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.

単純型項書換え系上の依存対法における 実効規則と直積型項へのラベル付け

櫻井　敬大†　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
〒 466-8603 名古屋市千種区不老町
E-mail: †tsakurai@trs.cm.is.nagoya-u.ac.jp, ††{kusakari,sakai,sakabe,nishida}@is.nagoya-u.ac.jp

あらまし　単純型項書換え系は関数プログラムの計算モデルであり，その重要な性質に停止性がある．単純型項書換え系の停止性証明法として我々は強計算依存対法を提案した．本論文ではこの証明法の実用性の向上を目指し，２つのアプローチを与える．１つは直積型項へのラベル付けによる順序付けのし易さの向上，もう１つは実効規則の概念の導入による制約の削減である．

キーワード　単純型項書換え系，停止性，ラベル付け，実効規則

Usable Rules and Labeling Product-Typed Term for Dependency Pair Method in Simply-Typed Term Rewriting Systems

Takahiro SAKURAI†, 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: †tsakurai@trs.cm.is.nagoya-u.ac.jp, ††{kusakari,sakai,sakabe,nishida}@is.nagoya-u.ac.jp

Abstract　Simply-typed term rewriting system is a computational model of functional programs, and termination is one of the important properties. We proposed dependency pair method based on strong computability as proving method for termination of simply-typed term rewriting system. In this paper, we enhance this method by two approach. One is reducing constraints by usable rules, the another is labeling product-typed terms.

Key words　Symply-typed Term Rewriting System, Termination, Labeling, Usable Rules

1. はじめに

単純型項書換え系（simply-typed term rewiting system; STRS）[1] は，理論的に取り扱いがし易いように制限を与えながらも関数型プログラミング言語に操作的意味を与えるのに十分な表現力を持つ高階の書換え系である．

STRS の停止性証明法として，我々は強計算依存対法を提案した [2]．これは STRS から再帰定義部分を抽出し，再帰が無限に起こらない事を証明することで停止性を証明する強力な手法である．本稿ではこの強計算依存対法を２つのアプローチにより更に強力なものとする．

強計算依存対法はそれぞれの再帰成分を分析する際に，STRS の全ての規則も同時に分析する．つまり，再帰の際に呼び出される事のない規則にも注目せねばならないのである．これに対し１つ目のアプローチとして強計算依存対法において再帰の際に呼び出す可能性のある規則（実効規則）にのみ注目すればよい事を示す．

２つ目のアプローチとして直積型表現に対応した強計算依存対法を提案する．具体的には直積型項を表現する特別な関数記号 tp に文脈に依存したラベル付けを行うことにより，tp の区別を行う．

以上２つのアプローチにより，強計算依存対法をより強力なものへとする．

2. 準　　備

2.1 単純型項書換え系

単純型項書換え系（STRS）は文献 [1] で提案された．本節では文献 [2], [4] に基づき，論文中で必要となる諸概念を与える．

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 $\mathcal{T}(\Sigma, \mathcal{V})$ は次のように帰納的に定義される; $a \in \Sigma \cup \mathcal{V}$ かつ $t_1, \ldots, t_n \in \mathcal{T}(\Sigma, \mathcal{V})$ ならば $a[t_1, \ldots, t_n] \in \mathcal{T}(\Sigma, \mathcal{V})$. 項 $a[]$ は単に a と記す. 2 つの項 s と t が構文的に等しいことを $s \equiv t$ で表す. また, $s \equiv a[s_1, \ldots, s_n]$ とするとき, $s[t_1, \ldots, t_m]$ と書いて項 $a[s_1, \ldots, s_n, t_1, \ldots, t_m]$ を表す. 項 t に出現する全ての変数の集合を $Var(t)$ で表す. 項 $t \equiv a[t_1, \ldots, t_n]$ の根の位置の記号 a 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, すなわち $\theta(a[t_1, \ldots, t_n])$ は, $a \in \Sigma$ のときには $a[\theta(t_1), \ldots, \theta(t_n)]$ で, $a \in \mathcal{V}$ かつ $\theta(a) = a'[t'_1, \ldots, t'_k]$ のときには $a'[t'_1, \ldots, t'_k, \theta(t_1), \ldots, \theta(t_n)]$ で定義される. $\theta(t)$ 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 $\square$ が, 一箇所だけ出現する項である. また, 葉の位置に $\square$ が出現する文脈を**葉文脈**, 根の位置の場合は**根文脈**と呼ぶ. 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. 項 t' が項 t の**部分項**であるとは, ある葉文脈 $C[\,]$ が存在して $t \equiv C[t']$ となることである. 項 t の部分項全体を $Sub(t)$ で記す.

空でない**基本型**の集合を $\mathcal{B}$ で表す. $\mathcal{B}$ から生成される**単純型**の集合 $\mathcal{S}$ は, 型構成子 $\to$, $\times$ を用いて $\mathcal{S} ::= \mathcal{B} \mid (\mathcal{S} \to \mathcal{S}) \mid (\mathcal{S} \times \cdots \times \mathcal{S})$ として定義される. $\to$ は右結合性を持ち, $\times$ よりも優先度が低いとし, () を適宜省略する. 本論文では, 単純型を単に型と呼ぶ事がある. $\alpha_1 \times \cdots \times \alpha_n$ の形の単純型を**直積型**と呼ぶ. 直積型の項を表現する為**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 $tp[t_1, \ldots, t_n]$ を $(t_1, \ldots, t_n)$ 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 $\tau(tp)$ は未定義とする. 項 $t \equiv a[t_1, \ldots, t_n] \in \mathcal{T}(\Sigma, \mathcal{V})$ が単純型 α を持つとは, 各 $t_i (i = 1, \ldots, n)$ が単純型 α_i を持ち, $a = tp$ のとき $\tau(a) = \alpha_1 \times \cdots \times \alpha_n (n \geqq 2)$, そうでないとき $\tau(a) = \alpha_1 \to \cdots \to \alpha_n \to \alpha$ となることである. 単純型 α が単純型 β の *suffix* であるとは, ある $\alpha_1, \ldots, \alpha_n$ に対し, $\beta = \alpha_1 \to \cdots \to \alpha_n \to \alpha$ となることであり, $\alpha \sqsubseteq \beta$ で記す. 単純型を持つ項を**単純型項**と呼ぶ. 全ての単純型項からなる集合を $\mathcal{T}_\tau(\Sigma, \mathcal{V})$ で記す. なお本論文中では, 代入や文脈は型の整合性を崩さないもののみを扱う. すなわち, 単純型項 t に対し $t\theta$ や $C[t]$ は共に単純型を持つとする.

単純型書換え規則とは $root(l) \in \Sigma \backslash \{tp\}$, $Var(l) \supseteq Var(r)$, $\tau(l) = \tau(r)$ の条件を満たす単純型項 l, r の対 (l, r) であり, $l \to r$ と記す. **単純型項書換え系** (STRS) とは単純型書換え規則の集合である. STRS R において単純型項 s が t に書換えられるとは, ある規則 $l \to r \in R$, 代入 θ, 文脈 $C[\,]$ が存在し $s \equiv C[l\theta]$ かつ $t \equiv C[r\theta]$ となることである. このとき, $s \to_R t$, または $s \to t$ と記す. STRS R が**有限分岐**であるとは, R における全ての項 t において $\{t' \mid t \underset{R}{\to} t'\}$ が有限であることである.

項 t が STRS R において**停止性**を持つとは ($SN(R, t)$ と記す), $t \to_R t' \to_R \cdots$ のような無限列がないことである. R が停止性を持つとは ($SN(R)$ と記す), 任意の項 t に対して $SN(R, t)$ であることである. STRS と項の直積上での述語 P に対し, $\mathcal{T}_P^{args}(R) = \{t \mid \forall u \in args(t). P(R, u)\}$ を定義する.

切り落とし関数 π は, 各 $f \in \Sigma$ ($\tau(f) = \alpha_1 \to \cdots \to \alpha_n \to \beta$) を $i_1 < \cdots < i_m \leqq n$ となる正整数のリスト $[i_1, \ldots, i_m]$ に割り当てる関数である. $i_j \leqq k$ を満たす $\pi(f)$ の最大部分リスト $[i_1, \ldots, i_j]$ を $\pi(f)^{\leqq k}$ で表す. 項 $a[t_1, \ldots, t_n]$ に対し $\pi(a[t_1, \ldots, t_n])$ は $a \in \mathcal{V}$ ならば $a[\pi(t_1), \ldots, \pi(t_n)]$, $a \in \Sigma$ ならば $\pi(a)^{\leqq n} = [i_1, \ldots, i_m]$ のとき $a[\pi(t_{i_1}), \ldots, \pi(t_{i_m})]$ で定義される. 項 t が**堅固**であるとは, t 中の変数が全て葉の位置で出現していることである. 狭義の半順序 $>$ から, $s \gtrsim_\pi t$ を $\pi(s) \geqq \pi(t)$ で定義する. 擬順序 $\gtrsim$ が**弱簡約化順序**であるとは, $\gtrsim$ が文脈と代入に閉じ, $\gnsim$ が整礎かつ代入に閉じていることである. 尚, STRS において全ての左辺項が堅固であるとき, 任意の簡約化順序 $>$(文脈と代入に閉じた整礎な狭義の半順序) に対して $\gtrsim_\pi$ は弱簡約化順序になる [4].

定義 2.1 $l \to r$ を $\tau(l) = \alpha_1 \to \cdots \to \alpha_n \to \alpha$, $\alpha \in \mathcal{S}^{nfun}$ である規則とする. このとき, $(l \to r)^{ex}$ を $\{l \to r, l[z_1] \to r[z_1], \ldots, l[z_1, \ldots, z_n] \to r[z_1, \ldots, z_n]\}$, $(l \to r)^{ex\uparrow}$ を $l[z_1, \ldots, z_n] \to r[z_1, \ldots, z_n]$ で定義する. ここで, $z_1, \ldots, z_n$ はフレッシュな変数であり, $\forall i. \tau(z_i) = \alpha_i$ である. また, $R^{ex} = \bigcup_{l \to r \in R} (l \to r)^{ex}$ とする.

補題 2.2 STRS R を考える. $s \underset{R}{\to} t$ ならばある規則 $l \to r \in R^{ex}$, 葉文脈 $C[]$, 代入 θ が存在し, $s \equiv C[l\theta]$ かつ $t \equiv C[r\theta]$.

2.2 強計算依存対法

この節では文献 [2] で提案した STRS の停止性証明法の 1 つである強計算依存対法を紹介する.

項 t の**ポジション**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. $t \equiv a[t_1, \ldots, t_n]$ のとき $Pos(t) = \{\epsilon\} \cup \{ip' \mid i \leqq i \leqq n, p' \in Pos(t_i)\}$. ポジション上の順序 $\succ$ について $p \succ q$ であるとは, ある w について $pw = q$ となることである. また, ポジション p における記号を $(t)_p$ とし, t のポジション p における部分項を $t|_p$ で記す.

R を STRS とする. R の規則 $l \to r$ の左辺 l の根の位置に出現する関数記号 ($root(l)$) を**被定義記号**と呼び, R の被定義記号全体からなる集合を $\mathcal{D}_R$ 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, R の構成子全体からなる集合を $\mathcal{C}_R$ で記す.

各関数記号 $f \in \Sigma$ に対し, **印付記号** $f^\#$ を用意する. また, **印付項** $t^\#$ で t の root 記号を対応する印付記号で置き換えた項を表す. ここで, $root(t) \in \mathcal{V}$ の場合には $t^\# \equiv t$ とする.

P を項の対の集合, $\gg$ を 2 項関係, T を項の集合とする. P 中の対の列 $\langle u_0, v_0 \rangle \langle u_1, v_1 \rangle \ldots$ が T 上の $\langle P, \gg \rangle$-鎖であるとは, $u_i\theta_i \in T, v_i\theta_i \in T$ となる代入 $\theta_0, \theta_1 \ldots$ が存在し, 各 i で $v_i\theta_i \gg^* u_{i+1}\theta_{i+1}$ となることである. T 上の $\langle P, \gg \rangle$-依存グラフとは有向グラフであり, そのノードは P の要素, また, e から e' への辺が存在するとは ee' が T 上の $\langle P, \gg \rangle$-鎖であることである. T 上の $\langle P, \gg \rangle$-再帰成分とは T 上の $\langle P, \gg \rangle$-依存グラフの強連結部分グラフのノードの集合である.

項 u が項 t の**拡張部分項**であるとは, ある文脈 $C[]$ が存在して $t \equiv C[u]$ となることである. u が t の拡張部分項であるとき

$t \geqq_{esub} u$ で表し，$>_{esub}$ を $\geqq \setminus \equiv$ とする．

定義 2.3 STRS R, 印付項の対の集合を C とする．C が**部分項基準**を満たすとは，全ての $\langle u^\#, v^\# \rangle \in C, root(u) \notin \mathcal{V}, root(v) \notin \mathcal{V}$ について D_R から空でない正整数への関数 π が存在し，(1) または (2) を満たすことである．

（１） ある $\langle u^\#, v^\# \rangle$ について $u|_{\pi(root(u))} >_{esub} v|_{\pi(root(v))}$

（２） 全ての $\langle u^\#, v^\# \rangle$ が以下を満たす

- $u|_{\pi(root(u))} \geqq_{esub} v|_{\pi(root(v))}$
- 全ての $p \prec \pi(root(u))$ について $(u)_p \notin \mathcal{V}$
- $q \neq \epsilon \Rightarrow$ 全ての $q \prec \pi(root(v))$ について $(v)_q \in C_R$

定義 2.4 STRS R に対し，$upt(R)$ を以下で定義．$\alpha \in upt(R)$ とは，α が直積型かつ $\tau(z) = \alpha$ となる規則 $l \to r \in R$, 変数 $z \in Var(R)$ が存在することである．STRS R, 項 l に対し，**拡張引数**の集合 e_args を以下で定義．$u \in e\_args(R,l)$ であるとは，$u \in args(l)$，または $(\ldots,u,\ldots) \in e\_args(R,l)$ かつ $\tau((\ldots,u,\ldots)) \notin upt(R)$ となることである．STRS R, 項 l に対し，**安全部分項**の集合 $safe(R,l)$ を $e\_args(l)$ と $\{u \in Sub(l) \mid u \not\equiv l, \tau(u) \in \mathcal{B} \cup upt(R)\}$ の和集合とする．

定義 2.5 項 t が R において**強計算性**を持つ（$SC(R,t)$ と記す）とは，以下を満たすことである．

(1) $\tau(t) \in \mathcal{B} \cup upt(R)$ の場合，$SN(R,t)$

(2) $\tau(t) = \alpha_1 \times \cdots \times \alpha_n$ かつ $\tau(t) \notin upt(R)$ の場合，$SN(R,t)$ かつ $t \xrightarrow[R]{*} tp[t_1,\ldots,t_n]$ となる各 t_i に対して $SC(t_i)$

(3) $\tau(t) = \alpha \to \beta$ の場合，$\forall u.(SC(R,u) \land \tau(u) = \alpha \Rightarrow SC(R,t[u]))$

定義 2.6 STRS R を考える．被定義記号 a，葉文脈 $C[]$ を用いて R^{ex} 中の規則が $l \to C[a[r_1,\ldots,r_m]]$ と表現でき，全ての $k \leqq m$ について $a[r_1,\ldots,r_k] \notin safe(R,l)$ を満たすとき，$\langle l^\#, a^\#[r_1,\ldots,r_m] \rangle$ を R の SC **依存対**と言う．また，R の SC 依存対の集合を $DP_{SC}(R)$ と記す．そして $\mathcal{T}_{SC}^{args}(R)$ 上の $\langle DP_{SC}(R), \to_R \rangle$-再帰成分の全てからなる集合を $RC_{SC}(R)$ で記す．

定義 2.7 STRS R が**直接関数渡し**であるとは，$a \in \mathcal{V}$ としたとき，全ての葉文脈 $C[]$，規則 $l \to C[a[r_1,\ldots,r_m]] \in R$ について，$a[r_1,\ldots,r_k] \in safe(R,l)$ となる $k \leqq m$ が存在することである．直接関数渡しである STRS を PFP-STRS と呼ぶ．

定理 2.8 全ての再帰成分 $C \in RC_{SC}(R)$ が，以下のいづれかを満たす STRS R は停止性を持つ．

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

例 2.9 以下の STRS を考える．

$$R_{mul} = \left\{ \begin{array}{rcl} add[(0,y)] & \to & y \\ add[(s[x],y)] & \to & s[add[(x,y)]] \\ mul[(0,y)] & \to & y \\ mul[(s[x],y)] & \to & add[(mul[(x,y)],y)] \end{array} \right.$$

$$R_{sub} = \left\{ \begin{array}{rcl} sub[(x,0)] & \to & x \\ sub[(0,x)] & \to & 0 \\ sub[(s[x],s[y])] & \to & sub[(x,y)] \end{array} \right.$$

R_{mul}, R_{sub} の再帰成分は以下のようになる．

$$RC_{SC}(R_{mul}) = \left\{ \begin{array}{rcl} add^\#[(s[x],y)] & \to & add^\#[(x,y)] \\ mul^\#[(s[x],y)] & \to & mul^\#[(x,y)] \end{array} \right.$$

$$RC_{SC}(R_{sub}) = \left\{ \begin{array}{rcl} subl^\#[(s[x],s[y])] & \to & subl^\#[(x,y)] \end{array} \right.$$

$R_1 = R_{mul} \cup R_{sub}$ は部分項基準のみにより，その停止性が証明可能である．

3. 実効規則の導入による強計算依存対法の改良

STRS の停止性証明法として，我々は強計算依存対法を提案した．これは依存対から再帰成分を取り出し，規則両辺と再帰成分の両辺それぞれで順序をつけて停止性を示すものである．これまで停止性を示すことができなかった STRS も証明ができるようになった強力な手法であるが，以下の R_2 は停止性を持つにもかかわらずその停止性は示すことができない．

$$R_{div} = \left\{ \begin{array}{rcl} div[(0,x)] & \to & 0 \\ div[(s[x],s[y])] & \to & s[div[sub[(x,y),s[y]]]] \end{array} \right.$$

$$R_2 = R_{sub} \cup R_{div} \cup \left\{ \begin{array}{rcl} foldl[(f,z,nil)] & \to & z \\ foldl[(f,z,cons[(x,xs)])] & \to & foldl[(f,f[(z,x)],xs)] \end{array} \right\}$$

R_2 の再帰成分 $RC_{SC}(R_2)$ は，以下のようになる．

$$RC_{SC}(R_2) = RC_{SC}(R_{sub}) \cup \left\{ \begin{array}{l} \{\langle div^\#[(s[x],s[y])], div^\#[(sub[(x,y)],s[y])] \rangle\} \\ \{\langle foldl^\#[f,z,cons[(x,xs)]], foldl^\#[f,f[(z,x)],xs] \rangle\} \end{array} \right\}$$

R_2 は停止性を持つが，定理 2.8 により R_2 の停止性を示すにあたって問題なのは div についての再帰成分である．この再帰成分は部分項基準を満たさない．よって順序付けの出来る簡約化順序の存在を示すことが制約となるが，その制約の１つに $foldl[(f,z,cons[(x,xs)])] \gtrsim foldl[(f,f[(z,x)],xs)]$ への順序付けがある．この規則は div の再帰には全く関係のない制約である．そしてこの制約が R_2 の停止性を示すことが出来ない原因である．そこで，全ての規則ではなく，各再帰成分が呼び出す規則のみ，順序をつければ良い事を示す．尚，本手法は文献 [6] で提案された実効規則の概念を一階から高階に拡張したものである．まず，対の集合に対して実効規則 (usable rules) を以下で定義する．

定義 3.1 $Sub_{\mathcal{V}}^{int}(t) = \{t' \in Sub(t) \mid root(t') \in \mathcal{V}, args(t') \neq [\,]\}$ とする．項の対 $\langle u,v \rangle$ に対し，STRS R の部分集合 $\mathcal{U}'(\langle u,v \rangle)$ を $l \to r \in \mathcal{U}'(\langle u,v \rangle)$ により定義，以下の一つを満たすとする．集合 $\mathcal{U}(\langle u,v \rangle)$ を $\mathcal{U}'(\langle u,v \rangle) \subseteqq \mathcal{U}(\langle u,v \rangle)$ かつ $\mathcal{U}(\langle l,r \rangle) \subseteqq \mathcal{U}(\langle u,v \rangle)$（ただし $l' \to r' \in \mathcal{U}(\langle u,v \rangle)$ かつ $l \to r \in (l' \to r')^{ex}$）を満たす最小の集合として定義する．項の対の集合 C に対し，**実効規則** $\mathcal{U}(C)$ を $\bigcup_{\langle u,v \rangle \in C} \mathcal{U}(\langle u,v \rangle)$ で定義する．

R_2 の div の再帰成分の実効規則は R_{sub} となる．これは $\pi(tp) = [1]$ とした再帰経路順序 [1] で順序付けが可能であり，R_2 の停止性を示すことが出来る．

前節の定理 2.8 に実効規則から導かれる次節の定理 3.8 を導入すると，以上の議論をを定式化した次の定理が成立．

定理 3.2 全ての $C \in RC_{SC}(R)$ について以下の一つを満たす STRS R は停止性を持つ．

- C が部分項基準を満たす.
- ある簡約化対 $(\gtrsim, >)$ が存在し，$C_e \cup \mathcal{U}(C) \cup C \subseteqq\ \gtrsim$, $C \cap > \neq \emptyset$.

3.1 証　　明

本節では R を有限分岐 STRS，C を印付項の対の集合と固定して議論する．また，Δ を以下を満たす項 t の集合とする．$t \in \Delta$ iff $root(t) = root(l)$ and $\tau(t) \sqsubseteq \tau(l)$ for some $l \to r \in R \setminus \mathcal{U}(C)$

補題 3.3 各規則 $l \to r \in C \cup \mathcal{U}(C)^{ex}$，代入 θ について，以下の性質が成立する．

(1) $root(v) \in \Sigma$ である全ての $v \in Sub(r)$ について $v\theta \notin \Delta$

(2) 全ての $v \in Sub^{int}_{\mathcal{V}}(r)$ について $v\theta \notin \Delta$

(3) $root(u) \in Var(r)$ である全ての $u \in Sub^{int}_{\mathcal{V}}(l)$ について $root(u)\theta \notin \Delta$

証明 定義 *3.1* より明らかである． □

定義 3.4 単純型 $\alpha \in \mathcal{S}$ に対して，$\tau(\bot_\alpha) = \alpha$，$\tau(\mathsf{c}_\alpha) = \alpha \to \alpha \to \alpha$ であるフレッシュな関数記号 $\bot_\alpha$，c_α を考える．解釈 I は停止性を持つ項 $\mathcal{T}_\tau(\Sigma, \mathcal{V})$ から項 $\mathcal{T}_\tau(\Sigma \cup \bigcup_{\alpha \in \mathcal{S}}\{\bot_\alpha, \mathsf{c}_\alpha\}, \mathcal{V})$ への写像である．$t \equiv a[t_1, \ldots, t_n]$ に関しては $\tau(t) = \alpha$ である．このとき，$I(t)$ を以下で定義．

$$I(t) = \begin{cases} a[I(t_1), \ldots, I(t_n)] & \text{if } t \notin \Delta \\ \mathsf{c}_\alpha[a[I(t_1), \ldots, I(t_n)], Red_\alpha(\{I(t') \mid t \xrightarrow[R]{} t'\})] & \text{if } t \in \Delta \end{cases}$$

$$Red_\alpha(T) = \begin{cases} \bot_\alpha & \text{if } T = \emptyset \\ \mathsf{c}_\alpha[t, Red(T \setminus \{t\})] & \text{if } T \neq \emptyset \text{ かつ } t \text{ は } T \text{ の最小の要素} \end{cases}$$

整列可能定理より，最小な $\mathcal{T}_\tau(\Sigma, \mathcal{V})$ の存在は保証される．停止性をもつ代入 θ，変数 $x \in \mathcal{V}$ に対し，θ^I を $\theta^I(x) = I(\theta(x))$ で定義する．

停止性を持つ項上で整礎な順序 $\rhd_{sub} \cup \xrightarrow[R]{}$ に関して帰納的に解釈 I は定義されている．また，R が有限分岐である為，集合 $\{I(t') \mid t \xrightarrow[R]{} t'\}$ は有限である．故に I は well-defined である．

直感的に言えば，$I(t)$ は t が簡約化可能な項を全て”集め”，集められた項は記号 c を用いて１つの項で表現される．ただし，考える簡約化は Δ 中に現れるシンボルを含む部分項の簡約化のみである．それ以外の簡約化は与えられた項の対 C 又は実効規則 $\mathcal{U}(C)$ を用いて行う．C_e による書換えを行うことで，集められた項から１つを選ぶことができる．

補題 3.5 $l\theta$ が停止性を持つような規則 $l \to r \in C \cup \mathcal{U}(C)^{ex}$ と，代入 θ を考える．このとき，$I(l\theta) \xrightarrow[C_e]{*} l\sigma$ である．ここで，σ について，$x \in Var(l) \setminus Var(r)$，$\theta(x) = a[u_1, \ldots, u_k]$ ならば $\sigma(x) = a[I(u_1), \ldots, I(u_k)]$，それ以外のとき，$\sigma(x) = \theta^I(x)$ とする．

証明 t の構造帰納法により証明できる． □

補題 3.6 $r\theta$ が停止性を持つような規則 $l \to r \in C \cup \mathcal{U}(C)^{ex}$，代入 θ を考える．このとき，$I(r\theta) \equiv r\theta^I$．

証明 t の構造帰納法で示すことができる． □

補題 3.7 $s \xrightarrow[R]{} t$ かつ s が停止性を持つならば $I(s) \xrightarrow[\mathcal{U}(C) \cup C_e]{+} I(t)$.

証明 補題 *2.2* より，$s \equiv C[l\theta]$ *and* $t \equiv C[r\theta]$ を満たすようなある規則 $l \to r \in R^{ex}$，葉文脈 $C[]$，代入 θ が存在．題意を $C[]$ の帰納法により示す．$C[]$ は葉文脈であるため，以下の３つの場合を考えればよい．

- $C[] \equiv \Box$ かつ $s \notin \Delta$ の場合．このとき，$l \to r \in \mathcal{U}(C)^{ex}$．ここで代入 σ を補題 *3.5* と同様に定義する．補題 *3.5*，*3.6* より，$I(l\theta) \xrightarrow[C_e]{*} l\sigma \xrightarrow[\mathcal{U}(C)]{} r\sigma \equiv r\theta^I \equiv I(r\theta)$.
- $C[] \equiv a[\ldots, C'[], \ldots]$ かつ $s \notin \Delta$ の場合．このとき $t \notin \Delta$，故に $I(C[l\theta]) \equiv a[\ldots, I(C'[l\theta]), \ldots] \xrightarrow[\mathcal{U}(C) \cup C_e]{+} a[\ldots, I(C'[r\theta]), \ldots] \equiv I(a[\ldots, C'[r\theta], \ldots])$.
- $s \in \Delta$ の場合．このとき，$I(C[l\theta]) \xrightarrow[C_e]{} Red(\{I(v) \mid C[l\theta] \xrightarrow[R]{} v\}) \xrightarrow[C_e]{+} I(C[r\theta])$. □

定理 3.8 : $T^{args}_{SN}(R)$ 上の任意の $\langle C, \xrightarrow[R]{} \rangle$-鎖は $T^{args}_{SN}(R)$ 上の $\langle C, \xrightarrow[\mathcal{U}(C) \cup C_e]{} \rangle$-鎖でもある．

証明 補題 *3.5*，*3.6*，*3.7* より明らか． □

4. 直積型項へのラベル付け

STRS において，直積型を表現するには記号 tp を用いる．tp を用いることで直積型項の自然な表現が可能になるが，停止性証明時に切り落とし関数を用いる場合に問題が起こる．以下，STRS R_3 を用い議論を進める．

$R_3 = R_{mul} \cup R_2$

$$\{div[(div[(x, y)], z)] \to div[(x, mul[(y, z)])]\}$$

R_3 の停止性を定理 3.2 により証明する場合，問題となる再帰成分は $\{\langle div^\#[(s[x], s[y])], div^\#[(sub[(x, y)], s[y])]\rangle \langle div^\#[(div[(x, y)], z)], div^\#$ である．この再帰成分が制約を満たすかどうかの判定の際に注目すべき点は，$add[(0, y)] \gtrsim y \cdots$ 　と $div[(s[x], \underline{s[y]})] \gtrsim s[div[(sub[(x, \underline{y})], \underline{s[y]})]] \cdots$ 　を同時に順序付けなければならない点である．　はそのままでは順序が付けられない為，切り落とし関数を適用する．具体的には，　の下線部を切り落とす，つまり『tp の第２引数を切り落とす』ように切り落とし関数を定義する．すると　は順序付けが成功する．

ところがこのとき　の順序がつけられない．それは切り落としが sub, div の下の tp のみならず add の下の tp でも行われ，切り落としの結果，$add[(0)] \to y$ となった為である．

これは切り落とし関数は関数記号毎に定義され，直積型項は全て tp という１つの関数記号で表してしまうためである．この問題に対し，我々は文脈依存のラベル付けによる tp の区別を提案する．次節でその手法を述べる．

4.1 直積型項へのラベル付け

本節では直積型項へのラベル付け手法（定義 4.1），そしてラベル付けされた STRS と補助 STRS A_R（定義 4.2）によりラベル付け前の STRS の書換え関係が模倣できることを述べる．そして直積型へのラベル付けの概念を強計算依存対法に組み込んだ定理（定理 4.3）を挙げる．尚，ラベル付けは文献 [5] でも提案されているが，そのラベル付けはボトムアップに行われるのに対し，本手法はトップダウンに行う．

まず最初に，例としてラベル付けされた R_{add} を例に示す．

$$\begin{cases} add[(0,y)_{(add,1)}] \to y \\ add[(s[x],y)_{(add,1)}] \to s[add[(x,y)_{(add,1)}]] \end{cases}$$

ラベル付けされた R_3 を $lab(R_3)$ とする．R_3 を $lab(R_3)$ で模倣が可能ならば，$lab(R_3)$ の停止性が R_3 の停止性を保証する．$lab(R_3)$ での模倣の例を示す．

$$div[(s[0],s[0])_{(div,1]} \xrightarrow[lab(R_3)]{} s[div[(sub[(0,0)_{(sub,1)}],s[0])_{(div,1)}]] \xrightarrow[lab(R_3)]{} s[div[(0,s[0])_{(div,1)}]]] \xrightarrow[lab(R_3)]{} s[0]$$

この例では R_3 は補助 STRS を用いずに $lab(R_3)$ で模倣できる．上記の様なラベル付けをするラベル付け関数を以下で定義する．

定義 4.1 関数記号 a，ポジション p に対し，$\perp$ または (a,p) を**ポジション対**と呼ぶ．$\perp$ は情報を持たない対を表す．ラベル付け関数 lab を以下で定義する．

$lab(a[t_1,\ldots,t_n],(a',p)) =$

$$\begin{cases} a[lab(t_1,(a,1)),\ldots,lab(t_n,(a,n))] & \text{if } a \in \Sigma\backslash\{tp\} \\ a[lab(t_1,\perp),\ldots,lab(t_n,\perp)] & \text{if } a \in \mathcal{V} \\ a_{(a',p)}[lab(t_1,(a',p1)),\ldots,lab(t_n,(a',pn))] & \text{if } a = tp \end{cases}$$

$lab(a[t_1,\ldots,t_n],\perp) =$

$$\begin{cases} a[lab(t_1,(a,1)),\ldots,lab(t_n,(a,n))] & \text{if } a \in \Sigma\backslash\{tp\} \\ a[lab(t_1,\perp),\ldots,lab(t_n,\perp)] & \text{if } a \in \mathcal{V}\cup\{tp\} \end{cases}$$

$lab(t,(a,p))$ を，項 $tp_{(a,p)}[t_1,\ldots,t_n]$ を $(t_1,\ldots,t_n)_{(a,p)}$ と表現することもある．STRS R に対し，$lab(R) = \{lab(l,\perp) \to lab(r,\perp) \mid l \to r \in R\}$ とし，文脈 $C[]$ に対し，$C_{lab} = lab(C[],\perp)$ とする．代入 θ に対し，θ_{lab} を以下で定義する．

$\theta(x) = a'[t'_1,\ldots,t'_k]$ であるとき，

$\theta_{lab}(x) =$

$$\begin{cases} a'[lab(t'_1,(a,1)),\ldots,lab(t'_k,(a,k))] & \text{if } a' \not\equiv tp \\ a'[lab(t'_1,\perp),\ldots,lab(t'_k,\perp)] & \text{if } a' \equiv tp \end{cases}$$

R_3 は $lab(R_3)$ で模倣できた．だが，この模倣は全ての STRS で可能な訳ではない．実際，以下に示す STRS R_4 は $lab(R_4)$ で模倣できない．

$$R_4 = \begin{cases} F[(x,y)] & \to & (x,y) \\ G[(x,y)] & \to & x \\ H[(f,(x,y))] & \to & f[(x,y)] \end{cases}$$

$$lab(R_4) = \begin{cases} F[(x,y)_{(F,1)}] & \to & (x,y) \\ G[(x,y)_{(G,1)}] & \to & x \\ H[(f,(x,y)_{(H,12)})_{(H,1)}] & \to & f[(x,y)] \end{cases}$$

このとき，$lab(R_4)$ において $lab(G[F[(0,0)]],\perp) \equiv G[F[(0,0)_{(F,1)}]] \xrightarrow[lab(R_4)]{} G[(0,0)] \not\equiv lab(G[(0,0)],\perp)$ という書換え系列が存在する．これは tp が R_4 の第一規則の右辺の根において出現しており，適当なラベル付けが文脈により変化するため確定しない．故に tp にラベル付けが不可能なのである．

また，$lab(H[(F,(0,0))],\perp) \equiv H[(F,(0,0)_{(H,12)})_{(H,1)}] \xrightarrow[lab(R_4)]{} F[(0,0)] \not\equiv lab(F[(0,0)],\perp)$ のような書換え系列も存在する．これは第 3 規則の右辺において，高階変数 f の下に tp が現れている事による．代入されて初めて文脈が確定するため tp へのラベル付けが不可能なのである．

このようなラベル付けが不可能な tp が出現している問題に対し，補助 STRS A_R を定義する．

定義 4.2 単純型 α に対し，$Aux(\alpha)$ を以下で定義．

$$Aux(\alpha) = \begin{cases} tp[Aux(\beta_1),\ldots,Aux(\beta_n)] \\ \qquad \text{if } \alpha \text{ is product type s.t. } \alpha = \beta_1 \times \cdots \times \beta_n \\ Z \quad \text{otherwise} \end{cases}$$

ここで，Z は型 α のフレッシュな変数である．補助 STRS A_R を以下で定義する．

$$A_R = \{Aux(\alpha) \to lab(Aux(\alpha),(a,i)) \mid \alpha \text{ is product-typed variable s.t. } \forall a \in \Sigma, \tau(a) = \alpha_1 \to \cdots \to \alpha_n \to \beta, \alpha_i = \alpha\}.$$

尚，通常の STRS では tp が規則の左辺の根に出現することは禁止しているが補助 STRS に関しては許すものとする．

定義に従い R_4 の補助 STRS A_{R_4} を以下で示す．

$$A_{R_4} = \begin{cases} (Z_1,Z_2) & \to & (Z_1,Z_2)_{(F,1)} \\ (Z_1,Z_2) & \to & (Z_1,Z_2)_{(G,1)} \\ (Z_1,(Z_2,Z_3)) & \to & (Z_1,(Z_2,Z_3)_{(H,12)})_{(H,1)} \end{cases}$$

これにより先ほど不可能であった R_4 の書換え関係の模倣が $lab(R_4) \cup A_{R_4}$ により可能となる．

$$G[F[(0,0)_{(F,1)}]] \xrightarrow[lab(R_4)]{} G[(0,0)] \xrightarrow[A_{R_4}]{} G[(0,0)_{(G,1)}] \xrightarrow[lab(R_4)]{} 0$$

$$H[(F,(0,0)_{(H,12)})_{(H,1)}] \xrightarrow[lab(R_4)]{} F[(0,0)] \xrightarrow[A_{R_4}]{} F[(0,0)_{(F,1)}] \xrightarrow[lab(R_4)]{} (0,0)$$

もう一例挙げる．規則の左辺に堅固ではないものが含まれる例である．

$$R_5 = \begin{cases} F\ [f(x,y)] & \to & f[(x,y)] \\ G[(x,y)] & \to & x \end{cases}$$

$$lab(R_5) = \begin{cases} F[f(x,y)] & \to & f[(x,y)] \\ G[(x,y)_{(G,1)}] & \to & x \end{cases}$$

R_5 においては $F[G[(0,0)]] \xrightarrow[R]{} G[(0,0)] \xrightarrow[R_5]{} 0$ という書換え系列が存在しているのに対し，$lab(R_5)$ においては項 $lab(F[G[(0,0)]]) \equiv F[G[(0,0)_{(F,1)}]]$ が規則にマッチングしないため，書換えができず，模倣が出来ない．故に規則の左辺は全て堅固でであるという制限を与えることで模倣が可能となる．以上の議論をを強計算依存対法に導入，補題 4.7，4.8，定理 3.2 より以下の定理が成立する．

定理 4.3 以下を満たすとき PFP-STRS R は停止性を持つ．

(1) R^{ex} に現れる変数は全て直積型を持たない

(2) 全ての規則 $l \to r \in R$ について l は堅固

(3) 全ての再帰成分 $C \in RC_{SC}(R)$ について，以下のいづれかを満たす

– C が部分項基準を満たす．

– ある簡約化対 $(\gtrsim, >)$ が存在し，$lab(\mathcal{U}(C)) \cup C_e \cup A_R \cup lab(C) \subseteq \gtrsim$，$lab(C) \cap > \neq \emptyset$．

この定理を用いると，R_3 の停止性を証明することが出来る．具体的には，切り落とし関数を $\pi(tp_{(div,1)}) = [1]$ と定義，簡約化順序として再帰経路順序を用いると停止性が証明できる．

4.2 証　明

補題 4.4 A_R を補助 STRS とする．任意の項 t，ポジション対 (a,p) に対し，$lab(t,\perp) \xrightarrow[A_R]{*} lab(t,(a,p))$ が成立．

証明　$t \equiv b[t_1, \ldots, t_n]$ とする．$lab(t, \perp) \xrightarrow[A_R]{*} lab(t, (a, p))$ を $|t|$ による帰納法で証明する．

$b \in \Sigma \backslash \{tp\}$ や $b \in \mathcal{V}$ の場合は明らか．$b \equiv tp$ の場合は

$$
\begin{aligned}
& lab(b[t_1, \ldots, t_n], \perp) \\
& \equiv b[lab(t_1, \perp), \ldots, lab(t_n, \perp)] \\
& \xrightarrow[A_R]{*} b[lab(t_1, (a, p1)), \ldots, lab(t_n, (a, pn))] \\
& \xrightarrow[A_R]{} b_{(a,p)}[lab(t_1, (a, p1)), \ldots, lab(t_n, (a, pn))] \\
& \equiv lab(b[t_1, \ldots, t_n], (a, p))
\end{aligned}
$$

□

補題 4.5　補助 STRS A_R，項 t，代入 θ を考える．t に出現するすべての変数が直積型を持たないとする．このとき，$lab(t, (a, p))\theta_{lab} \xrightarrow[A_R]{*} lab(t\theta, (a, p))$ が成立．さらに，t が堅固の場合，$lab(t, (a, p))\theta_{lab} \equiv lab(t\theta, (a, p))$．

証明　$t \equiv b[t_1, \ldots, t_n]$ とする．$|t|$ による帰納法で $lab(t, (a, p))\theta_{lab} \xrightarrow[A_R]{*} lab(t\theta, (a, p))$ を証明する．尚，$lab(t, (a, p))\theta_{lab} \equiv lab(t\theta, (a, p))$ の場合も同様に証明できる．

$b \in \Sigma$，$b \in \mathcal{V}, root(\theta(a)) \in \mathcal{V}$ の場合は明らか．条件より直積型を持つ変数は出現しないので，$b \in \mathcal{V}, root(\theta(a)) = tp$ の場合はない．$b \in \mathcal{V}, root(\theta(a)) \in \Sigma \backslash \{tp\}$ の場合は

$\theta_{lab}(b) \equiv b'[lab(t', (b', 1)), \ldots, lab(t'_m, (a', m))]$ とすると，補題 *4.4* より，

$$
\begin{aligned}
& lab(b[t_1, \ldots, t_n], (a, p))\theta_{lab} \\
& \equiv b[lab(t_1, \perp), \ldots, lab(t_n, \perp)]\theta_{lab} \\
& \equiv b'[lab(t'_1, (b', 1)), \ldots, lab(t'_m, (b', m)), \\
& \qquad lab(t_1, \perp)\theta_{lab}, \ldots, lab(t_n, \perp)\theta_{lab}] \\
& \xrightarrow[A_R]{*} b'[lab(t'_1, (b', 1)), \ldots, lab(t'_m, (b', m)), \\
& \qquad lab(t_1, (b', m+1))\theta_{lab}, \ldots, lab(t_n, (b', m+n))\theta_{lab}] \\
& \xrightarrow[A_R]{*} b'[lab(t'_1, (b', 1)), \ldots, lab(t'_m, (b', m)), \\
& \qquad lab(t_1\theta, (b', m+1)), \ldots, lab(t_n\theta, (b', m+n))] \\
& \equiv lab(b'[t'_1, \ldots, t'_m, t_1\theta, \ldots, t_n\theta], (a, p)) \\
& \equiv lab(b[t_1, \ldots, t_n]\theta, (a, p))
\end{aligned}
$$

□

補題 4.6　補助 STRS A_R，項 t，葉文脈 $C[]$ を考える．このとき，$C_{lab}[lab(t, \perp)] \xrightarrow[A_R]{*} lab(C[t], \perp)$. が成立．特に，$root(t) \not\equiv tp$ ならば，$C_{lab}[lab(t, \perp)] \equiv lab(C[t], \perp)$ が成立．

証明　$C_{lab}[]$ において □ がポジション対 (a, p) によりラベル付けされる，すなわち，$lab(C[t], \perp) \equiv C_{lab}[lab(t, (a, p))]$ とする．補題 *4.4* より，$C_{lab}[lab(t, \perp)] \xrightarrow[A_R]{*} C_{lab}[lab(t, (a, p))] \equiv lab(C[t], \perp)$ である．また，$root(t) \not\equiv tp$ である時 $lab(t, \perp) \equiv lab(t, (a, p))$ より $C_{lab}[lab(t, \perp)] \equiv lab(C[t], \perp)$ が成立．　□

補題 4.7　STRS R を考える．以下を満たすとき，$s \xrightarrow[R]{} t$ ならば $lab(s, \perp) \xrightarrow[lab(R) \cup A_R]{+} lab(t, \perp)$.

(1)　R^{ex} に現れる変数は全て直積型を持たない

(2)　全ての規則 $l \rightarrow r \in R$ について l は堅固

証明　$C[]$ を葉文脈, θ を代入とすると，$s \equiv C[l\theta] \xrightarrow[R^{ex}]{} C[r\theta] \equiv t$．仮定， 補題 *4.5*，*4.6*，$lab(s, \perp) \equiv lab(C[l\theta], \perp)$ より，

$C_{lab}[lab(l, \perp)\theta_{lab}] \xrightarrow[lab(R^{ex})]{} C_{lab}[lab(r, \perp)\theta_{lab}] \xrightarrow[A_{R^{ex}}]{*} lab(C[r\theta])$ $\equiv lab(t)$．従って $lab(s, \perp) \xrightarrow[lab(R^{ex}) \cup A_{R^{ex}}]{+} lab(t, \perp)$ を得る．□

補題 4.8　任意の STRS R に関して，$lab(RC_{SC}(R)) \equiv RC_{SC}(lab(R) \cup A_R)$.

謝辞　本研究は一部，科研費#16650005，#17700009，#18500011 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] K.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, 2001.

[2] 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.

[3] T.Arts, J.Giesl, Termination of Term Rewriting Using Dependency Pairs, Theoretical Computer Science, Vol.236, pp.133–178, 2000.

[4] K.Kusakari, Higher-Order Path Orders based on Computability, IEICE Transactions on Information and Systems, Vol.E87-D, No.2, pp.352–359, 2004.

[5] H. Zantema, Termination of Term Rewriting by Semantic Labelling, Fundamenta Informaticae, 1995, volume 24, pages 89-105

[6] R. Thiemann, J. Giesl, and P. Schneider-Kamp Improved Modular Termination Proofs Using Dependency Pairs IJCAR '04, Lecture Notes in Artificial Intelligence 3097, pages 75-90, 2004.