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

要旨. 単純型項書換え系は高階の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草刈と酒井は強計算依存対法を提案した．これは，静的な再帰の部分で無限ループが発生しない事を示す事により停止性を証明する手法である．本論文では，強計算依存対法により停止性証明を行う際に解く必要のある制約を取り扱い易いように削減・変換する 2 つの方法を提案する．1 つは実効規則の概念の導入であり，制約を劇的に削減することができる．もう 1 つは直積型項へのラベル付け法の導入であり，制約の解法の選択肢を増加させることができる．

Abstract. Simply-typed term rewriting systems model the computation of functional programs, and termination is one of the important properties of them. Kusakari and Sakai proposed the strongly computable dependency pair method for proving termination of simply-typed term rewriting systems. This method proves termination by showing that infinite loop is not generated in any static recursion component. In this paper, we enhance this method by two idea. One is reducing constraints by usable rules, the other is labeling product-typed terms.

1 はじめに

一階での項書換え系の停止性を証明するために，Arts と Giesl は依存対法を提案した [2]．依存対法は，無限書換え系列を作る恐れのある成分を項の対 (依存対) として書換え規則から抽出し，依存対間の関係を解析することにより再帰の部分を洗い出す．停止性を証明する際には，洗い出した再帰の部分で無限ループが発生しない事を保証する制約を解く必要がでてくる．

依存対法を高階の書換え系の停止性証明に用いるために，草刈が単純型項書換え系 (simply-typed term rewiting system; STRS) 上に [7]，酒井，渡辺，坂部が高階書換え系 (higher-order rewrite system) 上に [13] 拡張した．だがしかしこれらの手法は，動的な再帰構造，すなわち実行時の関数呼び出しの依存関係に基づく再帰構造を解析する．そのため，さまざまな関数が代入されうる高階変数を根に持つ項が解くべき制約中に含まれてしまい，実用的には不十分である．酒井と草刈は dependency forest の概念を導入し，静的な再帰構造解析，すなわち関数定義の依存関係に基づく再帰構造解析による依存対法を提案し，高階変数を根に持つ項を制約から取り除くことに成功した [12]．しかし，対象とする高階書換え系に非常に強い制限を課すことになった．また，青戸と山田は defunctionalization[11] の概念を用いて，制約中の項の根に出現する高階変数を，代入可能な全ての関数で展開することにより取り除く手法を提案した [1]．だが，これは本質的には動的な再帰構造解析に基づく手法である．

近年，草刈と酒井は，型付 λ 計算の停止性証明で導入された強計算性の概念を用いて，静的な再帰構造解析法である強計算依存対法を提案した [9]．強計算依存対法は直接関数渡し (Plain Function-Passing; PFP) と呼ばれる性質を満たす STRS に適用できる．直接関数渡しとは，直観的には右辺中の高階変数が左辺の被定義関数の引数になっているという性質であり，典型的な高階関数のほとんどが満たす性質である．それ故に，静的な再帰構造解析による証明の容易さばかりでなく，非常に実用性が高い．強計算依存対法が特に有効な例として次の型付き組合せ子論理がある [4]．

$$\begin{cases} S[f,g,x] & \to & f[x,g[x]] \\ K[x,y] & \to & x \end{cases}$$

組合せ子論理はチューリング機械と等価な表現力を持つため停止性を持たない．一方，型付き組合せ子論

理は停止性を持つことが知られている．強計算依存対法を用いると，静的な再帰を持たない事と全ての高階変数が左辺の引数部分に出現している（PFP），という 2 つの容易な判定のみで型付き組み合わせ子論理の停止性を証明することができる．

強計算依存対法で STRS の停止性を証明する際には，静的な再帰部分で無限ループが発生しないことを保証する制約を解く必要がある．本論文では，この制約を取り扱い易いように削減・変換する 2 つの手段を提案する．

強計算依存対法はそれぞれの再帰成分を解析する際に，STRS の全ての規則も同時に解析する．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．本論文では，この問題に対し再帰の際に呼び出す可能性のある規則（実効規則）に注目すれば十分である事を示す．実効規則の概念は，項書換え系に対して文献 [5], [14] で提案されたものであり，これにより STRS の停止性証明においても解かれるべき制約が削減され，証明効率が格段に向上し，証明能力も強力になる．

また，関数プログラムにおいて様々な型をもつ複数のデータを一つに束ねたデータを，STRS では直積型項を用いることにより自然に表現することができる．一方で強計算依存対法に基づいて関数プログラムの停止性を示す場合に，直積型項が制約を解く妨げとなり証明能力の低下を引き起こす．そこで本論文では直積型項へのラベル付け法を提案する．具体的には，直積型項を表現する特別な関数記号 tp を文脈に依存したラベル付けを行うことにより区別する．これにより，停止性証明時に満たすべき制約の解法の選択肢を増加させることが可能となる．本手法により，証明能力がさらに強力となる．

本論文は次のように構成される．2 節で本論文に必要な諸定義を与え，3 節で強計算依存対法を紹介する．4 節で STRS 上での実効規則の概念を提案し，5 節で直積型項へのラベル付け法を提案する．

2 準備

2.1 単純型項書換え系

単純型項書換え系（simply-typed term rewiting system; STRS）は文献 [7] で提案された．本節では文献 [9] 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 中の全ての変数の集合を $Var(t)$ で表す．項 $t\equiv a[t_1,\ldots,t_n]$ における位置の集合 $Pos(t)$ を正整数の列（空列を ε とする）を用いて $Pos(t)=\{\varepsilon\}\cup\{ip'\mid 1\leq i\leq n, p'\in Pos(t_i)\}$ で定義する．$Pos(t)$ 上の順序 $\succ$ を以下で定義する．$p\succ q$ であるとは，ある $w(\neq\varepsilon)$ で $p=qw$ となることである．ε を根の位置と呼び，$p\prec q$ となる q が存在しない p を葉の位置と呼ぶ．項 $t\equiv a[t_1,\ldots,t_n]$ の根の位置の記号 a を $root(t)$ で記し，$args(t)=\{t_1,\ldots,t_n\}$ とする．

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 $\theta(a[t_1,\ldots,t_n])$ を，$a\in\Sigma$ 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 $a'[t'_1,\ldots,t'_k,\theta(t_1),\ldots,\theta(t_n)]$ で定義する．$\theta(t)$ 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 $\Box$ が，一箇所だけ出現する項である．これらのうち特に，$\Box$ が根の位置に出現する文脈を根文脈，$\Box$ が葉の位置に出現する文脈を葉文脈と呼ぶ．文脈 $C[\,]$ 中の $\Box$ 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．このとき，$\Box$ の $C[\,]$ における出現位置 p を用いて，t' を $t|_p$ で表す．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．例えば，項 $a[a'[x,y]]$ の部分項は $a[a'[x,y]], a'[x,y], x, y$ であり，拡張部分項はこれらの他に $a[\,], a'[\,], a'[x]$ がある．項 t' が項 t の部分項であることを $t\trianglerighteq_{sub}t'$ で記し，拡張部分項であることを $t\trianglerighteq_{esub}t'$ で記す．また，$\triangleright_{sub}$ を $\trianglerighteq_{sub}\setminus\equiv$ で定義し，$\triangleright_{esub}$ を $\trianglerighteq_{esub}\setminus\equiv$ で定義する．項 t の全ての部分項からなる集合を $Sub(t)$ で記す．

空でない基底型の集合を $\mathcal{B}$ 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 $\rightarrow$，$\times$ を用いて $\mathcal{S}::=\mathcal{B}\mid(\mathcal{S}\rightarrow\mathcal{S})\mid(\mathcal{S}\times\cdots\times\mathcal{S})$ として定義される．$\rightarrow$ は右結合性を持ち $\times$ よりも優先度が低いとし，括弧を適宜省略する．本論文では，単純型を単に型と呼ぶ事がある．$\alpha_1\times\cdots\times\alpha_n(n\geq 2)$ の形の単純型を直積型と呼ぶ．直積型の項を表現するため，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，$\tau(tp)$ 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，$a=tp$ のときは $\alpha=\alpha_1\times\cdots\times\alpha_n(n\geq 2)$，そうで無いときは $\tau(a)=\alpha_1\to\cdots\to\alpha_n\to\alpha$ となることである．単純型 α が単純型 β の接尾辞であるとは，ある $\alpha_1,\ldots,\alpha_n$ に対し，$\beta=\alpha_1\to\cdots\to\alpha_n\to\alpha(n\geq 0)$ となる事であり，$\alpha\sqsubseteq\beta$ 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 $C[\,]$ は共に単純型を持つとする．本論文では単純型項しか取り扱わないため，以降では単純型項をを単に項と呼ぶこともある．

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，どの項 t においても $\{t'\mid t\underset{R}{\to}t'\}$ が有限であることである．

$l\to r$ を型 $\tau(l)=\alpha_1\to\cdots\to\alpha_n\to\alpha$ を持つ規則とする．ただ α は基底型または直積型である．このとき，$(l\to r)^{ex}$ を $\{l\to r,\ l[z_1]\to r[z_1],\ \ldots,\ l[z_1,\ldots,z_n]\to r[z_1,\ldots,z_n]\}$ で定義する．ここで，$z_1,\ldots,z_n$ は $\forall i.\tau(z_i)=\alpha_i$ である他に現れない新しい変数である．また，$R^{ex}=\bigcup_{l\to r\in R}(l\to r)^{ex}$ とする．

命題 2.1 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 $t\equiv C[r\theta]$．

項 t が STRS R において停止性を持つ ($SN(R,t)$ と記す) とは，t から始まる $\underset{R}{\to}$ による無限列がないことである．R が停止性を持つとは，任意の項 t に対して $SN(R,t)$ が成立することである．また，$\mathcal{T}^{args}_{SN}(R)=\{t\mid\forall u\in args(t).SN(R,u)\}$ とする．

最後に依存対法で用いる記法を紹介する．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 $\mathcal{D}_R$ で記す．また，それ以外の関数記号を構成子記号と呼び，R の構成子記号全体からなる集合を $\mathcal{C}_R$ で記す．各関数記号 $f\in\mathcal{D}_R$ に対し，印付記号 $f^\#$ を用意する．また，t の根記号を対応する印付記号で置き換えた項を $t^\#$ で表し，印付項と呼ぶ．ここで，$root(t)\notin\mathcal{D}_R$ の場合には $t^\#\equiv t$ とする．

2.2 簡約化対 (準簡約化対)

本節では依存対法で重要な役割をなす簡約化対 (準簡約化対) を紹介する．

項の対の上での述語 P の下で 2 項関係 $>$ が代入に閉じているとは，$s>t$ かつ $P(s,t)$ ならば $s\theta>t\theta$ であることである．また，$>$ が文脈 (葉文脈) に閉じているとは，$s>t$ ならば任意の文脈 (葉文脈) $C[\,]$ に対し，$C[s]>C[t]$ が成立することである．擬順序 $\gtrsim$ と狭義の半順序 $>$ の対 $(\gtrsim,>)$ が述語 P の下で簡約化対 (準簡約化対) であるとは，$\gtrsim,>$ が共に P の下で代入に閉じ，$\gtrsim$ が文脈 (葉文脈) に閉じ，$>$ が整礎であり，$\gtrsim\cdot>\subseteq>$ または $>\cdot\gtrsim\subseteq>$ が成立することである．

簡約化対の設計にはしばしば引数切り落とし法が利用される [7, 2]．以下では文献 [7] で提案された STRS 上へ拡張された引数切り落とし法を紹介する．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 $i_1<\cdots<i_m\leq n$ 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．$i_j\leq k$ を満たす $\pi(f)$ の最大部分リスト $[i_1,\ldots,i_j]$ を $\pi(f)^{\leq k}$ で表す．項 $a[t_1,\ldots,t_n]$ に対し $\pi(a[t_1,\ldots,t_n])$ を，$a\in\mathcal{V}$ ならば $a[\pi(t_1),\ldots,\pi(t_n)]$ で，$a\in\Sigma$ (ただし $\pi(a)^{\leq n}=[i_1,\ldots,i_m]$) ならば $a[\pi(t_{i_1}),\ldots,\pi(t_{i_m})]$ で定義する．また，$s\gtrsim^\pi t$ を $\pi(s)\gtrsim\pi(t)$ で，$s>^\pi t$ を $\pi(s)>\pi(t)$ で定義する．

次に STRS 上の再帰経路順序 [7] を紹介する．

定義 2.2 $\trianglerighteq$ を Σ 上の擬順序とし，$\sim$ を $\trianglerighteq$ から生成される同値関係とする．項 $s\equiv a[s_1,\ldots,s_n],t\equiv a'[t_1,\ldots,t_m]$ に対し $s>_{rpo}t$ であるとは以下のいずれかを満たすことである．

- $a\triangleright a'$ かつ各 j について $s>_{rpo}t_j$
- $a\sim a'$ かつ $\{s_1,\ldots,s_n\}>^{mul}_{rpo}\{t_1,\ldots,t_m\}$
- $s_i\geq_{rpo}t$ となる i が存在
- $a=a'\in\mathcal{V}$ かつ $\{s_1,\ldots,s_n\}>^{mul}_{rpo}\{t_1,\ldots,t_m\}$
- $a'\in\mathcal{V}$，a が $\trianglerighteq$ に関して最大，かつ $\{s_1,\ldots,s_n\}\geq^{mul}_{rpo}\{t_1,\ldots,t_m\}$

ここで，$\geq_{rpo}$ は $>_{rpo} \cup \equiv$ であり，$>_{rpo}^{mul}$ は $>_{rpo}$ の多重集合上への拡張である．

項 t が堅固 (firmness) であるとは，t 中の変数が全て葉の位置に出現していることである．項の対 (s,t) について s が堅固であるとき左堅固であるといい，$LF(s,t)$ で記す．

命題 **2.3** ([7]) 述語 LF の下で($\gtrsim_{rpo}^{\pi}, >_{rpo}^{\pi}$)は準簡約化対である．

2.3 部分項基準

簡約化対 (準簡約化対) と並び，依存対法で重要な役割をなす部分項基準の概念を紹介する．部分項基準は TRS 上で提案された [5]．以下では文献 [9] の定式化に従う．

定義 **2.4** R を STRS，$\mathcal{C}$ を印付項の対の集合とする．$\mathcal{C}$ が部分項基準を満たすとは，すべての $\langle u^{\#}, v^{\#}\rangle \in \mathcal{C}$ に対して $root(u), root(v) \notin \mathcal{V}$ であり，さらに $\mathcal{D}_R$ から空でない正整数列への関数 δ が存在し，以下を満たすことである．

1. ある $\langle u^{\#}, v^{\#}\rangle \in \mathcal{C}$ で $u|_{\delta(root(u))} \rhd_{esub} v|_{\delta(root(v))}$ が成立
2. 各 $\langle u^{\#}, v^{\#}\rangle \in \mathcal{C}$ で以下がすべて成立
   - $u|_{\delta(root(u))} \trianglerighteq_{esub} v|_{\delta(root(v))}$
   - 各 $p \prec \delta(root(u))$ について $root(u|_p) \notin \mathcal{V}$
   - $\varepsilon \prec q \prec \delta(root(v))$ を満たす各 q について $root(v|_q) \in \mathcal{C}_R$

各 $f \in \mathcal{D}_R$ に対し $\delta(f)$ は，文献 [5] では正整数しか許されなかったが，文献 [9] で正整数列がとれるように拡張されている．

3 強計算依存対法

この節では文献 [9] で提案された 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．この証明法は，静的な再帰成分を抽出し，それぞれの再帰の部分で無限ループが発生しない事を保証する制約を解くことにより停止性を示す手法である．まず，静的な再帰構造解析の土台となる SC 依存対の概念を紹介する．

定義 **3.1** STRS R に対し，used product type $upt(R)$ を以下で定義する．$\alpha \in upt(R)$ とは，α が直積型であり，ある規則 $l \to r \in R$ と変数 $z \in Var(r)$ が存在して $\tau(z) = \alpha$ となることである．STRS R，項 l に対し，拡張引数の集合 $e_args(R,l)$ を以下で定義する．$u \in e_args(R,l)$ 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，集合 $safe(R,l)$ を $e_args(R,l)$ と $\{u \in Sub(l) \mid u \not\equiv l, \tau(u) \in \mathcal{B} \cup upt(R)\}$ の和集合で定義する．

対 $\langle l^{\#}, a^{\#}[r_1,\ldots,r_m,z_1,\ldots,z_n]\rangle$ が STRS R の SC 依存対であるとは，この 2 つの要素が基底型か直積型であり，ある規則 $l \to C[a[r_1,\ldots,r_m]] \in R^{ex}$ が存在し，$\forall k \leq m.a[r_1,\ldots,r_k] \notin safe(R,l)$ が成立することである．ここで $a \in \mathcal{D}_R$，$C[\,]$ は葉文脈，$z_1,\ldots,z_n$ は他に現れない新しい変数であるとする．また，R の SC 依存対の集合を $DP_{SC}(R)$ で記す．

例 **3.2** 以下の STRS を考える．

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

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

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

$R_{sub} \cup R_{div}$ と R_{foldl} の SC 依存対 $DP_{SC}(R_{sub} \cup R_{div})$ と $DP_{SC}(R_{foldl})$ は以下のようになる．

$$DP_{SC}(R_{sub} \cup R_{div}) = \left\{ \begin{array}{ll} \langle sub^{\#}[(s[x],s[y])], sub^{\#}[(x,y)]\rangle & \cdots(1) \\ \langle div^{\#}[(s[x],s[y])], div^{\#}[(sub[(x,y)],s[y])]\rangle & \cdots(2) \\ \langle div^{\#}[(s[x],s[y])], sub^{\#}[(x,y)]\rangle & \cdots(3) \end{array} \right.$$

$$DP_{SC}(R_{foldl}) = \left\{ \langle foldl^{\#}[f,z,cons[(x,xs)]], foldl^{\#}[f,f[(x,z)],xs]\rangle \right.$$

このように SC 依存対は静的な関数の依存関係，すなわち関数定義による依存関係を表している．

特筆すべき点は高階変数による関数の動的な依存関係，すなわち高階変数に起因する関数呼び出しに基づく依存関係は考慮しない事である．具体的には $\langle foldl^{\#}[f,z,cons[(x,xs)]], f[(x,z)]\rangle$ は $DP_{SC}(R_{foldl})$ に含まれない．だが一般にこのような高階変数による動的な依存関係を無視することは

できない．例えば，

$$foo[bar[f]] \to f[bar[f]]$$

における関数 foo は静的な依存関係を持たない，すなわち foo の定義は他の関数に依存せず再帰を用いてもいない．しかし，この STRS は停止性を持たない（$foo[bar[foo]] \to foo[bar[foo]]$）．よって高階変数による動的な依存関係を無視することは一般にはできない．このため文献 [9] では，高階変数による動的な関数呼び出しを無視できる条件として直接関数渡しの概念が導入されている．

定義 3.3 STRS R が直接関数渡し (Plain Function-Passing) であるとは，各葉文脈 $C[\,]$ と $a \in \mathcal{V}$ と規則 $l \to C[a[r_1, \ldots, r_m]] \in R$ について，$a[r_1, \ldots, r_k] \in safe(R, l)$ となる $k \leq m$ が存在することである．直接関数渡しである STRS を PFP-STRS と略記する．

次に，強計算依存対法に理論的根拠を与える強計算性の概念を紹介する．

定義 3.4 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，$SC(R,u) \wedge \tau(u) = \alpha$ を満たす任意の u について $SC(R, t[u])$．

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

命題 3.5 $SC(R,t)$ ならば $SN(R,t)$ が成立．

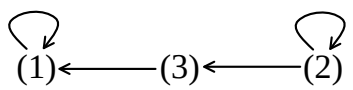
定義 3.6 $\mathcal{C}$ を印付項の対の集合，$\gg$ を 2 項関係，T を項の集合とする．$\mathcal{C}$ の対の列 $\langle u_0^\#, v_0^\# \rangle \langle u_1^\#, v_1^\# \rangle \ldots$ が T 上の $\langle \mathcal{C}, \gg \rangle$-鎖であるとは，各 i で $u_i\theta_i, v_i\theta_i \in T$ かつ $(v_i\theta_i)^\# \gg^* (u_{i+1}\theta_{i+1})^\#$ となる $\theta_0, \theta_1, \ldots$ が存在することである．

定理 3.7 ([9]) PFP-STRS R において，$\mathcal{T}^{args}_{SC}(R)$ 上の無限 $\langle DP_{SC}(R), \to_R \rangle$-鎖が存在しないならば R は停止性を持つ．

この定理が強計算依存対法の基本定理である．次に，効率良く無限鎖が存在しないことを示すために用いられる依存グラフと再帰成分の概念を紹介する．

定義 3.8 $\mathcal{C}$ を印付項の対の集合，$\gg$ を 2 項関係，T を項の集合とする．T 上の $\langle \mathcal{C}, \gg \rangle$-依存グラフとは，有向グラフであり，そのノードは $\mathcal{C}$ の要素，また $\langle u_0^\#, v_0^\# \rangle$ から $\langle u_1^\#, v_1^\# \rangle$ への辺が存在するとは $\langle u_0^\#, v_0^\# \rangle \langle u_1^\#, v_1^\# \rangle$ が T 上の $\langle \mathcal{C}, \gg \rangle$-鎖であることである．$T$ 上の $\langle \mathcal{C}, \gg \rangle$-再帰成分とは T 上の $\langle \mathcal{C}, \gg \rangle$-依存グラフの強連結部分グラフのノードの集合である．特に，STRS R に対して $\mathcal{T}^{args}_{SC}(R)$ 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．

例 3.9 例 3.2 で与えた R_{sub} と R_{div} を考える．以下に示す $\langle DP_{SC}(R_{sub} \cup R_{div}), \to_{R_{sub} \cup R_{div}} \rangle$-依存グラフ



において，再帰成分 $RC_{SC}(R_{sub} \cup R_{div})$ は $\{(1)\}$ と $\{(2)\}$ の 2 つである．

このようにして $DP_{SC}(R_{sub} \cup R_{div})$ から再帰成分 $RC_{SC}(R_{sub} \cup R_{div})$ が抽出される．強計算依存対法はこれらの再帰成分上の無限鎖が存在しないことを示すことにより停止性を証明する手法である．

このことは次の定理のように示すことができる．

定理 3.10 ([9]) STRS R と印付項の対の有限集合 $\mathcal{C}$ を考える．但し，$\mathcal{C}$ のどの要素 $\langle u^\#, v^\# \rangle$ についても $root(u) \notin \mathcal{V}$ であるとする．このとき，$T^{args}_{SN}(R)$ 上の各 $\langle \mathcal{C}, \to_R \rangle$-再帰成分 $\mathcal{C}'$ が以下を満たすとき，$T^{args}_{SN}(R)$ 上の無限 $\langle \mathcal{C}, \to_R \rangle$-鎖は存在しない．

(i) $\mathcal{C}'$ が部分項基準を満たす．

(ii) ある簡約化対 $(\gtrsim, >)$ が存在し，$R \cup \mathcal{C}' \subseteq \gtrsim$ かつ $\mathcal{C}' \cap > \neq \emptyset$．

(iii) ある準簡約化対 $(\gtrsim, >)$ が存在し，$R^{ex} \cup \mathcal{C}' \subseteq \gtrsim$ かつ $\mathcal{C}' \cap > \neq \emptyset$．

定理 3.7 と定理 3.10 を命題 3.5 を用いて組み合わせることで次の停止性証明法を得る．

定理 3.11 ([9]) $DP_{SC}(R)$ が有限集合であり，各再帰成分 $\mathcal{C} \in RC_{SC}(R)$ が，以下のいずれかを満たす PFP-STRS R は停止性を持つ．

(i) $\mathcal{C}$ が部分項基準を満たす．

(ii) ある簡約化対 $(\gtrsim, >)$ が存在し，$R \cup \mathcal{C} \subseteq \gtrsim$ かつ $\mathcal{C} \cap > \neq \emptyset$．

(iii) ある準簡約化対 $(\gtrsim, >)$ が存在し，$R^{ex} \cup \mathcal{C} \subseteq \gtrsim$ かつ $\mathcal{C} \cap > \neq \emptyset$ ．

例 3.2 における R_{sub} と R_{div} の和集合 $R_{sub} \cup R_{div}$ の再帰成分 $RC_{SC}(R_{sub} \cup R_{div})$ の $\{(1)\}$ は (i) を満たす．$\{(2)\}$ は $\pi(tp) = [1]$，$div \triangleright s \triangleright sub$ とし，準簡約化対として $(>^{\pi}_{rpo}, \gtrsim^{\pi}_{rpo})$ を用いれば (iii) を満たす．

$$\begin{array}{rcl} div^{\#}[(s[x], s[y])] & >^{\pi}_{rpo} & div^{\#}[(sub[(x, y)], s[y])] \\ R_{sub} & \subseteq & \gtrsim^{\pi}_{rpo} \\ R_{div} & \subseteq & \gtrsim^{\pi}_{rpo} \end{array}$$

これにより $R_{sub} \cup R_{div}$ の停止性が示される．

4 実効規則

実効規則の概念は一階の TRS の最内停止性証明のために導入された [2]．その後，一階の TRS の停止性証明にも適用できることが示された [5, 14]．本節では実効規則の概念を STRS 上に拡張する．一階の TRS の場合とは異なり，高階変数の詳細な解析が必要となる．

4.1 実効規則による強計算依存対法の改良

強計算依存対法 (定理 3.11) は強力な手法であり，例えば $R_{sub} \cup R_{div}$ や R_{foldl} の停止性を示すことができる．しかしながらその和集合 $R_{sub} \cup R_{div} \cup R_{foldl}$ の停止性は $(\gtrsim^{\pi}_{rpo}, >^{\pi}_{rpo})$ を用いて示すことはできない．この原因は以下の再帰成分にある．

$$\{\langle div^{\#}[(s[x], s[y])], div^{\#}[(sub[(x, y)], s[y])]\rangle\}$$

この再帰成分は部分項基準を満たさない．よって残る方法は以下の制約を解くことである．

$$\begin{array}{rcl} div^{\#}[(s[x], s[y])] & >^{\pi}_{rpo} & div^{\#}[(sub[(x, y)], s[y])] \\ R_{sub} & \subseteq & \gtrsim^{\pi}_{rpo} \\ R_{div} & \subseteq & \gtrsim^{\pi}_{rpo} \\ R_{foldl} & \subseteq & \gtrsim^{\pi}_{rpo} \end{array}$$

しかし，制約 $R_{foldl} \subseteq \gtrsim^{\pi}_{rpo}$ を満たす事はできない．

一方，多くのプログラマーは，div の再帰が無限に繰りかえされないことを検証するために，何故 div の定義とは関係の無い $foldl$ についての制約が必要なのだろうか，と考えるかもしれない．このような考えを実現するのが実効規則の概念である．

一階の実効規則の概念を STRS 上に拡張する際には，高階変数の取り扱いに注意が必要である．特に，引数を持つ高階変数については詳細な解析が必要である．そのような高階変数を根に持つ部分項の集合 $\{t' \in Sub(t) \mid root(t') \in \mathcal{V},\ args(t') \neq \emptyset\}$ を $Sub^{int}_{\mathcal{V}}(t)$ で記す．STRS 上の実効規則の定義を以下で与える．

定義 4.1 R を STRS とし，$\langle u, v\rangle$ を項の対とする．集合 $\mathcal{U}'(\langle u, v\rangle)$ を以下のいずれかを満たす $l \to r \in R$ からなる集合として定義する．

(1) $root(v') = root(l)$ かつ $\tau(v') \sqsubseteq \tau(l)$ となる $v' \in Sub(v)$ が存在

(2) $\tau(root(v')) \sqsubseteq \tau(root(l))$ かつ $\tau(v') \sqsubseteq \tau(l)$ となる $v' \in Sub^{int}_{\mathcal{V}}(v)$ が存在

(3) $\tau(root(u')) \sqsubseteq \tau(l)$ かつ $root(u') \in Var(v)$ となる $u' \in Sub^{int}_{\mathcal{V}}(u)$ が存在

集合 $\mathcal{U}(\langle u, v\rangle)$ を $\mathcal{U}'(\langle u, v\rangle^{ex})$ を含み，かつ，すべての $l \to r \in \mathcal{U}(\langle u, v\rangle)$ について $\mathcal{U}(\langle l, r\rangle^{ex})$ を含む最小の集合として定義する[1]．項の対の集合 $\mathcal{C}$ に対し，実効規則 $\mathcal{U}(\mathcal{C})$ を $\bigcup_{\langle u,v\rangle \in \mathcal{C}} \mathcal{U}(\langle u, v\rangle)$ で定義する．

実効規則の概念を用いると前述の div の再帰成分の実効規則は R_{sub} となり，制約 $R_{div} \subseteq \gtrsim^{\pi}_{rpo}$ と $R_{foldl} \subseteq \gtrsim^{\pi}_{rpo}$ を除去することができる．

残念ながら，文献 [5] で TRS の場合に対して述べられている理由と同様にこの実効規則の概念は一般には用いることができない．次の停止性を持つ STRS

$$\left\{ \begin{array}{rcl} f[0, 1, x] & \to & f[x, x, x] \end{array} \right.$$

に次の 2 つの規則を追加した STRS を考える．

$$\mathsf{c}[x, y] \to x, \quad \mathsf{c}[x, y] \to y$$

このとき，$f[\mathsf{c}[0, 1], \mathsf{c}[0, 1], \mathsf{c}[0, 1]] \xrightarrow{*} f[0, 1, \mathsf{c}[0, 1]] \to f[\mathsf{c}[0, 1], \mathsf{c}[0, 1], \mathsf{c}[0, 1]]$ という無限ループが発生する．この様に全く関係の無い規則の追加が，無限ループを持たなかった再帰に無限ループを発生させることがある．実はこの例で挙げた c のように引数を取り出す関数が無関係な関数の再帰に無限ループを発生させる関数の本質である．具体的には次で与えられる．

定義 4.2 各単純型 α に対し，$\tau(\mathsf{c}_\alpha) = \alpha \to \alpha \to \alpha$ となる特別な関数記号 c_α を考える．STRS C_e を $\mathsf{c}_\alpha[x, y] \to x, \mathsf{c}_\alpha[x, y] \to y$ の全体として定義する．

[1] 投稿版では，この文中で 2 個所使用されている “^{ex}” を書き忘れていたため補題 4.4(2) が成立しなくなっていました．申し訳ございませんでした．

実行規則の概念と STRS C_e を定理 3.11 に組み合わせると次の定理が導かれる．証明は定理 3.7 と次節で与える定理 4.9，そして定理 3.10 を組み合わせることで得られる．

定理 4.3 $DP_{SC}(R)$ が有限集合であり，各 $\mathcal{C} \in RC_{SC}(R)$ が以下のいずれかを満たす有限分岐 PFP-STRS R は停止性を持つ．

(i) $\mathcal{C}$ が部分項基準を満たす．

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

(iii) ある準簡約化対 $(\gtrsim, >)$ が存在し，
$\mathcal{C} \cup \mathcal{U}(\mathcal{C})^{ex} \cup C_e \subseteq \gtrsim$ かつ $\mathcal{C} \cap > \neq \emptyset$．

本定理を用いた場合，$R_{sub} \cup R_{div} \cup R_{foldl}$ における div の再帰成分について制約は以下の様になる．

$$\begin{array}{rcl} div^{\#}[(s[x], s[y])] & >^{\pi}_{rpo} & div^{\#}[(sub[(x, y)], s[y])] \\ R_{sub} & \subseteq & \gtrsim^{\pi}_{rpo} \\ C_e & \subseteq & \gtrsim^{\pi}_{rpo} \end{array}$$

ここで，各 c_α についての切り落としは行わない事で $C_e \subseteq \gtrsim^{\pi}_{rpo}$ は満たすようにできる．3 節で $R_{sub} \cup R_{div}$ の停止性を証明したときと同様に $\pi(tp) = [1], div \triangleright s \triangleright sub$ とすると，他の制約も満たされる．よって定理 4.3 により $R_{sub} \cup R_{div} \cup R_{foldl}$ の停止性が証明できる．これは定理 3.11 を用いて停止性を証明する際，妨げとなった制約 $R_{foldl} \subseteq \gtrsim$ が定理 4.3 では必要なくなったためである．

定理 3.11 では全ての再帰成分につき，STRS の規則すべてを取り扱う必要があったが，制約 $C_e \subseteq \gtrsim^{\pi}_{rpo}$ は事実上無視できるので，定理 4.3 では実効規則のみ取り扱えばよい．そのため証明能力が格段に向上し，また大幅な効率化を図ることができる．例えば，上記の例の場合は制約が 8 個から 4 個に半減している．一般に，停止性判定を行う STRS の規模が大きくなればなるほど，この効率化は効果的に機能する．

4.2 証明

本節では R を有限分岐 STRS，$\mathcal{C}$ を印付項の対の集合を表す記号に固定して用いる．また，$t \in \Delta$ であることを，ある $l \to r \in R \setminus \mathcal{U}(\mathcal{C})$ が存在し $root(t) = root(l)$ かつ $\tau(t) \sqsubseteq \tau(l)$ である，として定義する．直観的には，Δ は根の位置で $R \setminus \mathcal{U}(\mathcal{C})$ によって書換え可能な項からなる集合の近似である．

補題 4.4 任意の規則 $l \to r \in \mathcal{C} \cup \mathcal{U}(\mathcal{C})^{ex}$ と θ について，以下の性質が成立する．

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

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

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

証明 定義 4.1 より明らか． □

この補題の (1),(2) は補題 4.7 で，(3) は補題 4.6 で用いる．

次で定義する解釈 I が本証明の鍵となる．この様な解釈 I は停止性のモジュラー性を示す為に文献 [3] で与えられたアイデアを基に文献 [15] で与えられた．その後，一階の TRS の実効規則の正当性を示す為に用いられた [5, 14]．次の定義では，実効規則の定義における高階変数の詳細な解析に対応するために先に与えた Δ を用いる．この結果，高階変数を取り扱える STRS 上での実効規則の正当性を与えることが可能となる．

解釈 I を定義するにあたり，整列可能定理を用いる．整列可能定理より，項上の整礎な全順序が存在する．空でない項の集合 T のこの順序による最小元を $least(T)$ で記す．

定義 4.5 単純型 $\alpha \in \mathcal{S}$ に対して，$\tau(\bot_\alpha) = \alpha$, $\tau(\mathsf{c}_\alpha) = \alpha \to \alpha \to \alpha$ である新しい関数記号 $\bot_\alpha$，c_α を考える．解釈 I は $\mathcal{T}_\tau(\Sigma, \mathcal{V})$ 中の停止性を持つ項の全体から項 $\mathcal{T}_\tau(\Sigma \cup \bigcup_{\alpha \in \mathcal{S}} \{\bot_\alpha, \mathsf{c}_\alpha\}, \mathcal{V})$ への写像である．$\tau(t) = \alpha$ である $t \equiv a[t_1, \ldots, t_n]$ に対して $I(t)$ を以下で定義する．

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

ここで，$Red_\alpha(T)$ を $T = \emptyset$ のときは $\bot_\alpha$，$T \neq \emptyset$ のときは $\mathsf{c}_\alpha[least(T), Red_\alpha(T \setminus \{least(T)\})]$ で定義する．停止性をもつ代入 θ に対し，θ^I を $\theta^I(x) = I(\theta(x))$ で定義する．

関係 $\triangleright_{sub} \cup \xrightarrow[R]{}$ は停止性を持つ項上で整礎であり，R は有限分岐であるので集合 $\{I(t') \mid t \xrightarrow[R]{} t'\}$ は有限である．よって，解釈 I は矛盾なく定まる．

直観的には $I(t)$ は，t が $R \setminus \mathcal{U}(\mathcal{C})$ により簡約化可能な項を全て" 集め "，それらの項を記号 c_α を用いて 1 つの項で表現したものである．また規則 C_e によ

り，集められた項から任意の 1 つを取り出すことができる．その結果，考える書換え規則を $\mathcal{U}(\mathcal{C})$ と C_e のみとすることができる．

補題 **4.6** $l\theta$ が停止性を持つような規則 $l \to r \in \mathcal{C} \cup \mathcal{U}(\mathcal{C})^{ex}$ と代入 θ を考える．また，$x \in Var(l) \setminus Var(r)$ かつ $\theta(x) = a[u_1, \ldots, u_k]$ ならば $\sigma(x) = a[I(u_1), \ldots, I(u_k)]$，それ以外は $\sigma(x) = \theta^I(x)$ とする．このとき，$I(l\theta) \xrightarrow[C_e]{*} l\sigma$ である．

証明 $\forall t \in Sub(l).I(t\theta) \xrightarrow[C_e]{*} t\sigma$ を t の構造帰納法により示す．$t \equiv a[t_1, \ldots, t_n]$ とする．ここでは $a \in Var(l) \setminus Var(r)$ の場合と $a \in Var(r)$ かつ $a\theta \in \Delta$ の場合のみについて述べる．$a \in Var(l) \setminus Var(r)$ の場合，$a\theta \equiv a'[u_1, \ldots, u_k]$ とする．このとき σ の定義と帰納法の仮定より $I(t\theta) \equiv I(a'[u_1, \ldots, u_k, t_1\theta, \ldots, t_n\theta])$ $(\equiv \cup \xrightarrow[C_e]{})$ $a'[I(u_1), \ldots, I(u_k), I(t_1\theta), \ldots, I(t_n\theta)]$ $\xrightarrow[C_e]{*}$ $a'[I(u_1), \ldots, I(u_k), t_1\sigma, \ldots, t_n\sigma] \equiv t\sigma$.

$a \in Var(r)$ かつ $a\theta \in \Delta$ の場合，補題 4.4 (3) より $t \equiv a[\,]$ である．故に $I(t\theta) \equiv I(a\theta) \equiv a\theta^I \equiv t\sigma$． □

ここで θ^I ではなく σ を用いたことにより，定義 4.1(3) で $root(u') \in Var(v)$ の制限を追加でき，実効規則の数を削減することが可能となった．なお，これは高階変数を取り扱わない場合には考慮する必要が無い議論である．

補題 **4.7** $r\theta$ が停止性を持つような規則 $l \to r \in \mathcal{C} \cup \mathcal{U}(\mathcal{C})^{ex}$，代入 θ を考える．このとき，$I(r\theta) \equiv r\theta^I$．

証明 補題 4.6 と同様に示すことができる． □

補題 **4.8** $s \xrightarrow[R]{} t$ かつ s が停止性を持つならば $I(s) \xrightarrow[\mathcal{U}(\mathcal{C}) \cup C_e]{+} I(t)$．

証明 命題 2.1 より，$s \equiv C[l\theta]$ かつ $t \equiv C[r\theta]$ を満たすような規則 $l \to r \in R^{ex}$，葉文脈 $C[\,]$，代入 θ が存在する．$C[\,]$ の構造帰納法により示す．$C[\,]$ 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，$l \to r \in \mathcal{U}(\mathcal{C})^{ex}$．代入 σ を補題 4.6 と同様に定義すると，補題 4.6，4.7 より，$I(s) \equiv I(l\theta) \xrightarrow[C_e]{*} l\sigma \xrightarrow[\mathcal{U}(\mathcal{C})]{} r\sigma \equiv r\theta^I \equiv I(r\theta) \equiv I(t)$．
- $C[\,] \equiv a[\ldots, C'[\,], \ldots]$ かつ $s \notin \Delta$ の場合．このとき Δ の定義より $t \notin \Delta$，ゆえに帰納法の仮定より $I(s) \equiv I(C[l\theta]) \equiv a[\ldots, I(C'[l\theta]), \ldots] \xrightarrow[\mathcal{U}(\mathcal{C}) \cup C_e]{+} a[\ldots, I(C'[r\theta]), \ldots] \equiv I(a[\ldots, C'[r\theta], \ldots]) \equiv I(t)$．
- $s \in \Delta$ の場合．このとき，$I(s) \xrightarrow[C_e]{} Red(\{I(v) \mid s \xrightarrow[R]{} v\}) \xrightarrow[C_e]{+} I(t)$． □

定理 **4.9** 有限分岐 STRS R について $T^{args}_{SN}(R)$ 上の任意の $\langle \mathcal{C}, \xrightarrow[R]{} \rangle$-鎖は $T^{args}_{SN}(R)$ 上の $\langle \mathcal{C}, \xrightarrow[\mathcal{U}(\mathcal{C}) \cup C_e]{} \rangle$-鎖でもある．

証明 $\langle u_0^\#, v_0^\# \rangle \langle u_1^\#, v_1^\# \rangle \cdots$ を $T^{args}_{SN}(R)$ 上の $\langle \mathcal{C}, \xrightarrow[R]{} \rangle$-鎖とする．このとき各 i に対し，$u_i\theta_i, v_i\theta_i \in \mathcal{T}^{args}_{SN}(R)$ かつ $v_i^\#\theta_i \xrightarrow[R]{*} u_{i+1}^\#\theta_{i+1}$ となる $\theta_0, \theta_1, \ldots$ が存在する．また，$root(u_i^\#), root(v_i^\#) \notin \mathcal{D}_R$ より，$u_i^\#\theta_i, v_i^\#\theta_{i+1}$ は停止性を持つ．

ここで θ_i から σ_i を補題 4.6 と同様に定義する．このとき，$v_i\sigma_i, u_{i+1}\sigma_{i+1} \in T^{args}_{SN}(R)$ を得る．補題 4.6，4.7，4.8 より，$v_i^\#\sigma_i \equiv v_i^\#\theta_i^I \equiv I(v_i^\#\theta_i) \xrightarrow[\mathcal{U}(\mathcal{C}) \cup C_e]{+} I(u_{i+1}^\#\theta_{i+1}) \xrightarrow[C_e]{*} u_{i+1}^\#\sigma_{i+1}$．よって $\langle u_0^\#, v_0^\# \rangle \langle u_1^\#, v_1^\# \rangle \cdots$ は $T^{args}_{SN}(R)$ 上の $\langle \mathcal{C}, \xrightarrow[\mathcal{U}(\mathcal{C}) \cup C_e]{} \rangle$-鎖でもある． □

5 直積型項へのラベル付け

直積型を用いることでデータの組を自然に表現できる．しかし，強計算依存対法に基づいて STRS の停止性を証明する際，直積型が制約を解く妨げとなり証明能力の低下を引き起こすことがある．STRS では直積型を表現するために記号 tp を用いるが，本節では tp へトップダウンにラベル付けを行うことによりその問題を解決する手法を提案する．本手法により証明能力がさらに強力となる．なお，ラベル付け法は文献 [16] でも提案されているが，このラベル付けはボトムアップで行われている．

5.1 ラベル付けによる強計算依存対法の改良

以下，次で定義する STRS R_1 を用い議論を進める．

$$R_1 = R_{sub} \cup R_{div} \cup \left\{ \begin{array}{rcl} add[(0, y)] & \to & y \\ add[(s[x], y)] & \to & s[add[(x, y)]] \\ mul[(0, y)] & \to & 0 \\ mul[(s[x], y)] & \to & add[(mul[(x, y)], y)] \\ div[(div[(x, y)], z)] & \to & div[(x, mul[(y, z)])] \end{array} \right.$$

R_1 の停止性を定理 4.3 により証明する場合，問題となる再帰成分は

$$\left\{\begin{array}{l}\langle div^{\#}[(s[x],s[y])], div^{\#}[(sub[(x,y)],s[y])]\rangle\\ \langle div^{\#}[(div[(x,y)],z)], div^{\#}[(x,mul[(y,z)])]\rangle\end{array}\right\}$$

である．この再帰成分の制約において注目すべきは，

$$\begin{array}{rcl} add[(0,y)] & \gtrsim & y \\ div^{\#}[(s[x],s[y])] & \gtrsim & div^{\#}[(sub[(x,\underline{y})],s[y])] \\ div^{\#}[(div[(x,y)],\underline{z})] & \gtrsim & div^{\#}[(x,\underline{mul[(y,z)]})] \end{array}$$

を同時に順序付けなければならない点である．2 番目と 3 番目の制約はそのままでは順序が付けられないため，切り落とし関数の適用が必要である．よって，2 番目と 3 番目の制約の下線部を切り落とす，つまり $\pi(tp)=[1]$ として切り落としを行う．このとき，制約は以下の様になる．

$$\begin{array}{rcl} add[(0)] & \gtrsim & y \\ div^{\#}[(s[x])] & \gtrsim & div^{\#}[(sub[(x)])] \\ div^{\#}[(div[(x)])] & \gtrsim & div^{\#}[(x)] \end{array}$$

ところがこのとき 1 番目の制約を満たすことができない．これは切り落としが $sub, div^{\#}$ の下の tp のみならず add の下の tp でも行われた結果，変数 y が左辺から切り落とされたためである．

この問題を，ラベル付けによる tp の区別により解決する．具体的には文脈に依存したラベル付けを行い，制約を以下の様に変更する．

$$\begin{array}{l} add[(0,y)_{(add,1)}] \gtrsim y \\ div^{\#}[(s[x],s[y])_{(div^{\#},1)}] \gtrsim \\ \qquad div^{\#}[(sub[(x,\underline{y})_{(sub,1)}],s[y])_{(div^{\#},1)}] \\ div^{\#}[(div[(x,y)_{(div,1)}],\underline{z})_{(div^{\#},1)}] \gtrsim \\ \qquad div^{\#}[(x,\underline{mul[(y,z)_{(mul,1)}]})_{(div^{\#},1)}] \end{array}$$

このとき，切り落とし関数を $\pi(tp_{(sub,1)}) = \pi(tp_{(div^{\#},1)}) = [1]$ とすれば，下線部のみの切り落としに成功し，再帰経路順序によりこの制約を解くことができる．このようにして，制約の解法の選択肢を増加させても問題が起こらないことを示す．これにより，証明法をより強力なものとすることができる．

以降ではこのような直積型項へのラベル付け手法（定義 5.1）と，ラベル付けされた STRS と書換えによりラベル付けを行う補助 STRS A_R（定義 5.2）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 5.3）を示す．まずラベリング関数を以下で定義する．

定義 5.1 関数記号 a と位置 p の対 (a,p) または $\bot$ をポジション対と呼ぶ．$\bot$ は文脈情報を持たないことを表現する．ラベル付け関数 lab を以下で定義する．

$$lab(a[t_1,\ldots,t_n],\bot) = \left\{\begin{array}{ll} a[lab(t_1,(a,1)),\ldots,lab(t_n,(a,n))] & \text{if } a\in\Sigma\backslash\{tp\} \\ a[lab(t_1,\bot),\ldots,lab(t_n,\bot)] & \text{if } a\in\mathcal{V}\cup\{tp\}\end{array}\right.$$

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

$tp_{(a,p)}[t_1,\ldots,t_n]$ を $(t_1,\ldots,t_n)_{(a,p)}$ と表現することもある．また，STRS R に対し $lab(R) = \{lab(l,\bot)\to lab(r,\bot) \mid l\to r\in R\}$ とする．

ラベル付け後の STRS の停止性が元の STRS の停止性を保証するには，項 s,t に対して $s\xrightarrow[R]{} t$ ならば $lab(s,\bot)\xrightarrow[lab(R)]{+} lab(t,\bot)$ という書換え関係の模倣ができればよい．しかし，R_1 は $lab(R_1)$ で模倣できるが，この模倣は全ての STRS で可能な訳ではない．実際，次の STRS R_2 は $lab(R_2)$ で模倣できない．

$$R_2 = \left\{\begin{array}{rcl} id[(x,y)] & \to & (x,y) \\ fst[(x,y)] & \to & x \\ foo[(f,x)] & \to & f[(x,x)]\end{array}\right.$$

$$lab(R_2) = \left\{\begin{array}{rcl} id[(x,y)_{(id,1)}] & \to & (x,y) \\ fst[(x,y)_{(fst,1)}] & \to & x \\ foo[(f,x)_{(foo,1)}] & \to & f[(x,x)]\end{array}\right.$$

このとき，$lab(foo[(id,0)],\bot)\equiv foo[(id,0)_{(foo,1)}] \xrightarrow[lab(R_2)]{} id[(0,0)] \not\equiv lab(id[(0,0)],\bot)$ のように $lab(R_2)$ では模倣ができない例がある．これは 3 番目の規則の右辺において，高階変数 f の下に tp が現れていることによる．また，$lab(fst[id[(0,0)]],\bot) \equiv fst[id[(0,0)_{(id,1)}]]\xrightarrow[lab(R_2)]{} fst[(0,0)] \not\equiv lab(fst[(0,0)],\bot)$ のような模倣ができない例も存在する．これは tp が R_2 の第一規則の右辺の根に出現していることによる．これらの問題は規則の適用時毎に文脈が変化するため，規則のみからでは文脈が確定できないことによる．この問題に対処するため，補助 STRS A_Σ を与える．

定義 5.2 単純型 α に対し，$Aux(\alpha)$ を以下で定義．

$$\left\{\begin{array}{ll} tp[Aux(\beta_1),\ldots,Aux(\beta_n)] & \text{if } \alpha=\beta_1\times\cdots\times\beta_n \\ z & \text{otherwise}\end{array}\right.$$

ここで，z は型 α の新しい変数である．補助 STRS

A_Σ を以下で定義する．

$$A_\Sigma = \{Aux(\alpha_i) \to lab(Aux(\alpha_i),(a,i)) \mid a \in \Sigma, \tau(a) = \alpha_1 \to \cdots \to \alpha_n \to \beta, \alpha_i \text{は直積型}\}$$

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 に関しては許すことにする．

$\Sigma = \{id, fst, foo\}$ としたとき，A_Σ は以下のようになる．

$$\begin{cases} (z_1,z_2) \to (z_1,z_2)_{(id,1)} \\ (z_1,z_2) \to (z_1,z_2)_{(fst,1)} \\ (z_1,z_2) \to (z_1,z_2)_{(foo,1)} \end{cases}$$

これにより先ほど不可能であった R_2 の書換え関係の模倣が $lab(R_2) \cup A_{\Sigma_2}$ により可能となる．

しかしながら，補助 STRS を用いても模倣が不可能な場合が存在する．1 つは堅固でない，すなわち，葉でない位置に変数が出現する左辺項が存在する場合，もう 1 つは直積型をもつ変数が存在する場合である．これらの場合を除けば，模倣が可能となる．

定理 5.3 $DP_{SC}(R)$ が有限集合であり，以下を満たす有限分岐 PFP-STRS R は停止性を持つ．

(1) R^{ex} に現れる変数はいずれも直積型を持たない

(2) 各規則 $l \to r \in R$ の左辺 l は堅固である

(3) 各 $\mathcal{C} \in RC_{SC}(R)$ が，以下のいずれかを満たす

- $\mathcal{C}$ が部分項基準を満たす．
- ある簡約化対 $(\gtrsim, >)$ が存在し，$lab(\mathcal{C}) \cup lab(\mathcal{U}(\mathcal{C})) \cup C_e \cup A_\Sigma \subseteq \gtrsim$ かつ $lab(\mathcal{C}) \cap > \neq \emptyset$．
- ある準簡約化対 $(\gtrsim, >)$ が存在し，$lab(\mathcal{C}) \cup lab(\mathcal{U}(\mathcal{C})^{ex}) \cup C_e \cup A_\Sigma \subseteq \gtrsim$ かつ $lab(\mathcal{C}) \cap > \neq \emptyset$．

本定理は定理 4.3 と次節で与える補題 5.7，5.8 より導かれる．

5.2 証明

補題 5.4 A_Σ を補助 STRS とする．任意の項 t と任意のポジション対 (a,p) に対し，$lab(t,\bot) \xrightarrow[A_\Sigma]{*} lab(t,(a,p))$．ただし，$t$ が直積型の場合，(a,p) は A_Σ に出現するとする．

証明 t に関する構成帰納法で証明する．$t \equiv tp[t_1,\ldots,t_n]$ の場合のみ示す．

$$\begin{aligned} & lab(tp[t_1,\ldots,t_n],\bot) \\ & \equiv tp[lab(t_1,\bot),\ldots,lab(t_n,\bot)] \\ & \xrightarrow[A_\Sigma]{*} tp[lab(t_1,(a,p1)),\ldots,lab(t_n,(a,pn))] \\ & \xrightarrow[A_\Sigma]{} tp_{(a,p)}[lab(t_1,(a,p1)),\ldots,lab(t_n,(a,pn))] \\ & \equiv lab(tp[t_1,\ldots,t_n],(a,p)) \end{aligned}$$

□

補題 5.5 補助 STRS A_Σ，項 t，代入 θ を考える．また，t に出現するどの変数も直積型を持たないとし，$\theta(x) = a'[t'_1,\ldots,t'_k]$ であるとき $\theta_{lab}(x)$ を以下で定義する．

$$\begin{cases} a'[lab(t'_1,(a',1)),\ldots,lab(t'_k,(a',k))] & \text{if } d \in \Sigma\backslash\{tp\} \\ a'[lab(t'_1,\bot),\ldots,lab(t'_k,\bot)] & \text{if } d \in \mathcal{V}\cup\{tp\} \end{cases}$$

このとき，任意のポジション対 P に対して，$lab(t,P)\theta_{lab} \xrightarrow[A_\Sigma]{*} lab(t\theta,P)$ が成立する．さらに，t が堅固の場合には，$lab(t,P)\theta_{lab} \equiv lab(t\theta,P)$ が成り立つ．

証明 t に関する構造帰納法で $lab(t,P)\theta_{lab} \xrightarrow[A_\Sigma]{*} lab(t\theta,P)$ を証明する．なお，t が堅固である場合に $lab(t,P)\theta_{lab} \equiv lab(t\theta,P)$ が成立することも同様に証明できる．$t \equiv b[t_1,\ldots,t_n], \theta(b) = b'[t'_1,\ldots,t'_m]$ とする．

$b \in \Sigma$, または $b \in V$ かつ $b' \in \mathcal{V}$ の場合は明らか．また，条件より直積型を持つ変数は出現しないので，$b \in \mathcal{V}$ かつ $b' = tp$ の場合は存在しない．

$P = (a,p)$, $b \in \mathcal{V}$, $b' \in \Sigma\backslash\{tp\}$ の場合は

$\theta_{lab}(b) \equiv b'[lab(t'_1,(b',1)),\ldots,lab(t'_m,(b',m))]$ より，

$$\begin{aligned} & lab(b[t_1,\ldots,t_n],(a,p))\theta_{lab} \\ & \equiv b[lab(t_1,\bot),\ldots,lab(t_n,\bot)]\theta_{lab} \\ & \equiv b'[lab(t'_1,(b',1)),\ldots,lab(t'_m,(b',m)), \\ & \qquad lab(t_1,\bot)\theta_{lab},\ldots,lab(t_n,\bot)\theta_{lab}] \end{aligned}$$

ここで t_i が直積型のときは，$b' \in \Sigma\backslash\{tp\}$ より $(b', m+i)$ は A_Σ に出現．よって補題 5.4 より

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

$P = \bot$, $b \in \mathcal{V}$, $b' \in \Sigma\backslash\{tp\}$ の場合も同様に示すことができる． □

補題 **5.6** 補助 STRS A_Σ，項 t，葉文脈 $C[\,]$ を考える．また，$C_{lab}[\,] = lab(C[\,], \bot)$ とする．このとき，$C_{lab}[lab(t, \bot)] \xrightarrow[A_\Sigma]{*} lab(C[t], \bot).$ が成立する．特に，$root(t) \neq tp$ ならば，$C_{lab}[lab(t, \bot)] \equiv lab(C[t], \bot)$ が成立する．

証明 $C_{lab}[\,]$ 中の $\Box$ がポジション対 (a, p) によりラベル付けされる，すなわち，$lab(C[t], \bot) \equiv C_{lab}[lab(t, (a, p))]$ とする．補題 5.4 より，$C_{lab}[lab(t, \bot)] \xrightarrow[A_\Sigma]{*} C_{lab}[lab(t, (a, p))] \equiv lab(C[t], \bot)$ である．また，$root(t) \neq tp$ であるとき $lab(t, \bot) \equiv lab(t, (a, p))$ より $C_{lab}[lab(t, \bot)] \equiv lab(C[t], \bot)$ が成立する． $\Box$

補題 **5.7** STRS R を考える．以下を共に満たすとき，$s \xrightarrow[R]{} t$ ならば $lab(s, \bot) \xrightarrow[lab(R) \cup A_\Sigma]{+} lab(t, \bot)$．

(1) R^{ex} に現れる変数は全て直積型を持たない

(2) 各規則 $l \to r \in R$ について l は堅固

証明 命題 2.1，補題 5.5，補題 5.6 より明らか． $\Box$

補題 **5.8** 任意の STRS R に関して，$lab(RC_{SC}(R)) = RC_{SC}(lab(R) \cup A_\Sigma)$．

証明 A_Σ の定義より明らか． $\Box$

6 おわりに

本論文では実効規則と直積型項へのラベル付け法の導入により，強計算依存対法の証明能力を高めた．

定義 4.1(3) では実効規則の生成時に左辺の高階変数に隠れた関数呼び出しまで解析する必要がある．これは直観とは合わない定義であるが，補題 4.6 の証明に当たっては必要となった．実効規則の直観的意味から考えると，定義 4.1 における (3) は省けると推測される．これは今後の課題である．

1 節にて紹介した組み合わせ子論理の項書換え系の停止性は，強計算性を導入していない経路順序を用いた手法 [10] 等や依存対法では示すことができない．それに対し，強計算依存対法以外にも，強計算性と経路順序を融合・発展させた Computational Closure($\mathcal{CC}$) を用いた高階再帰経路順序 [6] でも型付き組み合わせ子論理の停止性証明が可能である．汎用的な停止性証明法で型付き組み合わせ子論理の停止性を示すことができるのは我々が知る限りこれら 2 つの手法のみである．これら 2 つの手法の比較は進んでいないが，これら 2 つの手法を融合するための研究がより重要であると考える．なぜならば，それぞれの手法の土台となった 1 階の TRS における経路順序と依存対法は切り落とし法を通して互いに協調的に機能するからである [2]．よって，高階においても同様の協調効果が期待できる．文献 [6] では再帰経路順序と強計算性を融合することで高階再帰経路順序を提案し，更に $\mathcal{CC}$ を用いることで改良している．$\mathcal{CC}$ を用いない高階再帰経路順序は文献 [8] で STRS 上に移植され，切り落とし法の適用に関する研究が行われている．しかしながら，$\mathcal{CC}$ を用いた高階再帰経路順序への切り落とし法の適用に関する研究は行われていない．切り落とし法は順序付けを阻害する引数を切り落とすという手法であるため項の型の整合性を崩してしまう．よって $\mathcal{CC}$ の複雑な定義と組み合わせることは困難であることが予測される一方で，もし $\mathcal{CC}$ を用いた高階再帰経路順序への切り落とし法の適用法が確立されれば，非常に強力な停止性証明法が期待できる．これは今後の課題である．

謝辞

本研究は，科研費#16650005，#17700009，#18500011，ならびに人工知能研究振興財団，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．

References

[1] T.Aoto and T.Yamada, Dependency pairs for simply typed term rewriting, Procs. 16th International Conference on Rewriting Techniques and Applications, LNCS 3467, pp.120-134, 2005.

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

[3] B.Gramlich, Generalized sufficient conditions for modular termination of rewriting, Applicable Algebra in Engineering, Communication and Computing, vol.5, pp.131–158, 1994.

[4] J.R.Hindley, J.P.Seldin, Introduction to Combinators and λ-Calculus, *Cambridge Univ. Press*, 1986.

[5] N.Hirokawa and A.Middeldorp, Dependency Pairs Revisited, Proceedings of the 15th International Conference on Rewriting Techniques and Applications, Aachen, LNCS 3091, pp.249–268, 2004.

[6] J.Jouannaud and A.Rubio, Higher-Order Recursive Path Orderings, Procs. 14th Annual IEEE Symposium on Logic in Computer Science, pp.402–411, 1999.

[7] K.Kusakari, 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.

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

[9] K.Kusakari and M.Sakai, Enhancing Dependency Pair Method by Strong Computability in Simply-Typed Term Rewriting Systems, IEICE Tech. Rep. (SS2005-65), Vol.105, No.491, pp.13–18, 2005.

[10] M.Lifantsev, L.Bachmair, An LPO-based Termination Ordering for Higher-Order Terms without lambda-abstraction, TPHOLs 1998, LNCS 1479, pp.277-293, 1998.

[11] J.Reynolds, Definitional Interpreters for Higher-Order Programming Languages, Higher Order Symbolic Computation, 11(4), pp.363–397, 1998. Reprinted from the Proc. of the 25th ACM National Comference, 1972.

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

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

[14] R.Thiemann, J.Giesl, and P.Schneider-Kamp, Improved Modular Termination Proofs Using Dependency Pairs, IJCAR '04, LNAI 3097, pp.75–90, 2004.

[15] X.Urbain, Modular & incremental automated termination proofs, Journal of Automated Reasoning, vol.31, No.4, pp.315–355, 2004.

[16] H.Zantema, Termination of Term Rewriting by Semantic Labelling, Fundamenta Informaticae, vol.24, pp.89–105, 1995.