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.

強計算依存対法による高階書換え系の停止性証明

磯谷　泰巨†　　草刈　圭一朗††　　酒井　正彦††　　坂部　俊樹††　　西田　直樹††

† 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: †isogai@sakabe.i.is.nagoya-u.ac.jp, ††{kusakari,sakai,sakabe,nishida}@is.nagoya-u.ac.jp

あらまし　項書換え系は関数プログラムの計算モデルであり，その停止性は重要な性質の 1 つである．停止性証明法の 1 つに依存対法と呼ばれる再起構造解析法があるが，これを高階書換え系に適用するこれまでの手法は能力が低い．本論文では単純型項書換え系のための強計算依存対法の高階書換え系への拡張を提案する．これにより，既存の手法では証明できなかった高階書換え系の停止性の証明が可能となった．
キーワード　関数プログラム，高階書換え系，強計算性，依存対法，匿名関数

Proving Termination of Higher-Order Rewrite Systems based on Strongly Computable Dependency Pair Method

Yasuo ISOGAI†, Keiichirou KUSAKARI††, Masahiko SAKAI††,
Toshiki SAKABE††, and Naoki NISHIDA††

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

Abstract　The higher-order rewrite systems (HRS for short) [14] are a computation model of functional programming languages, and hence the termination is one of the most important properties of them. While the dependency pair method is very useful for proving termination of term rewriting systems, previous methods to apply it for HRSs do not have enough power. In this paper, we extend the strongly computable dependency-pairs method for HRSs. There are some HRSs of which termination can be proved by our method but by no known methods.
Key words　Functional Program, Higher-order Rewrite System, Strongly Computabilty, Dependecy pair Method, Anonymous Function.

1. はじめに

項書換え系（Term Rewriting Systems, TRS）は，項の書換えの繰り返しにより計算を表現する計算モデルであり，関数型言語の計算モデルや定理自動証明，代数的仕様記述などに利用されている [9]．しかし，TRS は関数プログラムで広く利用される高階関数を取り扱うことができない．例えば ML において，代表的な高階関数 map は次のように定義されている．

```
fun map F nil = nil
  |  map F (X::L) = (F X)::(map F L);
```

この関数は TRS 上で直接的に表現できない．そこで，高階関数の表現に適した λ 計算と TRS を融合した高階書換え系（Higher-Order Rewrite Systems, HRS）が提案された [5]．HRS では map 関数を次のように表すことができる．

$$R_1 = \begin{cases} map(\lambda x.F(x), nil) \ \rightarrow \ nil \\ map(\lambda x.F(x), cons(X, L)) \\ \qquad \rightarrow \ cons(F(X), map(\lambda x.F(x), L)) \end{cases}$$

HRS の性質の一つに停止性がある．停止性とは，計算がその過程によらず必ず停止して値を返すことを意味する．これはプログラムが暴走しないことに相当する，非常に重要な性質である．しかし，HRS の停止性は決定不可能であるため，停止性を保証する十分条件が研究されている．近年，TRS において依存対法と呼ばれる停止性自動証明法が提案された [1]．依存対法とは，関数呼び出しの依存関係から再帰構造を解析し，停止性を証明する手法である．この手法は HRS 上にも拡張されている [6] [7]．しかし, [6] の方法は証明の際に部分項性を満た

す順序を必要とし, [7] の方法では手法が適用できる HRS のクラスが狭いため，実用性にかける．

近年，単純型項書換え系（Simply-Typed Term Rewriting Systems, STRS）において，直接関数渡し（plain function-passing, PFP）と呼ばれるクラスに対して停止性を証明できる強計算依存対法が新たに提案された [4]．PFP とは，高階変数が左辺の引数から直接右辺に渡されるというものである．ほとんどの関数プログラムは PFP であるため，非常に実用性が高い．この手法の特徴は強計算性と呼ばれる概念を用いていることである．強計算性とは，適切な入力に対して計算がその過程によらず必ず停止して，適切な値を返すことを意味する．直感的には関数としての停止性に対応する概念で，関数に適切な値を入力するとその解が得られるという性質である．この概念は型付き λ 計算の停止性証明のために導入された．

本研究では，STRS 上で提案された強計算依存対法を HRS 上で再構築する．この際，2 つの問題が特に障害となる．1 つは代入によって変化する項の構造への対処が必要となることである．この問題は強計算依存対法の再構築を多少複雑にしたが，再構築することには成功した．もう 1 つの問題は STRS では記述できなかった匿名関数の対処である．例えば，ML においてリストの各要素を倍にする以下の関数 d_map

```
fun d_map L = map (fn x => x+x) L;
```

において `(fn x => x+x)` が匿名関数である．これに対して，高階変数が左辺の引数から直接右辺に渡される PFP の条件に，右辺に現れる高階型の項が自身の束縛変数を引数にのみ持つという条件を加えることにより対処する．HRS では，上記の d_map 関数は自然数上の加算を表す関数 add と R_1 とともに次のように表現される．

$$R_2 = R_1 \cup \left\{ \begin{array}{rcl} add(0,Y) & \rightarrow & Y \\ add(s(X),Y) & \rightarrow & s(add(X,Y)) \\ d\_map(L) & \rightarrow & map(\lambda x.add(x,x),L) \end{array} \right.$$

この HRS R_2 は PFP の条件を満たしている．本手法は，HRS R_2 より再帰の依存関係を取り出し，そこから導ける再帰成分を解析することで R_2 の停止性を証明できる．

文献 [6] で提案された HRS における依存対法は，適用できる HRS のクラスこそ広いものの，停止性を証明するには弱かった．また，文献 [7] で提案された依存対法は強力な証明手法ではあったが，適用する HRS が線形性や変数の入れ子に対する制限を必要とし，適用できる HRS のクラスが狭かった．本論文で提案する HRS 上の強計算依存対は, [6] よりも強力であり, [7] よりも広いクラスの HRS に対して適用が可能である．

2. 準　　備

本節では，文献 [5] [7] に基づいて論文中で必要となる HRS の諸概念について述べる．

基本型の集合を $\mathcal{B}(\neq \emptyset)$ とする．単純型の集合 $\mathcal{S}$ は，$\mathcal{S} \supseteq \mathcal{B}$ および $S \supseteq \{\alpha \rightarrow \beta \mid \alpha, \beta \in \mathcal{S}\}$ の関係を満たす最小の集合として定義する．以降，型を表すために α, β を用いる．また，基本型でない単純型を高階型と呼ぶ．

型 α の変数集合を $\mathcal{V}_\alpha$ とし，変数全体の集合を $\mathcal{V} = \bigcup_{\alpha \in \mathcal{S}} \mathcal{V}_\alpha$ とする．型 α の関数記号の集合を $\mathcal{F}_\alpha$ とし，関数記号全体の集合を $\mathcal{F} = \bigcup_{\alpha \in \mathcal{S}} \mathcal{F}_\alpha$ とする．このとき，$\mathcal{V} \cap \mathcal{F} = \emptyset$ であると仮定する．また $\alpha \neq \beta$ のとき $\mathcal{V}_\alpha \cap \mathcal{V}_\beta = \emptyset$ および $\mathcal{F}_\alpha \cap \mathcal{F}_\beta = \emptyset$ が成り立つとする．関数記号の表記には f を用いる．また，a は変数，関数記号のどちらにも用いる．

型 α の単純型準項（simply typed preterm）の集合を以下の条件を満たす最小の集合として定義する．

- $a \in \mathcal{V}_\alpha \cup \mathcal{F}_\alpha$ のとき，a は型 α の単純型準項.
- s が型 $\beta \rightarrow \alpha$ の単純型準項であり，t が型 β の単純型準項であるとき，$(s\,t)$ は型 α の単純型準項.
- $x \in \mathcal{V}_\alpha$ かつ t が型 β の単純型準項であるとき，$(\lambda x.t)$ は型 $\alpha \rightarrow \beta$ の単純型準項.

以降，単純型準項を単に準項と呼ぶ．型 α の準項 t に対し，その型を明示する際は $\tau(t) = \alpha$ と表記する．準項を定義通りに表記すると，括弧が多く見にくいため，次の省略記法を用いる．

(1)　$\lambda x_1 x_2 \ldots x_m.t \equiv (\lambda x_1.(\lambda x_2.(\cdots(\lambda x_m.t)\cdots)))$.

(2)　$(t_0\,t_1\,t_2\,\cdots\,t_n) \equiv ((\cdots((t_0\,t_1)\,t_2)\cdots)\,t_n)$.

(1) において $m = 0$ のときは単に t とし，(2) においては $n = 0$ のとき単に t_0 として扱う．また準項 t における束縛変数と自由変数それぞれの集合を $BV(t), FV(t)$ で表す．$Var(t) = BV(t) \cup FV(t)$ とする．以降，束縛変数の表記には x, y を用い，自由変数の表記には F, L, X, Y を用いる．準項 t に対し，t の束縛変数を新しい変数に置き換えた準項 s は t と同一視できる．このとき，t と s は α 同値であるといい，$t \equiv s$ と記す．本論文では簡便のため，すべての束縛変数は異なる名前を持つとする．

文脈とは，$C[,\ldots,]_{\alpha_1,\ldots,\alpha_n}$ と表されるホールと呼ばれる型が α_i $(1 \leqq i \leqq n)$ の特殊な関数記号 $\Box_{\alpha_i}$ を含むような準項である．添え字 $\alpha_1, \ldots, \alpha_n$ は文脈に含まれるホールの型が左から順に $\alpha_1, \ldots, \alpha_n$ であることを示す．また，$C[,\ldots,]_{\alpha_1,\ldots,\alpha_n}$ に含まれる $\Box_{\alpha_1}, \ldots, \Box_{\alpha_n}$ をそれぞれ t_i $(\tau(t_i) = \alpha_i)$ で置き換えた準項を $C[t_1,\ldots,t_n]_{\alpha_1,\ldots,\alpha_n}$ と表す．混乱を招かない場合，型を省略して $C[,\ldots,]$ と記す．

代入 σ は，変数から準項への写像であり $\sigma(X)$ と表す．$\sigma(X)$ の型は X と同じとする．ここで，$Dom(\sigma) = \{X \mid X \not\equiv \sigma(X)\}$ および $Var(\sigma) = \bigcup_{X \in Dom(\sigma)} Var(\sigma(X))$ と定義する．$Dom(\sigma) = \{X_1, \ldots X_n\}$ かつ $\forall i.[\sigma(X_i) \equiv t_i]$ であるような代入 σ の表記を $\{X_1 \mapsto t_1, \ldots, X_n \mapsto t_n\}$ とする．すべての σ に対して，準項から準項への写像 $\tilde{\sigma}$ を以下のように定義する．

$$\tilde{\sigma}(X) \equiv \sigma(X), \quad \tilde{\sigma}(c) \equiv c, \quad \tilde{\sigma}(s\,t) \equiv (\tilde{\sigma}(s)\,\tilde{\sigma}(t)),$$
$$\tilde{\sigma}(\lambda x.t) \equiv \lambda y.(\tilde{\sigma}(t')) \quad \text{if} \quad \lambda x.t \equiv \lambda y.t' \ \text{ and }$$
$$y \notin Dom(\sigma) \cup Var(\sigma)$$

簡便のため σ と $\tilde{\sigma}$ を同一視し，$\sigma(t)$ は簡単に $t\sigma$ と表記する．

λ 計算より β 変換および η 拡大の概念を取り入れる．β 変換の基となる $((\lambda x.t)\,u)$ の形の準項を β 基（β-redex）と呼び，β 基を含まない準項を β 正規形と呼ぶ．η 拡大とは，高階型の準項に対して引数を補う操作である．η 拡大は $C[t] \underset{\eta}{\rightarrow}$

$C[\lambda x.(tx)]$ と定義される．ただし，以下の条件をすべて満たすものとする．

- t は高階型．
- $x \notin FV(t)$.
- 変換によって新たな β 基を生成しない．

η 拡大できない準項を η ロング 正規形と呼ぶ．また，η ロング 正規形かつ β 正規形な準項を η ロング β 正規形と呼び，準項 t の η ロング β 正規形を $t\downarrow$ で表す．すべての準項にはただ 1 つの η ロング β 正規形が存在することが知られている．型 α である準項の η ロング β 正規形を型 α の項と呼ぶ．型 α の項の集合 $\mathcal{T}_\alpha$ は $\{t\downarrow \mid t$ は型αの準項 $\}$ で定義する．また，項全体の集合を $\mathcal{T} = \bigcup_{\alpha\in\mathcal{S}} \mathcal{T}_\alpha$ とする．項は，一般的に $t \equiv \lambda x_1 \ldots x_m.(a\, t_1 \cdots t_n)$ と表せるが，通例に従って $t \equiv \lambda x_1 \ldots x_m.a(t_1, \ldots, t_n)$ と表記する．このとき，項 t の先頭記号を $top(t) = a$，引数集合を $args(t) = \{t_1, \ldots, t_n\}$ と定義する．また，t の部分項の集合 $Sub(t)$ を $t \equiv \lambda x.s$ ならば $\{t\} \cup Sub(s)$，$t \equiv a(t_1, \ldots, t_n)$ ならば $\{t\} \cup \bigcup_{i=1}^n Sub(t_i)$ と定義する．また $s \in Sub(t)$ のとき，$t \trianglerighteq_{sub} s$ と書く．さらに項 t の位置の集合 $Pos(t)$ を $t \equiv \lambda x.s$ ならば $\{\varepsilon\} \cup \{1p \mid p \in Pos(s)\}$，$t \equiv a(t_1, \ldots, t_n)$ ならば $\{\varepsilon\} \cup \bigcup_{i=1}^n \{ip \mid p \in Pos(t_i)\}$ と定義する．ここで，位置上の順序 $\prec$ を $p \prec q \stackrel{\text{def}}{\Longleftrightarrow} \exists w \neq \varepsilon.[pw = q]$ で定義する．また，項 t の位置 p における部分項を $t|_p$ で表す．以降，項の表記には l, r, s, t, u, v を用いる．

l, r を型 α の項とする．以下の条件を満たす $l \to r$ を型 α の高階書換え規則と呼ぶ．

- $top(l) \in \mathcal{F}$,
- $\alpha \in \mathcal{B}$,
- $FV(l) \supseteq FV(r)$.

また，高階書換え規則の集合を高階書換え系 (HRS) と呼ぶ．項 s が項 t に書き換えられるとは，ある文脈 $C[]$ と代入 θ と書換え規則 $l \to r \in R$ が存在し，$s \equiv C[l\theta\downarrow]$ かつ $t \equiv C[r\theta\downarrow]$ を満たすことであり，$s \to_R t$ と記す．$\to_R$ を書換え関係と呼び，その反射推移閉包を $\xrightarrow{*}_R$ で表す．混乱を招かない場合，添え字 R は省略する．

t から始まる無限列 $t \equiv t_0 \to t_1 \to \cdots$ が存在しなければ，t は停止性，あるいは強正規性を持つという．項 t が強正規性を持つとき $SN(R, t)$ と表す．R において，すべての t が停止性を持つ場合 R が停止性を持つといい，$SN(R)$ と表記する．HRS と項の直積上の述語 P に対して，項の集合 $\mathcal{T}_P(R) = \{t \mid P(R, t)\}$ および $\mathcal{T}_P^{args}(R) = \{t \mid \forall u \in args(t).P(R, u)\}$ を定義する．

HRS R の書換え規則 $l \to r$ の左辺の先頭記号 $top(l)$ を被定義記号と呼び，その集合を D_R と記す．また，R の被定義記号でない関数記号を R の構成子と呼び，その集合を C_R で記す．すべての $f \in D_R$ に対し，印付記号 $f^\#$ を用意する．また $t^\#$ を t の先頭記号を対応する印付記号で置き換えた印付項とする．なお，$top(t) \in \mathcal{V}$ の場合は $t^\# \equiv t$ とする．

3. 高階書換え系における強計算依存対法

本節では，単純型項書換え系における強計算依存対法 [4] を高階書換え系上で再構築する．強計算性は次のように定義する．

定義 3.1 (強計算性) HRS R において，項 t が強計算性（strongly computability）を持つ（$SC(R, t)$ と表記）とは，以下の帰納的な条件を満たすことをいう．

(1) $\tau(t) \in \mathcal{B}$ ならば $SN(R, t)$.

(2) $\tau(t) = \alpha \to \beta$ ならば
$\forall u.[SC(R, u) \wedge \tau(u) = \alpha \Rightarrow SC(R, (t\, u)\downarrow)]$.

依存対法では，真部分項は停止性を持つがそれ自体は停止性を持たない項に注目する．強計算依存対法では，引数は強計算性を持つがそれ自体は強計算性を持たない項に注目する．しかしながら強計算性は部分項関係に閉じていないため，引数の真部分項は強計算性を持つとは限らない．そこで強計算性を保証する項の集合を明示し，その集合に含まれる項は強計算依存対法の上では注目しないものとする．このような集合 $safe$ を規則に対して定義する．

定義 3.2 (safe subterm) R を HRS とする．書換え規則 $l \to r \in R$ の safe subterm の集合 $safe(l)$ を以下で定義する．

$$args(l) \cup \bigcup_{l' \in args(l)} \{u \in safe'(l', FV(l)) \mid FV(l) \supseteq FV(u)\}$$

ここで，$t \equiv \lambda x_1 \ldots x_m.a(t_1, \ldots, t_n)$ とするとき $safe'(t, P)$ は $a \in P$ ならば $\{a(t_1, \ldots, t_n)\}$，$a \notin P$ ならば $\{a(t_1, \ldots, t_n)\} \cup \bigcup_{i=1}^n safe'(t_i, P)$ を表す．

例 3.3 $l \equiv map(\lambda x.F(x), cons(X, L))$ である書換え規則 $l \to r \in R$ の safe subterm は以下の通りである．

$$safe(l) = \{\lambda x.F(x), cons(X, L), X, L\}$$

定義 3.4 (強計算依存対) R を HRS とする．書換え規則 $l \to r \in R$ が存在して，以下の条件を満たす対 $\langle l^\#, a^\#(r_1, \ldots, r_n)\rangle$ を R の強計算依存対（SC-dependency pair）と呼ぶ．

- $a \in D_R$.
- $\lambda x_1 \ldots x_m.a(r_1, \ldots, r_n) \in Cand(r)$.
- すべての $k\ (\leqq n)$ に対し，$a(r_1, \ldots, r_k)\downarrow \notin safe(l)$.

ここで，$Cand(t)$ の定義は $t \equiv \lambda x_1 \ldots x_m.a(t_1, \ldots, t_n)$ として以下の通り．

$$Cand(t) = \{t\} \cup \bigcup_{i=1}^n Cand(t_i\{x_j \mapsto c_{x_j} \mid 1 \leqq j \leqq m\})$$

ただし，c_{x_j} は束縛変数 x_j を関数記号で置き換えるための特殊な構成子である．R の強計算依存対の集合を $DP_{SC}(R)$ と表記する．

すべての強計算依存対は，高階型でなく，先頭記号が変数でない項の対である．

λ 項の部分項をとったとき，束縛変数が自由変数になってしまう場合がある．この問題に対応するために束縛変数 x を関数記号 c_x で置き換える $Cand(t)$ が導入された [6]．同様の理由

から本論文でも $Cand(t)$ を用いるが，定義自体は [6] と微妙に異なる．

例 3.5 1 節にて紹介した HRS R_2 の強計算依存対の集合 $DP_{SC}(R_2)$ は次の 4 対で構成されている．

$$\begin{cases} \langle add^{\#}(s(X),Y), add^{\#}(X,Y)\rangle \\ \langle map^{\#}(\lambda x.F(x), cons(X,L)), map^{\#}(\lambda x.F(x),L)\rangle \\ \langle d\_map^{\#}(L), map^{\#}(\lambda x.add(x,x),L)\rangle \\ \langle d\_map^{\#}(L), add^{\#}(x,x)\rangle \end{cases}$$

定義 3.6 (強計算依存鎖) P を項の対の集合とする．$\gg$ を項上の二項関係とし，$\mathcal{T}_l$ および $\mathcal{T}_r$ を項の集合とする．すべての i に対して $l_i\sigma_i{\downarrow} \in \mathcal{T}_l$, $v_i\sigma_i{\downarrow} \in \mathcal{T}_r$, $(v_i\sigma_i{\downarrow})^{\#} \gg^* (l_{i+1}\sigma_{i+1}{\downarrow})^{\#}$ であるような代入 σ_i が存在するとき，P の要素の列 $\langle u_0^{\#}, v_0^{\#}\rangle\langle u_1^{\#}, v_1^{\#}\rangle\cdots$ を $(\mathcal{T}_l,\mathcal{T}_r)$ 上の $\langle P,\gg\rangle$-鎖と呼ぶ．HRS R において強計算依存鎖 $DP_{SC}(R)$-鎖は $(\mathcal{T}_{SC}^{args}(R), \mathcal{T}_{SC}^{args}(R))$ 上の $\langle DP_{SC}(R), \rightarrow_R\rangle$-鎖である．

定義 3.7 (（近似）依存グラフ) P を項の対の集合とし，$\mathcal{T}_l$, $\mathcal{T}_r$ を項の集合とする．$(\mathcal{T}_l,\mathcal{T}_r)$ 上の $\langle P,\gg\rangle$-（近似）依存グラフは次の条件を満たす有効グラフである．

- 点集合は P.
- $\langle u_i^{\#}, v_i^{\#}\rangle\langle u_j^{\#}, v_j^{\#}\rangle$ が $(\mathcal{T}_l,\mathcal{T}_r)$ 上の $\langle P,\gg\rangle$-鎖ならば $\langle u_i^{\#}, v_i^{\#}\rangle$ から $\langle u_j^{\#}, v_j^{\#}\rangle$ への弧が存在する．

定義 3.8 (再帰成分) P を項の対の集合とし，$\mathcal{T}_l$ および $\mathcal{T}_r$ を項の集合とする．$(\mathcal{T}_l,\mathcal{T}_r)$ 上の $\langle P,\gg\rangle$-再帰成分（recursion components）とは，$(\mathcal{T}_l,\mathcal{T}_r)$ 上の $\langle P,\gg\rangle$-（近似）依存グラフの，ある強連結な部分グラフの点集合のことである．特に，すべての $(\mathcal{T}_{SC}^{args}(R), \mathcal{T}_{SC}^{args}(R))$ 上の $\langle DP_{SC}(R), \rightarrow_R\rangle$-再帰成分からなる集合を $RC_{SC}(R)$ で表す．

例 3.9 HRS R_2 の再帰成分 $C \in RC_{SC}(R_2)$ は次の 2 つの集合である．

$$\begin{cases} \{\langle add^{\#}(s(X),Y), add^{\#}(X,Y)\rangle\} \\ \{\langle map^{\#}(\lambda x.F(x), cons(X,L)), map^{\#}(\lambda x.F(x),L)\rangle\} \end{cases}$$

これらは d_map 関数の再帰構造にあたり，プログラマの直感に極めて近い．強計算依存対法はプログラマが直感的に理解する再帰構造を定式化することに成功した．

次に [4] で提案された，高階変数が左辺の引数から直接右辺に渡されるという直接関数渡しの条件を HRS 上で再定義する．この際，匿名関数に対応するため，右辺に現れる高階型の項は自身の束縛変数を引数にのみ持つという条件を加える．

定義 3.10 (直接関数渡し) HRS R が，すべての $l \rightarrow C[\lambda x_1 \ldots x_m.a(r_1,\ldots,r_n)] \in R$ に対して以下の条件を満たすとき，直接関数渡し（plain function-passing, PFP）と呼ぶ．

- $a \in \mathcal{V}$ ならば，$\exists k \leqq n.[a(r_1,\ldots,r_k){\downarrow} \in safe(l)]$.
- $\forall r_i.[\exists x_j.[r_i \equiv x_j{\downarrow}] \vee \{x_1,\ldots,x_m\} \cap FV(r_i) = \emptyset]$.

直接関数渡しである HRS を PFP-HRS と略記する．

次に，PFP-HRS を解析した再帰成分が無限ループを作らないことを保証する 2 つの条件を定義する．1 つは存在することによって PFP-HRS の停止性を保証することができる簡約化対．もう 1 つは [2] で定義され, [4] で拡張された部分項基準である．

定義 3.11 (簡約化対) $\gtrsim$ を擬順序，$>$ を狭義の半順序とする．対 $(\gtrsim,>)$ が以下を満たすとき，簡約化対と呼ぶ.

- $\gtrsim$ が文脈および代入に閉じている．
- $>$ が整礎でかつ代入に閉じている．
- $\gtrsim\circ>\subseteqq>$ もしくは $>\circ\gtrsim\subseteqq>$ が成り立つ．

定義 3.12 R を PFP-HRS，$C \in RC_{SC}(R)$ とする．以下の条件を満たす，D_R から空でない正整数のリストへの関数 π が存在するならば C は部分項基準を満たすという．

(α) ある $\langle u^{\#}, v^{\#}\rangle \in C$ が $u|_{\pi(top(u))} >_{sub} v|_{\pi(top(v))}$.

(β) すべての $\langle u^{\#}, v^{\#}\rangle \in C$ が以下の条件を満たす．

- $u|_{\pi(top(u))} \geqq_{sub} v|_{\pi(top(v))}$
- $\forall p \prec \pi(top(u)).[top(u|_p) \notin \mathcal{V}]$
- $\forall q \prec \pi(top(v)).[q = \varepsilon \vee top(v|_q) \in C_R]$

強計算性に基づき，簡約化対および部分項基準の概念を用いて，HRS の強力な停止性証明法を与える．

定理 3.13 R を PFP-HRS とする．すべての $C \in RC_{SC}(R)$ が以下の性質のいずれかを満たすならば R は停止する．

- C が部分項基準を満たす．
- $R \cup C \subseteqq \gtrsim$, $C \cap > \neq \emptyset$ を満たす簡約化対が存在する．

証明 次節で与える定理 4.8 および補題 4.9，4.11 より明らか.

例 3.14 1 節にて紹介した PFP-HRS R_2 の停止性を示す. $\pi(add) = 1$, $\pi(map) = 2$ とすると，例 3.9 で示した再帰成分 $C \in RC_{SC}(R_2)$ はいずれも部分項基準を満たす．よって定理 3.13 より R_2 は停止性を持つ．

4. 主定理の証明

本節では，前節で省いた定理 4.8 および補題 4.9，4.11 の証明を与える．

補題 4.1 HRS R において，ある項 s,t が $s \overset{*}{\rightarrow}_R t$ を満たすとする．このとき，任意の代入 σ に対して $s\sigma{\downarrow} \overset{*}{\rightarrow}_R t\sigma{\downarrow}$ が成立.

証明 $s \rightarrow_R t$ の場合を示せば十分．$\rightarrow_R$ の定義より，$s \equiv C[l\theta{\downarrow}], t \equiv C[r\theta{\downarrow}]$ を満たす文脈 $C[]$，代入 θ，書換え規則 $l \rightarrow r \in R$ が存在する．ここで $C'[,\ldots,] \equiv C[]\sigma{\downarrow}$ とすると，

$\tau(l) = \tau(r) \in \mathcal{B}$ より $s\sigma{\downarrow} \equiv C[l\theta{\downarrow}]\sigma{\downarrow} \equiv C'[l\theta\sigma{\downarrow}, \ldots, l\theta\sigma{\downarrow}] \xrightarrow{*}_R C'[r\theta\sigma{\downarrow}, \ldots, r\theta\sigma{\downarrow}] \equiv C[r\theta{\downarrow}]\sigma{\downarrow} \equiv t\sigma{\downarrow}$. □

補題 4.2 $SC(R, \lambda x.t) \Rightarrow SC(R, t)$.

証明 $s \equiv \lambda x.t$ とする．強計算性の定義から，$\tau(u) = \alpha$ かつ $SC(R, u)$ である任意の項 u に対して $SC(R, (s\,u){\downarrow})$．このとき $u \equiv x{\downarrow}$ とすれば $(s\,x){\downarrow} \equiv t$．ゆえに $SC(R, t)$. □

補題 4.3 すべての HRS R について，強計算性を持つ項は以下の性質を満たす.

(1) $SC(R, t) \Rightarrow SN(R, t)$.

(2) $SC(R, s) \wedge s \xrightarrow{*}_R t \Rightarrow SC(R, t)$.

(3) $\bigwedge_{i=0}^{n} SC(R, t_i) \Rightarrow SC(R, t_0(t_1, \ldots, t_n){\downarrow})$.

証明 (1), (3) は自明であるため (2) の証明のみを記述する．$s \equiv \lambda x_1 \ldots x_m.s', t \equiv \lambda x_1 \ldots x_m.t'$ とする．強計算性の定義から，$\tau(x_i) = \tau(u_i)$ である任意の $u_1, \ldots, u_m \in \mathcal{T}_{SC}(R)$ に対して $SC(R, s(u_1, \ldots, u_m){\downarrow})$ が成立する．また，補題 4.2 より $SC(R, s')$．このとき，任意の書換え規則 $l \to r \in R$ において $\tau(l) \in \mathcal{B}$ であるから $s' \xrightarrow{*} t'$．ここで補題 4.1 より $\sigma = \{x_i \mapsto u_i \mid 1 \leqq m\}$ として，$s'\sigma{\downarrow} \xrightarrow{*} t'\sigma{\downarrow}$ が成り立つ．このとき，$s'\sigma{\downarrow} \equiv s(u_1, \ldots, u_m){\downarrow}$ より $SC(R, s'\sigma{\downarrow})$ なので，(1) より $SN(R, s'\sigma{\downarrow})$．従って $SN(R, t'\sigma{\downarrow})$ である．ここで，$\tau(s'\sigma{\downarrow}) = \tau(t'\sigma{\downarrow}) \in \mathcal{B}$ であるから $SC(R, t'\sigma{\downarrow})$ が成り立つ．$t'\sigma{\downarrow} \equiv t(u_1, \ldots, u_m){\downarrow}$ より，強計算性の定義から $SC(R, t)$. □

補題 4.4 項 t を $\tau(t) = \alpha_1 \to \cdots \to \alpha_n \to \alpha \in \mathcal{B}$ かつ $\neg SC(R, t)$ であるような項とする．このとき，$\forall i.[\tau(u_i) = \alpha_i \wedge SC(R, u_i)]$ であるような項 $u_1, \ldots, u_n$ が存在して $\neg SC(R, t(u_1, \ldots, u_n){\downarrow})$.

証明 強計算性の定義より自明. □

補題 4.5 $\neg SN(R) \Rightarrow \mathcal{T}_{\mathcal{B}} \cap \mathcal{T}_{\neg SC}(R) \cap \mathcal{T}_{SC}^{args}(R) \neq \emptyset$.

証明 補題 4.3 (1) の対偶より $\neg SC(R)$ である．項 $t \equiv \lambda x_1 \ldots x_m.a(t_1, \ldots, t_n)$ は $\neg SC(R, t)$ を満たすサイズ最小の項とする．ここで 補題 4.4 より，ある $u_1, \ldots, u_m \in \mathcal{T}_{SC}(R)$ が存在して $t' \equiv t(u_1, \ldots, u_m){\downarrow}$ としたとき，$\neg SC(R, t')$ を満たす．このとき，$\sigma = \{x_j \mapsto u_j \mid 1 \leqq j \leqq m\}$ とすると $t' \equiv a\sigma(t_1\sigma, \ldots, t_n\sigma){\downarrow}$ である．ここで $t'_i \equiv \lambda x_1 \ldots x_m.t_i$ は t より小さいので $SC(R, t'_i)$．各 i において $t_i\sigma{\downarrow} \equiv t'_i(u_1, \ldots, u_m){\downarrow}$ であるため，補題 4.3 (3) より $SC(R, t_i\sigma{\downarrow})$.

- $a \in \{x_1, \ldots, x_m\}$ のとき，$a\sigma{\downarrow} \equiv u_j \in \mathcal{T}_{SC}(R)$ であるから，補題 4.3 (3) より $SC(R, t')$ となる．これは矛盾する.
- $a \notin \{x_1, \ldots, x_m\}$ のとき，$t' \equiv a(t_1\sigma{\downarrow}, \ldots, t_n\sigma{\downarrow})$ であるから $t' \in \mathcal{T}_{\mathcal{B}} \cap \mathcal{T}_{\neg SC}(R) \cap \mathcal{T}_{SC}^{args}(R)$. □

補題 4.6 R を HRS とし，代入 σ および，書換え規則 $l \to r \in R$ を考える．ここで $l\sigma{\downarrow} \in \mathcal{T}_{SC}^{args}(R)$ ならば，すべての $s \in safe(l)$ に対して $SC(R, s\sigma{\downarrow})$ が成り立つ.

証明 $l \equiv f(l_1, \ldots, l_n)$ とする．以下，$s \equiv l_i$ と $s \in safe'(l_i, FV(l))$ の場合に分けて $SC(R, s\sigma{\downarrow})$ を示す.

- $s \equiv l_i$ のとき，$l\sigma{\downarrow} \in \mathcal{T}_{SC}^{args}(R)$ より $SC(R, s\sigma{\downarrow})$.
- $s \in safe'(l_i, FV(l))$ のとき，$safe'$ の定義より，明らかに $\tau(s) \in \mathcal{B}$ なので，$SN(R, s\sigma{\downarrow})$ を示せば十分．$l\sigma{\downarrow} \in \mathcal{T}_{SC}^{args}(R)$ および補題 4.3 (1) より，$SN(R, l_i\sigma{\downarrow})$．以下，$SN(R, t\sigma{\downarrow})$ となる任意の $t \equiv \lambda x_1 \ldots x_m.a(t_1, \ldots, t_n) \in Sub(l_i)$ に対し $s \in safe'(t, FV(l))$ ならば $SN(R, s\sigma{\downarrow})$ であることを，$safe'$ の定義に関する帰納法で証明する.
  - $t \equiv \lambda x_1 \ldots x_m.s$ のとき，$t\sigma{\downarrow} \equiv \lambda x_1 \ldots x_m.(s\sigma{\downarrow})$ である．ゆえに $SN(R, s\sigma{\downarrow})$.
  - $s \in safe'(t_j, FV(l))$ $(1 \leqq j \leqq n)$ のとき，$FV(l) \supseteq Dom(\sigma)$ かつ $a \notin FV(l)$ より $a \notin Dom(\sigma)$ となる．よって $a\sigma{\downarrow} \equiv a{\downarrow}$ なので $t\sigma{\downarrow} \equiv \lambda x_1 \ldots x_m.a(t_1\sigma{\downarrow}, \ldots, t_n\sigma{\downarrow})$．ゆえに $SN(R, t_j\sigma{\downarrow})$ が成り立つ．帰納法の仮定より $SN(R, s\sigma{\downarrow})$. □

この証明から $safe$ を $args(l) \cup \bigcup_{l_i \in args(l)} safe'(l_i, FV(l))$ で定義できることがわかる．しかし，$safe'$ に含まれるすべての項が必要となるわけではない．$safe$ は直接関数渡しと強計算依存対の定義でのみ利用される．これらの定義は書換え規則の右辺 r に依存する定義である．このとき，$safe'(l_i, FV(l))$ には l_i の束縛変数を持つ項が含まれる場合があるが，そのような項は r に表れることはない．よって $FV(l) \supseteq FV(u)$ という制限を課しても，直接関数渡しと強計算依存対の定義に影響しない．ゆえに $safe'$ に制限 $FV(l) \supseteq FV(u)$ を課した $safe$ を safe subterm として定義した.

補題 4.7 R を PFP-HRS とする．すべての $t \in \mathcal{T}_{\mathcal{B}} \cap \mathcal{T}_{\neg SC}(R) \cap \mathcal{T}_{SC}^{args}(R)$ に対して，依存対 $\langle l^\#, v^\# \rangle \in DP_{SC}(R)$ および代入 σ が存在し，$t^\# \xrightarrow{*} (l\sigma{\downarrow})^\#$ かつ $l\sigma{\downarrow}, v\sigma{\downarrow} \in \mathcal{T}_{\mathcal{B}} \cap \mathcal{T}_{\neg SC}(R) \cap \mathcal{T}_{SC}^{args}(R)$ となる.

証明 $t \in \mathcal{T}_{SC}^{args}(R)$ および補題 4.3 (1) より $t \in \mathcal{T}_{SN}^{args}(R)$．また，$t \in \mathcal{T}_{\mathcal{B}} \cap \mathcal{T}_{\neg SC}(R)$ および強計算性の定義より $\neg SN(R, t)$ である．よって，ある書換え規則 $l \to r \in R$ および代入 σ' が存在して $t^\# \xrightarrow{*} (l\sigma'{\downarrow})^\#$ かつ $l\sigma'{\downarrow}, r\sigma'{\downarrow} \in \mathcal{T}_{\neg SN}(R)$．また $l, r \in \mathcal{T}_{\mathcal{B}}$ より，強計算性の定義から $l\sigma'{\downarrow}, r\sigma'{\downarrow} \in \mathcal{T}_{\neg SC}(R)$ となる．さらに，補題 4.3 (2) より $l\sigma'{\downarrow} \in \mathcal{T}_{SC}^{args}(R)$．ここで，$\neg SC(R, r\sigma'{\downarrow})$ より以下が成立する.

$$\{r' \in Cand(r) \mid \neg SC(R, r'\sigma'{\downarrow})\} \neq \emptyset$$

上記の集合を P とし，P においてサイズ最小の項を $v' \equiv \lambda x_1 \ldots x_m.a(r_1, \ldots, r_n)$ とする．このとき補題 4.4 より，強計算性を持つ項 $u_1, \ldots, u_m$ が存在して $\neg SC(R, v'(u_1, \ldots, u_m){\downarrow})$．$v \equiv a(r_1, \ldots, r_n)$, $\sigma = \sigma' \cup \{x_i \mapsto u_i \mid 1 \leqq i \leqq m\}$ とすると，$v\sigma{\downarrow} \equiv a\sigma(r_1\sigma, \ldots, r_n\sigma){\downarrow}$．このとき，$v\sigma{\downarrow} \in \mathcal{T}_{\mathcal{B}} \cap \mathcal{T}_{\neg SC}(R)$ である．また，$x_i \notin FV(l)$ より $l\sigma{\downarrow} \equiv l\sigma'{\downarrow}$ であるから，$l\sigma{\downarrow} \in \mathcal{T}_{\mathcal{B}} \cap \mathcal{T}_{\neg SC}(R) \cap \mathcal{T}_{SC}^{args}(R)$．PFP の定義より $\forall i.[\exists j.[r_i \equiv x_j{\downarrow}] \vee \{x_1, \ldots, x_m\} \cap FV(r_i) = \emptyset]$ である．次に，すべての r_i

について $SC(R, r_i\sigma\downarrow)$ が成り立つことを示す．

- $r_i \equiv x_j\downarrow (1 \leqq j \leqq m)$ ならば，$r_i\sigma\downarrow \equiv u_j \in \mathcal{T}_{SC}(R)$.
- $\{x_1, \ldots, x_m\} \cap FV(r_i) = \emptyset$ とする．このとき，$r_i \in Cand(r)$ かつ $r_i \notin P$ なので $SC(R, r_i\sigma\downarrow)$．また $r_i\sigma'\downarrow \equiv r_i\sigma\downarrow$ も成り立つ．ゆえに $SC(R, r_i\sigma\downarrow)$.

最後に $v\sigma\downarrow \in \mathcal{T}_{SC}^{args}(R)$，$\langle l^\#, v^\# \rangle \in DP_{SC}(R)$ を示す．

- $a \in \{x_i \mid 1 \leqq i \leqq m\}$ とする．$SC(R, a\sigma\downarrow)$ より，補題 4.3（3）から $SC(R, v\sigma\downarrow)$．これは矛盾する．
- $a \in FV(r)$ とする．R は PFP-HRS であるため，PFP の定義より，ある $k(\leqq n)$ が存在して $a(r_1, \ldots, r_k)\downarrow \in safe(l)$．このとき，補題 4.6 より $SC(R, a(r_1, \ldots, r_k)\sigma\downarrow)$ となる．ゆえに，補題 4.3（3）から $SC(R, v\sigma\downarrow)$．これは矛盾する．
- $a \in C_R$ とする．各 i において $SC(R, r_i\sigma\downarrow)$ なので，補題 4.3（1）より $SN(R, r_i\sigma\downarrow)$．また，$a \in C_R$ より $SN(R, v\sigma\downarrow)$．$v \in \mathcal{T}_\mathcal{B}$ より $SC(R, v\sigma\downarrow)$．これは矛盾する．
- $a \in D_R$ かつ，ある k $(\leqq n)$ が存在して $a(r_1, \ldots, r_k)\downarrow \in safe(l)$ とする．このとき，補題 4.6 より $SC(R, a(r_1, \ldots, r_k)\sigma\downarrow)$ となる．ゆえに，補題 4.3（3）から $SC(R, v\sigma\downarrow)$．これは矛盾する．

ゆえに $a \in D_R$ かつ，すべての $k(\leqq n)$ に対し $a(r_1, \ldots, r_k)\downarrow \notin safe(l)$ である．よって $\langle l^\#, v^\# \rangle \in DP_{SC}(R)$ を満たす．また，$v\sigma\downarrow \equiv a(r_1\sigma\downarrow, \ldots, r_n\sigma\downarrow)$ かつ，すべての r_i が $SC(R, r_i\sigma\downarrow)$ であるため，$v\sigma\downarrow \in \mathcal{T}_{SC}^{args}(R)$ である． □

定理 4.8 R を PFP-HRS とする．無限 $DP_{SC}(R)$-鎖が存在しなければ R は停止性を持つ．

証明 $\neg SN(R)$ と仮定する．補題 4.5 より項 $t \in \mathcal{T}_\mathcal{B} \cap \mathcal{T}_{\neg SC}(R) \cap \mathcal{T}_{SC}^{args}(R)$ が存在する．補題 4.7 を繰り返し適用することで，無限 $DP_{SC}(R)$-鎖が得られる．これは矛盾する． □

補題 4.9 R を PFP-HRS とする．すべての $C \in RC_{SC}(R)$ に対し，ある簡約化対 $(\gtrsim, >)$ が存在して $R \cup C \subseteqq \gtrsim$ かつ $C \cap > \neq \emptyset$ となるならば $SN(R)$ が成立．

証明 明らか． □

補題 4.10 $>_{sub} \circ \to_R \subseteqq \to_R \circ >_{sub}$.

証明 $s >_{sub} t \to_R u \Rightarrow s(\to_R \circ >_{sub})u$ を s に関する帰納法で証明できる． □

補題 4.11 R を HRS，C を印付項の対の集合とする．C が部分項基準を満たすとき，各 $\langle u^\#, v^\# \rangle \in C$ が無限個出現するような $(\mathcal{T}_{SC}^{args}(R), \mathcal{T}_{SC}^{args}(R))$ 上の無限 $\langle C, \to_R \rangle$-鎖 が存在しない．

証明 $\langle u_0^\#, v_0^\# \rangle \langle u_1^\#, v_1^\# \rangle \langle u_2^\#, v_2^\# \rangle \cdots$ のように，各 $\langle u^\#, v^\# \rangle \in C$ が無限個出現する $(\mathcal{T}_{SC}^{args}(R), \mathcal{T}_{SC}^{args}(R))$ 上の無限 $\langle C, \to_R \rangle$-鎖が存在すると仮定する．このとき，$(\mathcal{T}_{SC}^{args}(R), \mathcal{T}_{SC}^{args}(R))$ 上の無限鎖は $(\mathcal{T}_{SN}^{args}(R), \mathcal{T}_{SN}^{args}(R))$ 上に存在している．$\theta_0, \theta_1, \ldots$ を $(v_i\theta_i\downarrow)^\# \xrightarrow[R]{*} (u_{i+1}\theta_{i+1}\downarrow)^\#$ かつ $u_i\theta_i\downarrow, v_i\theta_i\downarrow \in \mathcal{T}_{SN}^{args}(R)$ である代入とし，各 i において $\pi(top(u_i)) = p_i$ とすると，$(v_i\theta_i\downarrow)^\# \xrightarrow[R]{*} (u_{i+1}\theta_{i+1}\downarrow)^\#$ なので $top(v_i) = top(u_{i+1})$．ゆえに条件（β）から以下が成立．

$$\begin{aligned}(u_0\theta_0\downarrow)|_{p_0} &\geqq_{sub} (v_0\theta_0\downarrow)|_{p_1} \xrightarrow[R]{*} (u_1\theta_1\downarrow)|_{p_1} \\ &\geqq_{sub} (v_1\theta_1\downarrow)|_{p_2} \xrightarrow[R]{*} \cdots\end{aligned}$$

条件（α）より無限個の $>_{sub}$ が上記の列に含まれている．$>_{sub}$ が整礎でかつ $>_{sub} \circ \to_R \subseteqq \to_R \circ >_{sub}$ を満たす（補題 4.10）ため，$(u_0\theta_0\downarrow)|_{p_0}$ から始まる無限の書換え関係が存在する．p_0 は空でないため $j \preceq p_0$ を満たす正整数 j が存在する．$u_0\theta_0\downarrow \in \mathcal{T}_{SN}^{args}(R)$ であるため $SN(R, (u_0\theta_0\downarrow)|_j)$ が成り立つ．ゆえに $SN(R, (u_0\theta_0\downarrow)|_{p_0})$．これは矛盾する． □

5. おわりに

強計算依存対法を HRS 上で再構築したことによって，既存の依存対法では証明できなかった HRS の停止性証明が可能となった．ただし，定理 3.13 における 2 つの条件のうちの 1 つ，簡約化対の設計法は知られていない．そのため，本手法によって停止性を証明できるのは部分項基準を満たす PFP-HRS に限られてしまう．今後の課題に，定理 3.13 の条件を満たすような簡約化対の設計がある．

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] T.Arts and J.Giesl, Termination of term rewriting using dependency pairs, Theoretical Computer Science, vol.236, pp.133–178, 2000.

[2] Hirokawa,N., Middeldorp,A., Dependency Pairs Revisited. Proc. of the 15th Int. Conf. on Rewriting Techniques and Applications, LNCS 3091 (RTA04), pp.249–268, 2004.

[3] Kusakri,K., On Proving Termination of Term Rewriting Systems with Higher-Order Variables, IPSJ Transactions on Programming, Vol.42, No.SIG 7 (PRO 11), pp.35–45, Jul 2001.

[4] Kusakari,K., Sakai,M., Enhancing Dependency Pair Method by Strong Computability in Simply-Typed Term Rewriting, 電子情報通信学会技術研究報告 (SS2005-65), Vol.105, No.491, pp.13–18, 2005.

[5] T.Nipkow, Higher-order critical pairs, Proc. 6th Annual IEEE Symposium on Logic in Computer Sciene, pp.342–349, 1991.

[6] Sakai,M., Watanabe,Y., An Extension of the Dependency Pair Method for Proving Termination of Higher-Order Rewrite Systems, IEICE Transactions on Information and Systems, Vol.E84-D, No.8, pp.1025–1032, 2001.

[7] Sakai,M., Kusakari,K., On Dependency Pair Method for Proving Termination of Higher-Order Rewrite Systems, IEICE Transactions on Information and Systems, Vol.E88-D, No.3, pp.583–593, 2005.

[8] 高橋正子: 計算論 –計算可能性とラムダ計算–, 近代科学社, 1991.

[9] Terese, Term Rewriting Systems, Cambridge Tracts in Theoretical Computer Science, Vol.55, Cambridge University Press, 2003.