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

信学技報
IEICE Technical Report
SS2007-13 (2007-06)

二階の書換え系における引数切り落とし法

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 高階書換え系は関数プログラムの計算モデルであり，停止性は重要な性質の一つである．停止性証明法の一つに強計算性に基づく静的依存対法と呼ばれる再帰構造解析法がある．依存対法を用いる際には，引数切り落とし法と呼ばれる手法が重要となる．高階の書換え系に適用可能な引数切り落とし法は既に知られているが，適用する際に，λ 抽象に対応していないという問題と，型の構造を破壊してしまうという問題がある．本論文では，これら二つの問題を解決する高階書換え系上の引数切り落とし法を提案する．提案した手法の正当性を一般には証明できなかったが，規則の各辺が堅固または二階の場合には健全であることを証明する．

キーワード 関数プログラム，高階書換え系，停止性，強計算性，静的依存対法，引数切り落とし法

Argument Filtering Method for Second-Order Higher-Order Rewrite Systems

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 Higher-order rewrite systems are computation models of functional programming languages, and the termination property is one of the most important ones of them. Recently, static dependency pair method based on strong computability was introduced, which proves the termination effectively and efficiently. An argument filtering method plays an important role in this method. However, existing argument filtering method in higher-order rewrite systems has two problems: it cannot handle λ-abstraction, and destructs type structures. In order to overcome these problems, we extend the method. Although we did not show its soundness in general, we prove the soundness under the restriction that both sides of rules are either firmness or second-order.

Key words Functional Program, Higher-Order Rewrite System, Termination, Strong Computability, Static Dependency Pair Method, Argument Filtering Method.

1. はじめに

項書換え系（Term Rewriting System, TRS）に λ 計算を融合させ，高階関数の利用を可能とした計算モデルを高階書換え系（Higher-Order Rewrite System, HRS）と呼ぶ [8]．以下に示す HRS は代表的な高階関数である map を表している．

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

HRS は λ 計算を利用しているため，単に高階変数を扱えるだけでなく，匿名関数を用いたプログラムもそのまま表現することが可能となる．

近年提案された強力な停止性証明法に，強計算性に基づく静的依存対法がある [7]．この手法は単純型付き項書換え系（Simply-Typed Term Rewriting System, STRS）で提案された手法で，ある種の再帰構造を解析することによって停止性を証明する．その後，この手法は HRS 上に拡張された [3], [6]．例えば，TRS 上で提案され [2]，STRS 上 [7]，HRS 上 [3] へ拡張された部分項基準と呼ばれる概念を，静的依存対法と組み合

わせることにより，map における再帰成分の第二引数（下線部参照）が $cons(X,L)$ から L のように部分項関係の意味で減少するという理由から，HRS R_1 の停止性を示すことができる．この自然な推論に理論的基礎を与えることはそれほど容易ではなく，型付き λ 計算の停止性証明に導入された強計算性の概念が必要であった．

ここで次の HRS R_2 を考える．

$$R_1 \cup \begin{cases} sub(X,0) & \rightarrow & X \\ sub(0,Y) & \rightarrow & 0 \\ sub(s(X),s(Y)) & \rightarrow & sub(X,Y) \\ div(0,s(Y)) & \rightarrow & 0 \\ div(\underline{s(X)},s(Y)) & \rightarrow & s(div(\underline{sub(X,Y)},s(Y))) \\ divs(Z,L) & \rightarrow & map(\lambda x.div(x,Z),L) \end{cases}$$

文献 [6] の手法でこの HRS の停止性を証明するためには，関数 div における再帰成分の第一引数（下線部参照）が簡約化対 $(\gtrsim,>)$ と呼ばれる順序の対の $>$ により $s(X) > sub(X,Y)$ のように順序付けられる必要がある．このような簡約化対の設計には引数切り落とし法（argument filtering method）が用いられる．これは項を比較する際に不必要な引数を切り落としてから比較するという手法である．例えば sub の第二引数を切り落とし Y を除去することにより $s(X) > sub(X,Y)$ を $s(X) > sub(X)$ として比較する．引数切り落とし法は TRS 上で提案され [1]，STRS 上に拡張されている [4]．これらの手法には二つの問題がある．一つは λ 抽象に対応していないという問題．もう一つは型の構造が破壊されるという問題である．

本論文では引数切り落とし法を λ 抽象に対応している HRS 上に拡張する．また，$\perp$ 記号の導入により型構造の破壊を解決する．例えば上記の $sub(X,Y)$ においては，既存の手法では $sub(X)$ とするのに対し，本論文の手法では $sub(X,\perp)$ とすることで型の構造が破壊されることを防いでいる．残念ながら一般の HRS において本手法の健全性は現在のところ示すことができていないが，各規則の両辺が堅固または 2nd-order である場合には健全であることを示す．

本論文の構成は次の通り．2 節では本論文で必要となる定義や記法を与える．3 節では文献 [6] で与えられた強計算性に基づく静的依存対法を紹介する．4 節で HRS 上の引数切り落とし法を提案する．5 節で今後の課題を述べる．

2. 準　　備

本節では文献 [8] に基づき，後の節で必要となる定義や記述方法を与える．

$\rightarrow$ によって生成される単純型の集合を $\mathcal{S}$ とする．型 α の変数の集合を $\mathcal{V}_\alpha$ と表記し，同様に型 α の関数記号の集合も Σ_α と表記する．また $\mathcal{V} = \bigcup_{\alpha\in\mathcal{S}} \mathcal{V}_\alpha$，$\Sigma = \bigcup_{\alpha\in\mathcal{S}} \Sigma_\alpha$ と定義する．$\mathcal{V}\cup\Sigma$ から λ 抽象と λ 適用によって生成される型 α の単純型 preterm の集合を $\mathcal{T}^{pre}_\alpha$ とする．単純型 preterm t の η ロング β 正規形を $t{\downarrow}$ と表記する．型 α を持つ単純型項の集合 $\mathcal{T}_\alpha$ を $\{t{\downarrow} \mid t\in\mathcal{T}^{pre}_\alpha\}$ で定義する．$t\in\mathcal{T}_\alpha$ のとき $type(t)=\alpha$ と表記．また単純型項全体の集合を $\mathcal{T} = \bigcup_{\alpha\in\mathcal{S}}\mathcal{T}_\alpha$，基本型である項の集合を $\mathcal{T}_\mathcal{B} = \bigcup_{\alpha\in\mathcal{B}}\mathcal{T}_\alpha$ と定義する．以下，単純型項を単に項と記す．本論文では，項を $\lambda x_1\ldots x_m.a(t_1,\ldots,t_n)$ または省略して $\lambda\overline{x_m}.a(\overline{t_n})$ と表記する．また変数と関数記号は重ならないものとする．項の α 合同を $\equiv$ で表記する．項 $t\equiv\lambda\overline{x_m}.a(\overline{t_n})$ に対し，t の先頭記号を $top(t)=a$ と表記する．また t の引数の集合を $args(t)=\{t_1,\ldots,t_n\}$ と表記する．t 中の自由変数の集合を $FV(t)$ と表記する．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．$s\in Sub(t)$ のとき $t\geq_{sub} s$ と表記し，$>_{sub}=\geq_{sub}\setminus\equiv$ と定義する．正整数の列である項 t の位置の集合 $Pos(t)$ を $Pos(\lambda x.t)=\{\varepsilon\}\cup\{1p\mid p\in Pos(t)\}$，$Pos(a(t_1,\ldots,t_n))=\{\varepsilon\}\cup\bigcup_{i=1}^n\{ip\mid p\in Pos(t_i)\}$ と定義する．位置上の順序 $\succ$ を $p\succ q \Longleftrightarrow \exists w.[p=qw\wedge w\neq\varepsilon]$ と定義し，位置 p における項 t の部分項を $t|_p$ と表記する．

$top(l)\in\Sigma, type(l)=type(r)\in\mathcal{B}$ かつ $FV(l)\supseteq FV(r)$ であるような項の対 (l,r) を書換え規則と呼び $l\rightarrow r$ と表記する．高階書換え系（HRS）は書換え規則の集合である．HRS R の書換え関係 $\underset{R}{\rightarrow}$ を，ある書換え規則 $l\rightarrow r\in R$，文脈 $C[]$ および代入 σ が存在して $s\equiv C[l\sigma{\downarrow}]$ かつ $t\equiv C[r\sigma{\downarrow}]$ ならば，そのときに限り $s\underset{R}{\rightarrow}t$ と定義する．

HRS R において，項 t から始まる無限の書換え系列が存在しなければ t は停止性を持つ，または強正規性を持つと呼び $SN(R,t)$ と表記する．

HRS R の書換え規則の左辺の先頭記号を被定義記号と呼び，その集合を $\mathcal{D}_R$ と表記する．一方，それ以外の関数記号を構成子記号と呼び，その集合を $\mathcal{C}_R$ と表記する．各被定義記号 $f\in\mathcal{D}_R$ に対して印付記号 $f^\#$ を用意し，印付項 $t^\#$ を次のように定義する．t が $a\in\mathcal{D}_R$ であるような $a(t_1,\ldots,t_n)$ という形をとっていれば $t^\#\equiv a^\#(t_1,\ldots,t_n)$，それ以外の場合 $t^\#\equiv t$．

3. 強計算性に基づく静的依存対法

強計算性に基づく静的依存対法は，草刈と酒井により STRS 上で提案された静的再帰構造解析による非常に強力な停止性証明法である [7]．STRS は HRS と異なり λ 抽象を扱えないため，我々はこれを HRS 上に拡張した [3], [6]．本節では，文献 [6] で与えられた静的依存対法を紹介する．

定義 3.1 (強計算性)　HRS R において，項 t が強計算性（strong computability）を持つ（$SC(R,t)$ と表記）ということを，以下のように定義する．

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

(2)　$type(t)=\alpha\rightarrow\beta$ ならば
$\forall u\in\mathcal{T}_\alpha.(SC(R,u)\Rightarrow SC(R,(tu){\downarrow}))$ で定義．

また $\mathcal{T}^{args}_{SC}(R)=\{t\mid\forall u\in args(t).SC(R,u)\}$ と定義する．

定義 3.2　R を HRS とし，$l\rightarrow r\in R$ とする．l の pattern computability subterm の集合 $pcs(l)$ を以下のように定義する．

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

ここで $pcs'(\lambda\overline{x_m}.a(\overline{t_n}), V)$ は $a \in V$ ならば $\{a(\overline{t_n})\}$，それ以外の場合は $\{a(\overline{t_n})\} \cup \bigcup_{i=1}^{n} pcs'(t_i, V)$ と定義する．

定義 3.3 (直接関数渡し) HRS R において，全ての書換え規則 $l \to r \in R$ と 先頭記号が自由変数であるような全ての $F(r_1, \ldots, r_n) \in Sub(r)$ に対して $F(r_1, \ldots, r_k)\downarrow\ \in pcs(l)$ であるような $k\ (\leqq n)$ が存在するとき R を直接関数渡し（plain function-passing, PFP）と呼ぶ．

以降，直接関数渡しである HRS を PFP-HRS と省略する．

定義 3.4 項 $t \equiv \lambda\overline{x_m}.a(\overline{t_n})$ に対し $Cand(t)$ を以下のように定義する．

$$Cand(t) = \{t\} \cup \bigcup_{i=1}^{n} Cand(\lambda\overline{x_m}.t_i)$$

定義 3.5 (静的依存対) R を HRS とする．対 $\langle l^\#, a^\#(\overline{r_n})\rangle$ が静的依存対（static dependency pair）であるとは，以下を満たす $l \to r \in R$ が存在することである．

- $a \in \mathcal{D}_R$
- $\lambda\overline{x_m}.a(\overline{r_n}) \in Cand(r)$
- $\forall k \leqq n. a(\overline{r_k})\downarrow\ \notin pcs(l)$

R における静的依存対の集合を $SDP(R)$ と記述する．以降，静的依存対を単に依存対と呼び，依存対 $\langle l^\#, a^\#(\overline{r_n})\rangle$ を $l^\# \to a^\#(\overline{r_n})$ と記述する．

例 3.6 $SDP(R_2)$ は次のようになる．

$$\left\{\begin{array}{l} sub^\#(s(X), s(Y)) \ \to\ sub^\#(X, Y) \\ div^\#(s(X), s(Y)) \ \to\ div^\#(sub(X, Y), s(Y)) \\ divs^\#(Z, L) \ \to\ map^\#(\lambda x.div(x, Z), L) \\ divs^\#(Z, L) \ \to\ div^\#(x, Z) \\ map^\#(\lambda x.F(x), cons(X, L)) \\ \qquad\qquad \to\ map^\#(\lambda x.F(x), L) \end{array}\right\}$$

定義 3.7 (静的再帰成分) HRS R において，各 i に対し $(v_i\sigma_i\downarrow)^\# \xrightarrow{*} (u_{i+1}\sigma_{i+1}\downarrow)^\#$ かつ $u_i\sigma_i\downarrow, v_i\sigma_i\downarrow\ \in \mathcal{T}^{args}_{SC}(R)$ であるような代入 $\sigma_0, \sigma_1, \ldots$ が存在するとき，依存対の列 $u_0^\# \to v_0^\#, u_1^\# \to v_1^\#, \ldots$ を R における静的依存鎖と呼ぶ．R の静的依存グラフは $SDP(R)$ がノード，$u_i^\# \to v_i^\#, u_j^\# \to v_j^\#$ が依存鎖であるような $u_i^\# \to v_i^\#$ から $u_j^\# \to v_j^\#$ への弧が存在する有向グラフである．依存グラフにおける強連結な部分グラフのノードの集合を静的再帰成分（static recursion component）と呼ぶ．R の静的再帰成分の集合を $SRC(R)$ と表記する．

例 3.8 $SRC(R_2)$ は次のようになる．

$$\left\{\begin{array}{l} \left\{sub^\#(s(X), s(Y)) \ \to\ sub^\#(X, Y)\right\} \\ \left\{div^\#(s(X), s(Y)) \ \to\ div^\#(sub(X, Y), s(Y))\right\} \\ \left\{\begin{array}{l} map^\#(\lambda x.F(x), cons(X, L)) \\ \qquad\qquad \to\ map^\#(\lambda x.F(x), L) \end{array}\right\} \end{array}\right\}$$

定義 3.9 (non-looping) 静的再帰成分 $C \in SRC(R)$ が non-looping であるとは，C の要素のみで構成され各 C の要素が無限回出現するような依存鎖が存在しないことである．

定理 3.10 R を PFP-HRS とする．無限 $SDP(R)$-鎖が存在しないならば R は停止する．

定理 3.11 R を $SDP(R)$ が有限な PFP-HRS とする．全ての再帰成分が non-looping ならば R は停止する．

次に，この定理に基づき停止性を証明する際に重要な役割をなす部分項基準と簡約化対の概念を紹介する．

定義 3.12 (部分項基準) R を PFP-HRS，$C \in SRC(R)$ とする．以下の条件を満たす D_R から空でない正整数の列への関数 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 $u^\# \to v^\# \in C$ で $u|_{\pi(top(u))} >_{sub} v|_{\pi(top(v))}$.

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

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

定義 3.13 (簡約化対) $\gtrsim$ を quasi order，$>$ を strict order とする．以下が成立するとき $(\gtrsim, >)$ が述語 P のもとで簡約化対であると定義する．

- 任意の項 s, t に対して $P(s, t) \wedge s \gtrsim t \Rightarrow \forall\sigma.s\sigma\downarrow\ \gtrsim t\sigma\downarrow$ かつ $P(s, t) \wedge s > t \Rightarrow \forall\sigma.s\sigma\downarrow\ > t\sigma\downarrow$.
- $\gtrsim, >$ が文脈に閉じている．
- $>$ が整礎である．
- $\gtrsim \circ > \subseteqq >$ または $> \circ \gtrsim \subseteqq >$.

ただし $\gtrsim$ において $\perp^\alpha\downarrow$ は任意の項 $t \in \mathcal{T}_\alpha$ に対して $t \gtrsim \perp\downarrow$ を満たすものとする．

定理 3.14 R を $SDP(R)$ が有限な PFP-HRS とする．すべての $C \in SRC(R)$ が以下のどちらかの性質を満たすとき R は停止する．

- C が部分項基準を満たす．
- $R \cup C \subseteqq \gtrsim$ かつ $C \cap > \neq \emptyset$ である簡約化対 $(\gtrsim, >)$ が存在．

$SRC(R_2)$ の一つ目と三つ目の再帰成分は部分項基準を満たす．しかし二つ目の再帰成分

$$\left\{div^\#(s(X), s(Y)) \ \to\ div^\#(sub(X, Y), s(Y))\right\}$$

は部分項基準を満たさないため適切な簡約化対を設計する必要がある．

4. 引数切り落とし法

前節で紹介したように，依存対法において簡約化対は重要である．簡約化対を設計するために Arts と Giesl は一階の項書換え系において引数切り落とし法を提案した [1]．これは草刈に

より高階関数を直接扱える STRS 上へ拡張された[4]. 本節ではこの手法を, STRS とは異なり λ 抽象が扱える HRS 上に拡張する.

既存の引数切り落とし法は型の構造を破壊してしまうという問題を持つ. この問題を解決するために $sub(X,Y)$ の Y を既存の手法のように切り落とし $sub(X)$ とするのではなく, Nat 型の $\perp$ 記号 $\perp^N$ で埋めて $sub(X,\perp^N)$ とすることによって解決する.

定義 4.1 (切り落とし関数) 切り落とし関数 π とは関数記号から正整数リストへの関数であり, $\perp^\alpha$ を型 α を持つ fresh な関数記号とする. 切り落とし関数 π を $\pi(\lambda\overline{x_m}.a(\overline{t_n})) = \lambda\overline{x_m}.a(\overline{t'_n})$ で項上に拡張する. ここで各 t'_i は以下で定められる.

$$t'_i = \begin{cases} \pi(t_i) & \text{if } i \in \pi(a) \text{ または } a \in \mathcal{V} \\ \perp^{type(t_i)}\downarrow & \text{otherwise.} \end{cases}$$

例 4.2 R_2 の div に関する再帰成分の両辺に $\pi(sub) = [1]$ となる切り落とし関数を適用すると次のようになる.

$$\left\{ div^\#(s(X), s(Y)) \ \rightarrow \ div^\#(sub(X,\perp), s(Y)) \right\}$$

引数切り落とし法は与えられた簡約化順序から簡約化対を生成する手法である. 簡約化順序とは, 代入と文脈に閉じている整礎な疑順序のことである. 簡約化対の生成は次のように行う.

定義 4.3 $\gtrsim$ を簡約化順序とする. このとき, 切り落とし関数 π を用いた項上の順序 $\gtrsim_\pi, >_\pi$ を以下のように定義する.

$$s \gtrsim_\pi t \overset{\text{def}}{\iff} \pi(s) \gtrsim \pi(t)$$

$$s >_\pi t \overset{\text{def}}{\iff} \pi(s) > \pi(t)$$

ここで $(\gtrsim_\pi, >_\pi)$ が簡約化対であることを示したい. 代入に関する条件以外は比較的容易に示すことができる.

補題 4.4 $\gtrsim$ が葉文脈に閉じているならば, $\gtrsim_\pi$ も葉文脈に閉じている.

証明 $s \gtrsim_\pi t \Rightarrow \forall C[].C[s] \gtrsim_\pi C[t]$ を葉文脈 $C[]$ における帰納法で証明する.

- $C[] \equiv \square$ のときは自明.
- $C[] \equiv a(\ldots, t_{i-1}, C'[], t_{i+1}, \ldots)$ のとき.
  - $i \notin \pi(a)$ の場合,
    $\pi(C[s]) \equiv a(\ldots, \pi(t_{i-1}), \perp, \pi(t_{i+1}), \ldots) \equiv \pi(C[t])$.
  - $i \in \pi(a)$ の場合, 帰納法の仮定より $\pi(C'[s]) \gtrsim \pi(C'[t])$. よって

$$\begin{aligned} \pi(C[s]) &\equiv a(\ldots, \pi(t_{i-1}), \pi(C'[s]), \pi(t_{i+1}), \ldots) \\ &\gtrsim a(\ldots, \pi(t_{i-1}), \pi(C'[t]), \pi(t_{i+1}), \ldots) \\ &\equiv \pi(C[t]) \end{aligned}$$

- $C[] \equiv \lambda x.C'[]$ のとき, 帰納法の仮定より $\pi(C'[s]) \gtrsim \pi(C'[t])$. よって
  $\pi(C[s]) \equiv \lambda x.\pi(C'[s]) \gtrsim \lambda x.\pi(C'[t]) \equiv \pi(C[t])$. □

補題 4.5 $>$ が整礎ならば, $>_\pi$ も整礎である.

証明 $>$ を整礎とする. 無限列 $t_0 >_\pi t_1 >_\pi t_2 >_\pi \cdots$ が存在すると仮定すると, 任意の i に対して $t_i >_\pi t_{i+1} \Leftrightarrow \pi(t_i) > \pi(t_{i+1})$ より, 無限列は $\pi(t_0) > \pi(t_1) > \pi(t_2) > \cdots$ となる. これは $>$ の整礎性に矛盾する. よって $>_\pi$ は整礎である. □

補題 4.6 項上の順序 $\gtrsim$ で定義される $\gtrsim_\pi, >_\pi$ に対して $>_\pi \circ \gtrsim_\pi \subseteq >_\pi$ が成立.

証明 $s >_\pi \circ \gtrsim_\pi t \Rightarrow s \gtrsim_\pi t \land s \not\lesssim_\pi t$ を示せばよい. $s >_\pi u \gtrsim_\pi t$ を満たす u に対し $s \gtrsim_\pi u \gtrsim_\pi t$. $\gtrsim_\pi$ の推移性より $s \gtrsim_\pi t$ は明らか. また $s \lesssim_\pi t$ と仮定すると $s \lesssim_\pi t \lesssim_\pi u$ であり $\gtrsim_\pi$ の推移性より $s \lesssim_\pi u$. これは $s >_\pi u \Longleftrightarrow s \gtrsim_\pi u \land s \not\lesssim_\pi u$ に反する. ゆえに $s \not\lesssim_\pi t$. □

残りは代入に関する条件を示すだけである.

代入 σ に対して切り落とし関数を適用した代入を $\sigma_\pi(x) = \pi(\sigma(x))$ で定義する. $\gtrsim_\pi$ が代入に関する条件を満たすためには, 全ての $u \to v \in R \cup SDP(R)$ に対し $\pi(u\sigma\downarrow) \sim \pi(u)\sigma_\pi\downarrow$ および $\pi(v)\sigma_\pi\downarrow \gtrsim \pi(v\sigma\downarrow)$ が成立すればよい. 残念ながらこれらが一般に成立するか否かは現状では判明していない. 本論文では, 規則の両辺が堅固および 2nd-order のいずれかの場合にはこれらが成立することを示す.

定義 4.7 (堅固) 項中の自由変数が, すべて $F\downarrow \equiv \lambda\overline{x_n}.F(\overline{x_n})$ の形で出現するとき, この項は堅固 (firmness) であるという.

定義 4.8 (2nd-order) 項に現れる高階変数の引数がすべて基本型であるとき, その項を 2nd-order であるという.

補題 4.9 項 t が堅固であるとき, 任意の代入 σ に対して $\pi(t\sigma\downarrow) \equiv \pi(t)\sigma_\pi\downarrow$ が成立する.

証明 t は堅固であるので t 中の自由変数はすべて $F\downarrow$ の形で出現する. ここで $FV(t) \supseteq Dom(\sigma)$ および $BV(t) \cap Dom(\sigma) = \emptyset$ を仮定する. さらに $S(t) = \{t\} \cup \bigcup_{u \in args(t)} S(u)$ として, 任意の $u \in S(t)$ に対し $\pi(u\sigma\downarrow) \equiv \pi(u)\sigma_\pi\downarrow$ が成立することを $|u|$ に関する帰納法で証明する.

- $u \equiv \lambda x_1 \ldots x_m.f(u_1, \ldots, u_n)$ のとき.
  - $m = 0$ のとき. $i \in \pi(f)$ ならば $u'_i \equiv u_i$, $i \notin \pi(f)$ ならば $u'_i \equiv \perp\downarrow$ であるような $u'_1, \ldots, u'_n$ を考える. このとき, 各 i で帰納法の仮定より $\pi(u'_i)\sigma_\pi\downarrow \equiv \pi(u'_i\sigma\downarrow)$. よって

$$\begin{aligned} \pi(u\sigma\downarrow) &\equiv \pi(f(u_1, \ldots, u_n)\sigma\downarrow) \\ &\equiv \pi(f(u_1\sigma\downarrow, \ldots, u_n\sigma\downarrow)) \\ &\equiv f(\pi(u'_1\sigma\downarrow), \ldots, \pi(u'_n\sigma\downarrow)) \\ &\equiv f(\pi(u'_1)\sigma_\pi\downarrow, \ldots, \pi(u'_n)\sigma_\pi\downarrow) \\ &\qquad (\because \text{I.H., 補題 4.4}) \\ &\equiv f(\pi(u'_1), \ldots, \pi(u'_n))\sigma_\pi\downarrow \end{aligned}$$

$$\equiv \pi(f(u_1,\ldots,u_n))\sigma_\pi{\downarrow}$$
$$\equiv \pi(u)\sigma_\pi{\downarrow}$$

– $m>0$ のとき．帰納法の仮定より

$$\begin{aligned}&\pi(\lambda x_1\ldots x_m.f(u_1,\ldots,u_n)\sigma{\downarrow})\\ \equiv\ &\lambda x_1\ldots x_m.\pi(f(u_1,\ldots,u_n)\sigma{\downarrow})\\ \equiv\ &\lambda x_1\ldots x_m.\pi(f(u_1,\ldots,u_n))\sigma_\pi{\downarrow}\\ \equiv\ &\pi(\lambda x_1\ldots x_m.f(u_1,\ldots,u_n))\sigma_\pi{\downarrow}\end{aligned}$$

- $u\equiv\lambda x_1\ldots x_m.F(u_1,\ldots,u_n)$ のとき．
  – $F\notin Dom(\sigma)$ のときは上記と同様．
  – $F\in Dom(\sigma)$ のとき．t は堅固なので $u\equiv\lambda x_1\ldots x_n.F(x_1\ldots x_n)$ であり $\sigma(F)\equiv s$ とすると $u\sigma{\downarrow}\equiv s$ が成立する．定義から $\sigma_\pi(F)\equiv\pi(s)$ であり，また $\pi(u)\equiv u$ となるため $\pi(u)\sigma_\pi{\downarrow}\equiv u\sigma_\pi{\downarrow}\equiv\pi(s)\equiv\pi(u\sigma{\downarrow})$． □

補題 4.10 基本型の項 $t_1,\ldots,t_n$，代入 σ に対して各 i で $\pi(t_i)\sigma_\pi{\downarrow}\equiv\pi(t_i\sigma{\downarrow})$ が成り立つとき，任意の項 v および代入 $\theta=\{y_i\mapsto t_i\sigma{\downarrow}\mid 1\leqq i\leqq n\}$，$\theta'=\{y_i\mapsto\pi(t_i)\sigma_\pi{\downarrow}\mid 1\leqq i\leqq n\}$ に対して $v\theta'{\downarrow}\equiv v\theta_\pi{\downarrow}$ が成立．

証明 項 v 上の構造帰納法で示す．

- $v\equiv\lambda x.v'$ のとき．補題 4.4 および帰納法の仮定より
$$v\theta'{\downarrow}\ \equiv\ \lambda x.(v'\theta'{\downarrow})\ \equiv\ \lambda x.(v'\theta_\pi{\downarrow})\ \equiv\ v\theta_\pi{\downarrow}$$
- $v\equiv f(v_1,\ldots,v_m)$ のとき．補題 4.4 と帰納法の仮定より
$$v\theta'{\downarrow}\equiv f(v_1\theta'{\downarrow},\ldots,v_m\theta'{\downarrow})\equiv f(v_1\theta_\pi{\downarrow},\ldots,v_m\theta_\pi{\downarrow})\equiv v\theta_\pi{\downarrow}$$
- $v\equiv F(v_1,\ldots,v_m)$ のとき．$F\notin Dom(\theta)$ ならば $v\equiv f(v_1,\ldots,v_m)$ の場合と同様に示される．また $F\in Dom(\theta)$ ならば，ある i が存在して $F\equiv y_i$．$type(y_i)\in\mathcal{B}$ であるため，$v\equiv y_i$．ゆえに，帰納法の仮定から
$$v\theta'{\downarrow}\ \equiv\ \pi(t_i)\sigma_\pi{\downarrow}\ \equiv\ \pi(t_i\sigma{\downarrow})\ \equiv\ v\theta_\pi$$

よって題意は示された． □

補題 4.11 項 t が 2nd-order であるとき，任意の代入 σ に対して $\pi(t\sigma{\downarrow})\equiv\pi(t)\sigma_\pi{\downarrow}$ が成立する．

証明 preterm $t\sigma$ 上の $(\underset{\beta}{\rightarrow}\cup>_{sub})$ に関する帰納法で証明する．

- $t\equiv\lambda x.t'$ のとき．$\pi(t)\sigma_\pi{\downarrow}\equiv\pi(\lambda x.t')\sigma_\pi{\downarrow}\equiv\lambda x.\pi(t')\sigma_\pi{\downarrow}$ である．帰納法の仮定より $\lambda x.\pi(t')\sigma_\pi{\downarrow}\equiv\lambda x.\pi(t'\sigma{\downarrow})\equiv\pi(\lambda x.t'\sigma{\downarrow})$．よって $\pi(t)\sigma_\pi{\downarrow}\equiv\pi(t\sigma{\downarrow})$ となる．
- $t\equiv f(t_1,\ldots,t_n)$ のとき．$i\in\pi(f)$ ならば $t'_i\equiv t_i$，$i\notin\pi(f)$ ならば $t'_i\equiv\bot{\downarrow}$ であるような $t'_1,\ldots,t'_n$ を考える．このとき，各 i に対し $t\sigma>_{sub}t_i\sigma$ なので，帰納法の仮定より $\pi(t_i)\sigma_\pi{\downarrow}\equiv\pi(t_i\sigma{\downarrow})$．また $t'_i\equiv\bot{\downarrow}$ ならば $\pi(\bot{\downarrow})\sigma_\pi{\downarrow}\equiv\bot{\downarrow}\equiv\pi(\bot\sigma{\downarrow})$ なので
$$\begin{aligned}&\pi(f(t_1,\ldots,t_n))\sigma_\pi{\downarrow}\\ \equiv\ &f(\pi(t'_1),\ldots,\pi(t'_n))\sigma_\pi{\downarrow}\\ \equiv\ &f(\pi(t'_1)\sigma_\pi{\downarrow},\ldots,\pi(t'_n)\sigma_\pi{\downarrow})\\ \equiv\ &f(\pi(t'_1\sigma{\downarrow}),\ldots,\pi(t'_n\sigma{\downarrow}))\quad(\because\text{I.H., 補題 4.4})\\ \equiv\ &\pi(f(t_1\sigma{\downarrow},\ldots,t_n\sigma{\downarrow}))\\ \equiv\ &\pi(f(t_1,\ldots,t_n)\sigma{\downarrow})\end{aligned}$$
- $t\equiv F(t_1,\ldots,t_n)$ かつ $F\notin Dom(\sigma)$ のとき．
$$\begin{aligned}&\pi(F(t_1,\ldots,t_n))\sigma_\pi{\downarrow}\\ \equiv\ &F(\pi(t_1),\ldots,\pi(t_n))\sigma_\pi{\downarrow}\\ \equiv\ &F(\pi(t_1)\sigma_\pi{\downarrow},\ldots,\pi(t_n)\sigma_\pi{\downarrow})\end{aligned}$$
このとき，各 t_i に対し $t\sigma>_{sub}t_i\sigma$ なので，帰納法の仮定より $\pi(t_i)\sigma_\pi{\downarrow}\equiv\pi(t_i\sigma{\downarrow})$．よって
$$\begin{aligned}&F(\pi(t_1)\sigma_\pi{\downarrow},\ldots,\pi(t_n)\sigma_\pi{\downarrow})\\ \equiv\ &F(\pi(t_1\sigma{\downarrow}),\ldots,\pi(t_n\sigma{\downarrow}))\quad(\because\text{I.H., 補題 4.4})\\ \equiv\ &\pi(F(t_1\sigma{\downarrow},\ldots,t_n\sigma{\downarrow}))\\ \equiv\ &\pi(F(t_1,\ldots,t_n)\sigma{\downarrow})\end{aligned}$$
- $t\equiv F(t_1,\ldots,t_n)$ かつ $F\in Dom(\sigma)$ のとき．$\sigma(F)\equiv\lambda y_1\ldots y_n.a(u_1,\ldots,u_k)$ かつ $\theta=\{y_i\mapsto t_i\sigma{\downarrow}\mid 1\leqq i\leqq n\}$，$\theta'=\{y_i\mapsto\pi(t_i)\sigma_\pi{\downarrow}\mid 1\leqq i\leqq n\}$ とする．このとき，帰納法の仮定より任意の i に対して $\pi(t_i)\sigma_\pi{\downarrow}\equiv\pi(t_i\sigma{\downarrow})$ が成り立つため，補題 4.10 より，すべての u_i に対して $u_i\theta'{\downarrow}\equiv u_i\theta_\pi{\downarrow}$ が成立する．
  – $a\in\mathcal{F}$ のとき $i\in\pi(a)$ ならば $u'_i\equiv u_i$，$i\notin\pi(a)$ ならば $u'_i\equiv\bot{\downarrow}$ であるような $u'_1,\ldots,u'_k$ を考える．このとき，各 i に対し
$$\begin{aligned}t\sigma\ &\equiv\ F(t_1,\ldots,t_n)\sigma\\ &\equiv\ (\lambda y_1\ldots y_n.a(u_1,\ldots,u_k))(t_1\sigma,\ldots,t_n\sigma)\\ &\overset{*}{\underset{\beta}{\rightarrow}}\ a(u_1\theta,\ldots,u_k\theta)\\ &>_{sub}\ u_i\theta\end{aligned}$$
なので，帰納法の仮定より $\pi(u_i)\theta_\pi{\downarrow}\equiv\pi(u_i\theta{\downarrow})$．よって
$$\begin{aligned}&\pi(F(t_1,\ldots,t_n))\sigma_\pi{\downarrow}\\ \equiv\ &F(\pi(t_1),\ldots,\pi(t_n))\sigma_\pi{\downarrow}\\ \equiv\ &\pi(\lambda\overline{y_n}.a(u_1,\ldots,u_k))(\pi(t_1)\sigma_\pi,\ldots,\pi(t_n)\sigma_\pi){\downarrow}\\ \equiv\ &(\lambda\overline{y_n}.a(\pi(u'_1),\ldots,\pi(u'_k)))(\pi(t_1)\sigma_\pi,\ldots,\pi(t_n)\sigma_\pi){\downarrow}\\ \equiv\ &a(\pi(u'_1)\theta'{\downarrow},\ldots,\pi(u'_k)\theta'{\downarrow})\\ \equiv\ &a(\pi(u'_1)\theta_\pi{\downarrow},\ldots,\pi(u'_k)\theta_\pi{\downarrow})\quad(\because\text{補題 4.10})\\ \equiv\ &a(\pi(u'_1\theta{\downarrow}),\ldots,\pi(u'_k\theta{\downarrow}))\quad(\because\text{I.H., 補題 4.4})\\ \equiv\ &\pi(a(u_1\theta{\downarrow},\ldots,u_k\theta{\downarrow}))\end{aligned}$$

$$\equiv \pi((\lambda\overline{y_n}.a(u_1,\dots,u_k))(t_1\sigma,\dots,t_n\sigma)\!\downarrow)$$
$$\equiv \pi(F(t_1,\dots,t_n)\sigma\!\downarrow)$$

– $a \in \mathcal{V} \wedge a \notin \{y_1,\dots,y_n\}$ のとき.

$$\pi(F(t_1,\dots,t_n))\sigma_\pi\!\downarrow$$
$$\equiv F(\pi(t_1),\dots,\pi(t_n))\sigma_\pi\!\downarrow$$
$$\equiv \pi(\lambda\overline{y_n}.a(u_1,\dots,u_k))(\pi(t_1)\sigma_\pi,\dots,\pi(t_n)\sigma_\pi)\!\downarrow$$
$$\equiv (\lambda\overline{y_n}.a(\pi(u_1),\dots,\pi(u_k)))(\pi(t_1)\sigma_\pi,\dots,\pi(t_n)\sigma_\pi)\!\downarrow$$
$$\equiv a(\pi(u_1)\theta'\!\downarrow,\dots,\pi(u_k)\theta'\!\downarrow)$$
$$\equiv a(\pi(u_1)\theta_\pi\!\downarrow,\dots,\pi(u_k)\theta_\pi\!\downarrow) \quad (\because \text{補題 4.10})$$
$$\equiv a(\pi(u_1\theta\!\downarrow),\dots,\pi(u_k\theta\!\downarrow)) \quad (\because \text{I.H., 補題 4.4})$$
$$\equiv \pi(a(u_1\theta\!\downarrow,\dots,u_k\theta\!\downarrow))$$
$$\equiv \pi((\lambda\overline{y_n}.a(u_1,\dots,u_k))(t_1\sigma,\dots,t_n\sigma)\!\downarrow)$$
$$\equiv \pi(F(t_1,\dots,t_n)\sigma\!\downarrow)$$

– $a \in \{y_1,\dots,y_n\}$ のとき. $a = y_j$ とすると t の満たす条件 (2nd-order) より $type(y_j) \in \mathcal{B}$ なので $\sigma(F) \equiv \lambda y_1 \dots y_n.y_j$.

$$\pi(F(t_1,\dots,t_n))\sigma_\pi\!\downarrow$$
$$\equiv F(\pi(t_1),\dots,\pi(t_n))\sigma_\pi\!\downarrow$$
$$\equiv (\lambda y_1\dots y_n.y_j)(\pi(t_1)\sigma_\pi,\dots,\pi(t_n)\sigma_\pi)\!\downarrow$$
$$\equiv \pi(t_j)\sigma_\pi\!\downarrow$$
$$\equiv \pi(t_j\sigma\!\downarrow) \quad (\because \text{I.H., 補題 4.4})$$
$$\equiv (\lambda y_1\dots y_n.y_j)(\pi(t_1\sigma\!\downarrow),\dots,\pi(t_n\sigma\!\downarrow))\!\downarrow$$
$$\equiv \pi(F(t_1,\dots,t_n)\sigma\!\downarrow)$$

□

定理 4.12 $\gtrsim$ が簡約化順序であるとき, $(\gtrsim_\pi, >_\pi)$ は述語 P ($P(s,t) \stackrel{\text{def}}{\Longleftrightarrow} s,t$ が堅固または 2nd-order) のもとで簡約化対である.

証明 補題 4.4, 4.5, 4.6, 4.9, 4.11 より明らか.

5. おわりに

本論文では HRS 上の引数切り落とし法を提案した. 本手法は既存の引数切り落とし法における型の構造が破壊されるという問題を解決している. また, 各規則の両辺が堅固または 2nd-order であれば本手法が健全であることを示した. 残念ながら一般に適用できるか否かは現時点ではわかっていない. 文献 [4] の STRS における引数切り落とし法は規則の左辺が堅固であることを要求した. また, 文献 [5] ではさらに複雑な制限を要求した. STRS と HRS は体系が異なるため単純に比較することはできないが, 何らかの制限が必要となることが予測される. これは今後の課題である.

謝辞 本研究は一部, 科研費#15500007, #16650005, #17700009 の補助を受けている.

文献

[1] Arts.T., Giesl,J., 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] 磯谷泰巨, 草刈圭一朗, 酒井正彦, 坂部俊樹, 西田直樹, 強計算依存対法による高階書換え系の停止性証明, 電子情報通信学会技術研究報告 (SS2006-6), Vol.106, No.15, pp.31–36, 2006.
[4] Kusakari,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.
[5] Kusakari,K., Higher-Order Path Orders based on Computability, IEICE Transactions on Information and Systems, Vol.E87-D, No.2, pp.352-359, Feb 2004.
[6] Kusakari,K., Isogai,Y., Sakai,M., Sakabe,T., Nishida,N., Static Dependency Pair Method for Proving Termination of Higher-Order Rewriting Systems, Tech. Rep. of IEICE, to appear.
[7] Kusakari,K., Sakai,M., Enhancing Dependency Pair Method using Strong Computability in Simply-Typed Term Rewriting Systems, Applicable Algebra in Engineering, Communication and Computing, to appear
[8] Nipkow,T., Higher-order critical pairs, Proc. 6th Annual IEEE Symposium on Logic in Computer Science, pp.342–349, 1991.