l が単一化可能であるとする。 $\theta = \text{mgu}_{inf}(s|_p, l)$ 、 $t = s[r]_p\theta$ とおくと、 $s \rightsquigarrow_{p, \theta, \mathcal{R}} t$ と記す。これを位置 p における項 s から項 t へのナローイングステップとよぶ。

3 正則項の有限表現に基づく単一化手続き

定義 1 (正則項 [3]) 項 t が正則であるとは、 t の部分項集合が有限集合であるときをいう。

命題 2 (正則項の単一化 [3]) 正則項の無限項上での単一化問題は決定可能であり、単一化可能であるときに最汎単一化子を求めるアルゴリズムが存在する。また、最汎単一化子は正則代入となる。

²書き換え規則の左辺を有限項に制限するのは一般的である [4, 14]。本論文では文献 [4] と同様に右辺も有限項に制限する。

有限代入 $\{x_1 := u_1, \dots, x_n := u_n\}$ が以下の条件を満たすとき、これを再帰式表現とよび、 $[x_1 := u_1, \dots, x_n := u_n]$ と記す:

$$\neg \exists i_1, \dots, i_k \in \{1, \dots, n\}. (\forall 1 \leq j < k. u_{i_j} = x_{i_{j+1}}) \wedge u_{i_k} = x_{i_1} \quad (1)$$

再帰式表現を用いて、正則項を表現することが出来る。今、 $\theta = [x_1 := t_1, \dots, x_n := t_n]$ とおき、任意の $1 \leq i \leq n$ について、 $t_i = C_i[x_{i_1}, \dots, x_{i_{k_i}}]_{p_{i_1}, \dots, p_{i_{k_i}}}$ とする。ただし、ここで、 $x_{i_1}, \dots, x_{i_{k_i}}$ は変数 x_1, \dots, x_n のうち t_i 中に出現するもの全てであるとする。このとき、項 $\theta^*(x_1), \dots, \theta^*(x_n)$ を以下のように定義する。

$$\theta^*(x_i)(p) = \begin{cases} C_i(p) & (p \in \text{Pos}(C_i) \wedge C_i|_p \neq \square \text{ の場合}) \\ \theta^*(x_{i_j})(q) & (\exists j, q. p = p_{i_j}.q \text{ の場合}) \end{cases}$$

項 $\theta^*(x_0), \dots, \theta^*(x_n)$ を再帰式表現の解とよぶ。

命題 3 (正則項と再帰式表現 [2, 3]) 再帰式表現の解は正則項である。また、任意の正則項はある再帰式表現の解となる。

Jaffer [8] は、文献 [12, 13] の手続きを拡張して、有限項の等式の有限集合 $T = \{s_1 \approx t_1, \dots, s_n \approx t_n\}$ が与えられたときに、無限項上の T の単一化問題を解く効率の良い手続きを与えている。Jaffer の手続き ($\text{unif-proc}_j(T)$ と記す) に基づいて、再帰式表現で与えられた正則項の単一化手続きが実現できる。

定理 4 (再帰式表現に基づく単一化手続き) $\theta = [x_1 := s_1, \dots, x_m := s_m]$, $\gamma = [y_1 := t_1, \dots, y_n := t_n]$ を、 $\{x_1, \dots, x_m\} \cap \{y_1, \dots, y_n\} = \emptyset$ なる再帰式表現とし、 $s = \theta^*(x_1)$, $t = \gamma^*(y_1)$ とおく。このとき、 s と t の単一化問題は $\text{unif-proc}_j(\gamma \cup \theta \cup \{x_1 \approx y_1\})$ によって解ける。

4 ω -強頭部正規化可能性の反証手続き

定義 5 (ω -強頭部正規化可能性) \mathcal{R} を無限項書き換えシステムとする。任意の無限書き換え列 $t_0 \rightarrow_{p_0, \mathcal{R}} t_1 \rightarrow_{p_1, \mathcal{R}} t_2 \rightarrow_{p_2, \mathcal{R}} \dots$ について、あるインデックス $n_0 \in \omega$ が存在して任意の n ($n_0 \leq n < \omega$) について $p_n \neq \epsilon$ となるとき、 \mathcal{R} は ω -強頭部正規化可能性を持つという。無限項書き換えシステム \mathcal{R} が ω -強頭部正規化可能性を持つことを $\text{SHN}^\omega(\mathcal{R})$ と記す。

Zanema [15] や Endrullis ら [4] によって、無限項書き換えシステムが ω -強頭部正規化可能性を持つことを示す手法が知られている。一方、無限項書き換えシステムが ω -強頭部正規化可能性を持たないための十分条件として以下の条件が知られている [15]。

命題 6 (縮退システムの ω -強頭部正規化可能性) 無限項書き換えシステム \mathcal{R} が縮退性を持つとき、 ω -強頭部正規化可能性を持たない。

縮退性を持たない無限項書き換えシステム \mathcal{R} についても、根書き換えが無限回出現するような無限書き換え列を構成することによって、 \mathcal{R} が ω -強頭部正規化可能性を持たないことを示せる場合がある。

補題 7 (非 ω -強頭部正規化可能性の十分条件) \mathcal{R} を無限項書き換えシステムとする。ある書き換え規則 $l \rightarrow r \in \mathcal{R}$ とある代入 σ, η が存在して、 $r\sigma \xrightarrow{*}_{\mathcal{R}} l\eta$ となるとき、 \mathcal{R} は ω -強頭部正規化可能性を持たない。

定義 8 (有限代入上の関係 \triangleright) $\theta = \{x_1 := u_1, \dots, x_n := u_n\}$ を有限代入、 $X = \bigcup_{1 \leq i \leq n} \mathcal{V}(u_i)$, $Y = X \cup \{x_1, \dots, x_n\}$ とおく。ある代入 $\sigma: X \rightarrow Y$ について、 $\theta' = \sigma \circ \theta$ となり、 θ' が再帰式表現であるとき、 $\theta' \triangleright \theta$ と記す。

手続き 9 (ω -強頭部正規化可能性反証手続き) 表 1 に SML 風のシンタックスをもつ関数型プログラムとして ω -強頭部正規化可能性の反証手続き $\text{disprove-}\omega\text{-SHN}$ を与える。

定理 10 (SHN^ω 反証手続きの健全性) 無限項書き換えシステム \mathcal{R} に対して、手続き 9 の ω -強頭部正規化可能性の反証手続きが成功するとき、 \mathcal{R} は ω -強頭部正規化可能性を持たない。

表 1: ω -強頭部正規化可能性の反証手続き

```

1: fun disprove-omega-shn ( $\mathcal{R}$ ) =
2:   let fun ccheck ( $l, t, \sigma$ ) =
3:     if  $\exists \theta. \theta = \text{mgu}_{\text{inf}}(\text{REN}(l\sigma), t), \exists \delta \triangleright \theta. \text{match}_{\text{inf}}(l\sigma\delta^*, t\delta^*)$ 
4:     then SOME  $l\sigma\delta^*$  else NONE
5:   fun narrow ( $l, t, \sigma$ ) = [( $l, t', \rho \circ \sigma$ ) |  $t \rightsquigarrow_{\rho} t'$ ]
6:   fun step [] = FAIL
7:     | step (( $l, t, \sigma$ ):: $xs$ ) = case check ( $l, t, \sigma$ ) of
8:       SOME  $t \Rightarrow \text{SUCCESS } t$  | NONE  $\Rightarrow \text{step } (xs @ \text{narrow } (l, t, \sigma))$ 
9:   in step [( $l, r, \theta$ ) |  $l \rightarrow r \in \mathcal{R}$ ] end

```

5 実装および実験

4節で与えた ω -強頭部正規化可能性の反証手続き(手続き9)をSML/NJを用いて実装した。

この実装を用いて行った反証手続きの実験について報告する。実験には、1.2GHz Intel Pentium プロセッサのCPU、1GBのメモリを持つPCを用いた。step関数の適用を1ステップとカウントし、ステップ数の最大を20ステップとした。この最大値を越えた場合は、手続きは停止しない(“発散”)と見做した。

岩見[7]は組合せ子の書き換え規則に対して、 ω -強頭部正規化可能性が成立しないことを、反例を構成することによって報告している。文献[7]で示された組合せ子の書き換え規則の例に対して提案手法を適用した結果、 ω -強頭部正規化可能性が成立しない25例に対して、提案手法に基づく ω -強頭部正規化可能性の自動反証に成功した。また、 ω -強頭部正規化可能性が従来検証されていなかった組合せ子 F^* ([5])についても自動反証に成功した。

6 おわりに

本論文では、無限項書き換えシステムにおける ω -強頭部正規化可能性の反証手続きを与えるとともに、これらの手続きの正しさ(健全性)を証明した。また、提案手法を実装し、反証実験を行った。

ω -強頭部正規化可能性の反証法は、簡単な十分条件を除いて従来知られていなかったが、我々の手続きに基づいて、その十分条件を満たさない例についても自動反証が可能となる場合があることを確認した。

謝辞

本研究に対してアドバイスやコメントをいただきました外山芳人教授に感謝致します。本研究は日本学術振興会科学研究費 20500002, 21700017 の補助を一部受けて行われた。

参考文献

- [1] Baader, F. and Nipkow, T.: *Term Rewriting and All That*, Cambridge University Press, 1998.
- [2] Colmerauer, A.: Equations and inequations on finite and infinite trees, *Proc. of FGCS 1984*, 1984, pp. 85-99.
- [3] Courcelle, B.: Fundamental properties of infinite trees, *Theoretical Computer Science*, Vol. 25(1983), pp. 95-169.

- [4] Endrullis, J., Grabmayer, C., Hendriks, D., Klop, J. W., and de Vrijer, R.: Proving infinitary normalization, *Proc. of TYPES 2008*, LNCS, Vol. 5497, Springer-Verlag, 2009, pp. 64–82.
- [5] Fujita, K.-E.: An interpretation of $\lambda\mu$ -calculus in λ -calculus, *Information Processing Letters*, Vol. 84, No. 5(2002), pp. 261–264.
- [6] Huet, G. and Oppen, D. C.: Equations and rewrite rules: A survey, Technical report, Stanford University, Stanford, CA, USA, 1980.
- [7] 岩見宗弘: 組合せ子の強収束性, *FIT2009 第1分冊*, 2009, pp. 251–258. A-011.
- [8] Jaffar, J.: Efficient unification over infinite trees, *New Generation Computing*, Vol. 2(1984), pp. 207–219.
- [9] Kennaway, J. R., Klop, J. W., Sleep, M. R., and de Vries, F. J.: On the adequacy of graph rewriting for simulating term rewriting, *ACM Transactions on Programming Languages and Systems*, Vol. 16, No. 3(1994), pp. 493–523.
- [10] Kennaway, J. R., Klop, J. W., Sleep, M. R., and de Vries, F. J.: Transfinite reductions in orthogonal term rewriting systems, *Information and Computation*, Vol. 119(1995), pp. 18–38.
- [11] Klop, J. W. and de Vrijer, R.: Infinitary normalization, *We Will Show Them! Essays in Honour of D. Gabbay*, Vol. 2, College Publications, 2005, pp. 169–192.
- [12] Martelli, A. and Montanari, U.: An efficient unification algorithm, *ACM Transactions on Programming Languages and Systems*, Vol. 4, No. 2(1982), pp. 239–257.
- [13] Mukai, K.: A unification algorithm for infinite trees, *Proc. of IJCAI 1983*, 1983, pp. 547–549.
- [14] Terese: *Term Rewriting Systems*, Cambridge University Press, 2003.
- [15] Zantema, H.: Normalization of infinite terms, *Proc. of RTA 2008*, LNCS, Vol. 5117, Springer-Verlag, 2008, pp. 441–455.
- [16] Zantema, H.: Well-definedness of streams by termination, *Proc. of RTA-2009*, LNCS, Vol. 5595, Springer-Verlag, 2009, pp. 164–178.