論　文

消去法による項書換え系の停止性判定について

中村　正樹[†]　　草刈圭一朗[†]　　外山　芳人[†]

On Proving Termination by General Dummy Elimination

Masaki NAKAMURA[†], Keiichirou KUSAKARI[†], and Yoshihito TOYAMA[†]

あらまし
従来の停止性判定法を適用できない項書換え系 (TRS) の停止性を示すために，Ferreira は一般消去法を提案した．一般消去法は，規則に現れる適当な関数記号を消去することでより単純な構造の TRS に変換する．変換後の TRS の停止性が元の TRS の停止性を保証するため，停止性判定に有効である．一方，Arts と Giesl は，TRS の規則に現れる関数定義記号の出現に注目し，無限書換え列の本質を簡潔に記述できる依存対の概念を導入した．本稿では，一般消去法が変換後に付け加える規則に注目し，不必要な規則を除くことで改良一般消去法を定義する．改良一般消去法の正当性は，依存対の概念を用いることで示す．

キーワード　項書換え系，停止性，変換，消去法，依存対．

1. 序

等式推論は，定理自動証明，仕様記述，関数型言語などの計算機科学のさまざまな分野で広く使われている．項書換え系 (TRS) は，等式に方向付けを行なうことで等式推論を効率的に実現するための基礎を与える [3].

TRS の重要な性質のひとつに停止性がある．TRS を計算モデルとしてみる場合，停止性は解の存在を保証する．TRS が停止性を持つかどうかは決定不能な問題であるため，停止性の十分条件に関する研究が活発に行なわれている [1] [2] [4]〜[7].

停止性判定には，大きく分けて意味的な手法と構文的な手法がある [3]．意味的な手法は，項を整礎な順序を持つ集合上に解釈し，書換えによって対応する集合上の要素が真に小さくなることで停止性を保証する．全ての停止性を持つ TRS に対して，そのような整礎な集合が存在することが分かっている [6]．一方，構文的な手法では，規則に現れる項の構造を解析し左辺より右辺の方が小さくなるように項上の整礎な順序を定義することで停止性を示す．構文的手法の代表的なものに再帰経路順序 (RPO) による順序付けがある [3]．項がその部分項よりも真に大きくなるような順序を単純化順序と呼ぶ．RPO は，単純化順序となっているため，停止性を示すことのできる TRS が限られている．しかし，RPO による二項間の順序付けは決定可能なため停止性判定の実装に適している．

TRS の変換は，停止性判定の有効な手法のひとつである [4]〜[7]．変換手法は，複雑な構造の TRS を単純化することで停止性判定を容易にする．以下のような TRS R_0 は単純化順序で規則に順序付けできないので RPO によって停止性を示すことができない．

$$R_0 = \{f(f(x)) \to f(g(f(x)))\}$$

TRS R_0 は，変換手法のひとつである置換消去法によって関数記号 g を消去することで TRS $\mathcal{E}(R_0)$ に変換される．

$$\mathcal{E}(R_0) = \begin{cases} f(f(x)) & \to & f(\diamond) \\ f(f(x)) & \to & f(x) \end{cases}$$

置換消去法には，変換後の $\mathcal{E}(R)$ が停止性を持つならば R は停止性を持つという性質がある [4]．TRS $\mathcal{E}(R_0)$ は，RPO による規則の順序付けによって停止性を示すことができる．置換消去法による変換は，自動で行なうことができるので実装が容易であり，決定可能な停止性のクラスを広げることができるため有効である．

[†] 北陸先端科学技術大学院大学 情報科学研究科，石川県
School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa, 923-1292, Japan

一方，近年提案された依存対 [1] [2] は，TRS の無限書換え列の解析に有効な概念である．依存対により規則の中の関数定義記号の出現を記述することで，無限書換え列の性質を簡明に記述できる．本研究では，構文的な変換手法として代表的な一般消去法の改良を試みる．具体的には，変換後の規則に現れる関数定義記号の出現に注目し，停止性に関する変換の正当性に不必要な規則を除去することで改良を行なう．改良された消去法の正当性は依存対の概念を用いて示す．

本稿は以下のように構成されている．次章では，TRS とその停止性についての基礎的な説明を行なう．特に置換消去法，分配消去法の一般化である一般消去法，及び依存対について詳しく紹介する．３章で，既存の一般消去法は不必要な規則を付け加えることがあることを例を挙げて説明し，一般消去法の改良を提案する．さらに，改良一般消去法の定義を示し，その証明を依存対を用いて行なう．最後に４章で，本研究の成果をまとめ，今後の課題を述べる．

2. 準　　備

項書換え系 (TRS) に関する諸定義は，文献 [3] [4] による．

2.1 項書換え系

[**定義** 2.1] 項は，関数記号 $f, g, \ldots$ の集合 F と変数 $x, y, \ldots$ の集合 V から再帰的に定義される．

(1) 変数 $x \in V$ は項である．
(2) 関数記号 $f \in F$, 項 $t_1, t_2, \ldots, t_n$ に対して，$f(\overrightarrow{t_n})$ は項である
（但し，$f(\overrightarrow{t_n})$ は $f(t_1, \ldots, t_n)$ の略記）．

項 $s, t, u, \ldots$ の集合を $T(F, V)$ で表す．規則 $l \to r$ は，項 l が変数でなく，項 r に現れる変数が必ず項 l にも出現するような項の対である．項書換え系 (TRS) は，規則の集合 R で記述される．

代入 $\theta : V \to T(F, V)$ は，変数から項への写像で，項から項への写像に自然に拡張される．

$$\theta(t) = \begin{cases} \theta(t) & t \in V \\ f(\overrightarrow{\theta(t_n)}) & t = f(\overrightarrow{t_n}) \end{cases}$$

以後，$\theta(t)$ を $t\theta$ と書く．文脈 $C[\,]$ とは，特別な定数 $\square$ をただ一つ含む項で，文脈 $C[\,]$ の $\square$ の部分を項 t で置き換えたものを $C[t]$ で表す．また，$s \equiv t$ と書いた時は項 s, t は同じ項であるとする．TRS R における書換え関係 $\underset{R}{\to}$ を以下のように定義する．

[**定義** 2.2] $s \underset{R}{\to} t \overset{\text{def}}{\Longleftrightarrow}$ ある代入 θ，文脈 $C[\,]$，規則 $l \to r \in R$ が存在して，$s \equiv C[l\theta], t \equiv C[r\theta]$.

TRS R が停止性を持つとは，無限書換え列 $t_1 \underset{R}{\to} t_2 \underset{R}{\to} \cdots$ が存在しないことである．書換え関係 $\underset{R}{\to}$ の反射・推移閉包を $\underset{R}{\overset{*}{\to}}$，推移閉包を $\underset{R}{\overset{+}{\to}}$ で記述する．

[**例** 2.3] 論理式の真偽を判定するための TRS R_1 を考える．

$$R_1 = \begin{cases} and(x, x) & \to & x \\ and(x, F) & \to & F \\ and(F, x) & \to & F \\ not(T) & \to & F \\ not(F) & \to & T \end{cases}$$

TRS R_1 により，以下のような書換え列が存在する．

$$not(and(not(F), F)) \underset{R_1}{\to} not(F) \underset{R_1}{\to} T$$

TRS R_1 は，項に現れる関数記号の数が書換えによって減少するため停止性を持っている．

停止性の判定は，項の上に整礎な二項関係を定義することで行なうのが一般的である．二項関係 $\rhd$ に対して，以下の性質を定義する．

任意の代入 θ，項 s, t に対して，$s \rhd t$ ならば $s\theta \rhd t\theta$ となるとき $\rhd$ は代入に閉じているという．任意の文脈 $C[\,]$，項 s, t に対して，$s \rhd t$ ならば $C[s] \rhd C[t]$ となるとき $\rhd$ は文脈に閉じているという．

項上の整礎な半順序 $>$ が代入と文脈に閉じているとき，書換え順序と呼び，擬順序 $\gtrsim$ が代入と文脈に閉じ，厳格部分 $>= (\gtrsim \setminus \lesssim)$ が代入に閉じて整礎なとき，弱書換え順序と呼ぶ．

[**定理** 2.4] ([3]) TRS R が停止性を持つことと，任意の規則 $l \to r \in R$ に対して $l > r$ となる書換え順序 $>$ が存在することは同値である．

2.2 -代　　数

F-代数は，TRS の停止性判定に有効な概念である．F-代数 $(A, >)$ は，ある空でない集合 A と A 上の半順序 $>$ の対で，各関数記号 $f \in F$ に対して写像 $f_A : A^n \to A$ が定義されているものである．

変数の割り当て $\sigma : V \to A$ を以下のように項上に拡張する．

$$[\sigma](t) = \begin{cases} \sigma(t) & t \in V \\ f_A(\overrightarrow{[\sigma](t_n)}) & t = f(\overrightarrow{t_n}) \end{cases}$$

項上へ拡張された割り当て $[\sigma]$ によって変数を持た

ない任意の項は集合 A の要素に解釈される．任意の $f \in F$ に対して，$a > b\,(a \geqq b)$ ならば $f_A(...,a,...) > f_A(...,b,...)\,(\,f_A(...,a,...) \geqq f_A(...,b,...)\,)$ であるとき F-代数は単調性 (弱単調性) を持つという．また，任意の $l \to r \in R$，任意の割り当て $\sigma : V \to A$ に対して，$[\sigma](l) > [\sigma](r)$ となるとき F-代数 $(A,>)$ は TRS R と両立するという．

F-代数 $(A,>)$ に対して二項関係 $>_A$，$\gtrsim_A$ を定義する．

$$s >_A t \overset{\text{def}}{\iff} \forall \sigma : V \to A, [\sigma](s) > [\sigma](t)$$
$$s \gtrsim_A t \overset{\text{def}}{\iff} \forall \sigma : V \to A, [\sigma](s) \geqq [\sigma](t)$$

F-代数 $(A,>)$ が TRS R と両立し整礎かつ単調 (弱単調) ならば $>_A$ ($\gtrsim_A$) は書換え順序 (弱書換え順序) になる．定理 2.4 より以下の定理が成り立つ．

[**定理** 2.5]（Zamtema [6]） TRS R が停止性を持つことと，R と両立する整礎で単調な F-代数が存在することは同値である(注1)．

2.3 消 去 法

TRS の構文的な変換手法のひとつに消去法がある．[4] [5] [7]．消去法は，規則の中の適当な関数記号を消去することで規則を単純化し停止性判定を容易にする．消去法によって変換して得られた TRS が停止性などの条件を満たせば元の TRS も停止性を持つ．これまでに置換消去法 [5] や分配消去法 [7]，一般消去法 [4] が提案されている．一般消去法は，置換消去法と分配消去法を融合させた消去法である．本研究では，特に一般消去法を考察する．一般消去法に関する諸定義は文献 [4] による．

[**定義** 2.6] 状態関数 $\tau : F \to \mathcal{P}(\mathcal{N}) \times \mathcal{N}$ は引数が n の関数記号 f に対して，$\tau(f) = (I,i)$ で $I = \emptyset, i = 0$，または，$I \neq \emptyset, I \subset \{1,2,...,n\}, i \in I$ を満たすものとする．

消去する関数記号を e，その状態を $\tau(e) = (I,i)$ とする．このとき項 t のキャップ部分 $cap_i(t)$，残余部分 $E(t)$，分解部分 $dec(t)$ を以下のように定義する（以下では，$f \neq e$ とする）．

[**定義** 2.7] $cap_i : T(F,V) \to T(F,V)$,

$$cap_i(t) = \begin{cases} t & t \in V \\ f(\overrightarrow{cap(t_n)}) & t = f(\overrightarrow{t_n}) \\ cap_i(t_i) & t = e(\overrightarrow{t_n}), i \neq 0 \\ \diamond & t = e(\overrightarrow{t_n}), i = 0 \end{cases}$$

但し，$\diamond \in F$ は新たな定数とする．

[**定義** 2.8] $E, E_i : T(F,V) \to \mathcal{P}(T(F,V))$,

$$E_i(t) = \begin{cases} \{t\} & t \in V \\ \{f(\overrightarrow{s_n}) | s_k \in E_i(t_k)\} & t = f(\overrightarrow{t_n}) \\ E(t_i) & t = e(\overrightarrow{t_n}) \end{cases}$$

$$E(t) = \begin{cases} \{t\} & t \in V \\ \{cap_0(t)\} & I = \emptyset \\ \bigcup_{k \in I} E_k(t) & I \neq \emptyset \end{cases}$$

[**定義** 2.9] $dec : T(F,V) \to \mathcal{P}(T(F,V))$,

$$dec(t) = \begin{cases} \phi & t \in V \\ \bigcup_n dec(t_i) & t = f(\overrightarrow{t_n}) \\ \bigcup_n dec(t_i) \cup \bigcup_{k \notin I} E(t_k) & t = e(\overrightarrow{t_n}) \end{cases}$$

一般消去法は，このように定義されたキャップ部分，残余部分，分解部分によって次のように定義される．

[**定義** 2.10] $\tau(e) = (I,i)$ のとき，

$$\mathcal{E}(R) = \{ \quad cap_i(l) \to u \mid l \to r \in R, \\ u \in E(r) \cup dec(r)\}$$

.

消去する関数記号 e の状態により部分項の取り扱いが決まる．$\tau(e) = (I,i)$ のとき，I の要素に対応する部分項を残余し，それ以外は分解する．項 t のキャップ部分 $cap_i(t)$ は，項 t における e の出現を，$i = 0$ ならば定数 $\diamond$ に (図 1)，$i \neq 0$ ならば e の直下で i 番目の部分項に (図 2) それぞれ置き換えて得られる項である．残余部分 $E(t)$ は，項 t における e の出現を，$I = \emptyset$ ならば定数 $\diamond$(図 1) に，$I \neq \emptyset$ ならば I の要素に対応する e の直下の部分項 (図 2) にそれぞれ置き換えて得られる項の集合である．分解部分 $dec(t)$ は，I の要素でない e の直下の部分項 t の残余部分 $E(t)$ の和集合である．

一般消去法で得られる $\mathcal{E}(R)$ は各規則 $l \to r \in R$ に対して左辺 $cap_i(l)$，右辺は集合 $E(r) \cup dec(r)$ の要素であるような規則からなる．置換消去法は，部分項の扱いが全て分解になったもの $(\tau(e) = (\emptyset, 0))$，分配消去法は，全て残余したもの $(\tau(e) = (\{1,2,...,n\}, i))$ に対応する．従って一般消去法は，置換消去法，分配

(注1)：TRS R が停止性を持つならば，F-代数 $(T(F,V),\ \frac{+}{R}\ )$ は R と両立し整礎で単調である．

消去法を含んでいる．また，分解，残余を混在させることもできるので置換消去法でも分配消去法でも扱えなかった TRS にも適用できる．

一般消去法について次の定理が成り立つ．

［**定理** 2.11］（Ferreira [4]） $\mathcal{E}(R)$ が停止性を持つならば，R は停止性を持つ．

［**例** 2.12］ 一般消去法により，TRS R_2 を $\tau(g) = (\{1\}, 1)$ で変換する．右辺 r のキャップ部分，残余部分，分解部分，及び $\mathcal{E}(R_2)$ は以下のようになる．

$$R_2 = \{f(x,x) \to f(g(a,c), g(b,c))\}$$

$$\begin{aligned} cap_i(r) &= f(a,b) \\ E(r) &= \{f(a,b)\} \\ dec(r) &= \{c\} \\ \mathcal{E}(R_2) &= \begin{cases} f(x,x) \to f(a,b) \\ f(x,x) \to c \end{cases} \end{aligned}$$

TRS $\mathcal{E}(R_2)$ は明らかに停止性を持つので定理 2.11 より R_2 は停止性を持つ．

2.4 依 存 対

TRS の規則の左辺の最外の関数記号を関数定義記号と呼ぶ．依存対は，規則の中の関数定義記号の出現を記述したものである．依存対によって定義される依存対列は，無限書換え列をその本質を失うことなく簡明に記述できる．依存対に関する諸定義は文献[1][2]による．

［**定義** 2.13］ R の依存対の集合 $DP(R)$ を以下で定義する．関数定義記号の集合を $D_R = \{root(l) | l \to r \in R\}$ とする．

$$DP(R) = \{ \quad \langle f^{\#}(\overrightarrow{t_n}), g^{\#}(\overrightarrow{s_m}) \rangle \mid g \in D_R, f(\overrightarrow{t_n}) \to C[g(\overrightarrow{s_m})] \in R\}$$

ここで，関数記号の名前変えのために $F^{\#} = F \cup$

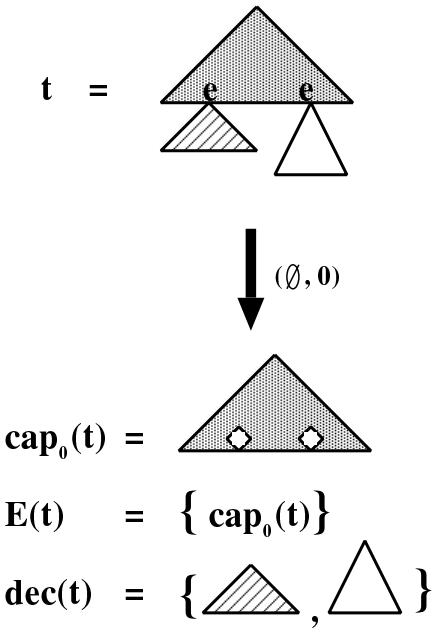

図 1 $\tau(e) = (\emptyset, 0)$ でのキャップ部分 $cap_0(t)$，残余部分 $E(t)$，分解部分 $dec(t)$
Fig. 1 $cap_0(t)$,$E(t)$, $dec(t)$ ($\tau(e) = (\emptyset, 0)$)

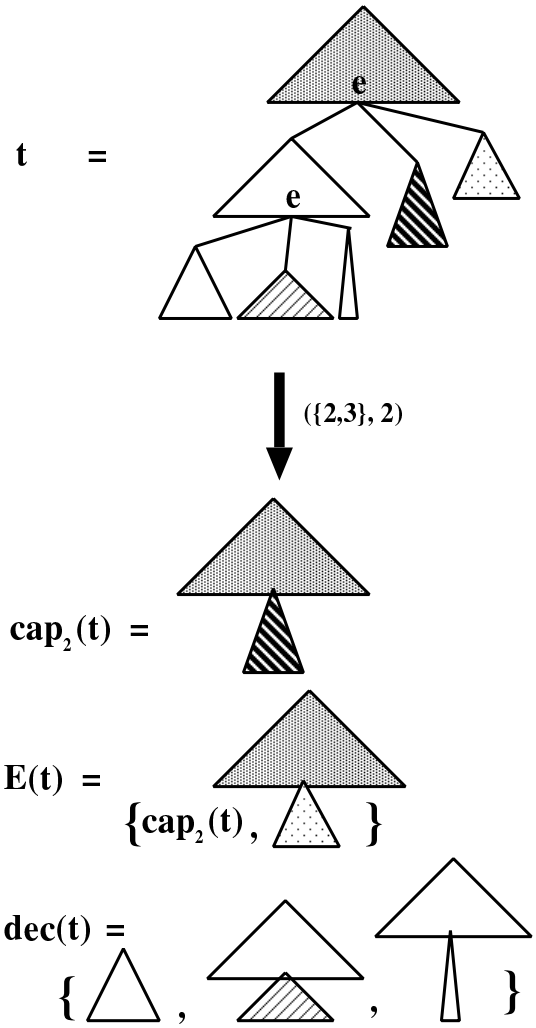

図 2 $\tau(e) = (\{2,3\}, 2)$ でのキャップ部分 $cap_2(t)$，残余部分 $E(t)$，分解部分 $dec(t)$
Fig. 2 $cap_2(t)$,$E(t)$, $dec(t)$ ($\tau(e) = (\{2,3\}, 2)$)

$\{f^{\#}|f \in F\}$ を新たに導入する．また $t = f(\overrightarrow{t_n})$ のとき，$t^{\#} = f^{\#}(\overrightarrow{t_n})$ とする．以下では，$\langle u^{\#}, v^{\#}\rangle \in DP(R)$ を書換え規則 $u^{\#} \to v^{\#}$ とみなすことにより，$DP(R)$ を項書換え系と取り扱う場合がある．

［**定義** 2.14］ 依存対列とは，変数を共有しない依存対の列 $\langle u_1^{\#}, v_1^{\#}\rangle\langle u_2^{\#}, v_2^{\#}\rangle\langle u_3^{\#}, v_3^{\#}\rangle...$ であり，ある代入 θ に対して $v_i^{\#}\theta \xrightarrow[R]{*} u_{i+1}^{\#}\theta$ を満たすものである．

依存対は次の性質を持つ．

［**定理** 2.15］(Arts, Giesl [1][2]) TRS が停止性を持つことと無限依存対列が存在しないことは同値である．

［**定理** 2.16］(Arts,Giesl [1][2]) 次を満たす弱書換え順序 $\gtrsim$ が存在するならば TRS R は停止性を持つ．

(1) 任意の $l \to r \in R$ に対して,$l \gtrsim r$.
(2) 任意の $\langle u^{\#}, v^{\#}\rangle \in DP(R)$ に対して,$u^{\#} > v^{\#}$.

［**補題** 2.17］(Arts,Giesl [2]) TRS R が停止性を持つならば,(依存対を規則とみなした) TRS $R \cup DP(R)$ は停止性を持つ．

3. 拡　　張

停止性を持つにも関わらず，一般消去法で停止性のある TRS に変換できない例が存在する．

$$R_3 = \left\{ \begin{array}{rcl} f(a) & \to & f(b) \\ b & \to & g(a) \end{array} \right.$$

この TRS R_3 はどのような関数記号 e，状態 $\tau(e)$ を取っても停止性のある $\mathcal{E}(R_3)$ に変換できない．

例えば，関数記号 g を状態 $\tau(g) = (\emptyset, 0)$ で消去して得られた以下の $\mathcal{E}(R_3)$ は停止性を持たない．

$$\mathcal{E}(R_3) = \left\{ \begin{array}{rcl} f(a) & \to & f(b) \\ b & \to & \diamond \\ b & \to & a \end{array} \right.$$

一般消去法の定義より，$\tau(g)$ をどのように定めても TRS $\mathcal{E}(R_3)$ には，消去する関数記号 g の直下の部分項 a が，それを右辺とするような規則 $b \to a$ の形で保存される．TRS $\mathcal{E}(R_3)$ が停止性を持たないのは規則 $b \to a$ が加えられているからである．以下では，このような規則を加えないように定義しなおすことで一般消去法を改良する．

3.1 改良一般消去法

改良一般消去法を以下のように定義する．

［**定義** 3.1］ 関数定義記号が出現している項の集合を T_{D_R} とする．

$$\begin{array}{rcl} E'(t) & = & \{s \mid s \in E(t) \cap T_{D_R}\} \\ dec'(t) & = & \{s \mid s \in dec(t) \cap T_{D_R}\} \end{array}$$

［**定義** 3.2］ 消去する記号を $e \notin D_R$，$\tau(e) = (I, i)$ とするとき，

$$\mathcal{E}'(R) = \{ \begin{array}{l} cap_i(l) \to u \mid l \to r \in R, \\ \qquad u \in \{cap_i(r)\} \cup E'(r) \cup dec'(r)\}. \end{array}$$

関数記号 $e \in D_R$ のとき，従来のまま $\mathcal{E}'(R) = \mathcal{E}(R)$.

従来の一般消去法では，消去される関数記号下の全ての部分項は，残余部分，分解部分のどちらかに保存されていた．今回新たに定義した改良一般消去法においては，関数定義記号が出現しない項を除いたものが新たに残余部分，分解部分となる．残余部分，分解部分は生成される規則の右辺に対応するため，改良一般消去法で変換して得られる TRS の規則の数は従来の一般消去法よりも少なくなる．改良一般消去法についても以下の定理が成り立つことを示す．

［**定理** 3.3］ $\mathcal{E}'(R)$ が停止性を持つならば，R は停止性を持つ．

(証明) $e \in D_R$ の場合は，定理 2.11 より R は停止性を持つ．以下では，$e \notin D_R$ と仮定する．$\mathcal{E}'(R)$ が停止性を持つならば，補題 2.17 より $\mathcal{E}'(R) \cup DP(\mathcal{E}'(R))$ は停止性を持ち，また定理 2.4 より $\mathcal{E}'(R) \cup DP(\mathcal{E}'(R))$ と両立する整礎で単調な $F^{\#}$-代数が存在するので，それを $(A, >)$ とする．この $F^{\#}$-代数 $(A, >)$ から定理 2.16 を満たす弱書換え順序 $\gtrsim_A$ を定義する．改良一般消去法によって消去された関数記号を e とする．$F^{\#}$-代数 $(A, >)$ には，e の解釈 e_A がないのでここで新たに定義する．

$\tau(e) = (I, i)$ のとき，

$$e_A(x_1, ..., x_n) = \left\{ \begin{array}{ll} \diamond_A & i = 0 \\ x_i & i \neq 0. \end{array} \right.$$

仮定より $e \notin D_R$ なので，これで $R, DP(R)$ に出現する全ての関数記号に解釈が与えられた．消去された関数記号の解釈 e_A を導入した $F^{\#}$-代数 $(A, >)$ は，単調性を持たないが，弱単調性を持つことが分かる．

改良一般消去法の定義より，$l \to r \in R$ に対して，$cap_i(l) \to cap_i(r) \in \mathcal{E}'(R)$ が存在する．

一方，$u^{\#} \to v^{\#} \in DP(R)$ に対して，$u \to C[v] \in R$ となるから，$cap_i(u) \to C'[cap_i(v)] \in \mathcal{E}'(R)$ が存在する．よって，$cap_i(u^{\#}) \to cap_i(v^{\#}) \in DP(\mathcal{E}'(R))$ が存在する．

さらに，任意の項 t，割り当て $\sigma : V \to A$ に対して，$F^{\#}$-代数 $(A, >)$ 上で $[\sigma](t) = [\sigma](cap_i(t))$ となることは容易に確かめられる．よって，$l \to r \in R \cup DP(R)$ に対し任意の割り当て $\sigma : V \to A$ で $[\sigma](l) = [\sigma](cap_i(l)) > [\sigma](cap_i(r)) = [\sigma](r)$ が成り立つ．

以上より，$F^{\#}$-代数 $(A, >)$ は整礎で弱単調かつ $R \cup DP(R)$ と両立し，弱書換え順序 $\gtrsim_A$ は定理 2.16 を満たす．よって R は停止性を持つ．　□

$\mathcal{E}'(R) \subseteq \mathcal{E}(R)$ より $\mathcal{E}(R)$ が停止性を持つならば $\mathcal{E}'(R)$ も当然停止性を持つ．よってこの証明は定理 2.11 の証明にもなっている．

従来の一般消去法では停止性を持つ TRS に変換できなかった TRS R_3 において，

$$R_3 = \left\{ \begin{array}{rcl} f(a) & \to & f(b) \\ b & \to & g(a) \end{array} \right.$$

関数記号 g を 状態 $\tau(g) = (\emptyset, 0)$ で消去すると以下のようになる．

$$\mathcal{E}'(R_3) = \left\{ \begin{array}{rcl} f(a) & \to & f(b) \\ b & \to & \diamond \end{array} \right.$$

TRS $\mathcal{E}'(R_3)$ は明らかに停止性を持つ．よって，定理 3.3 より TRS R_3 は停止性を持つことが分かる．従って，改良一般消去法は従来の真の拡張になっている．

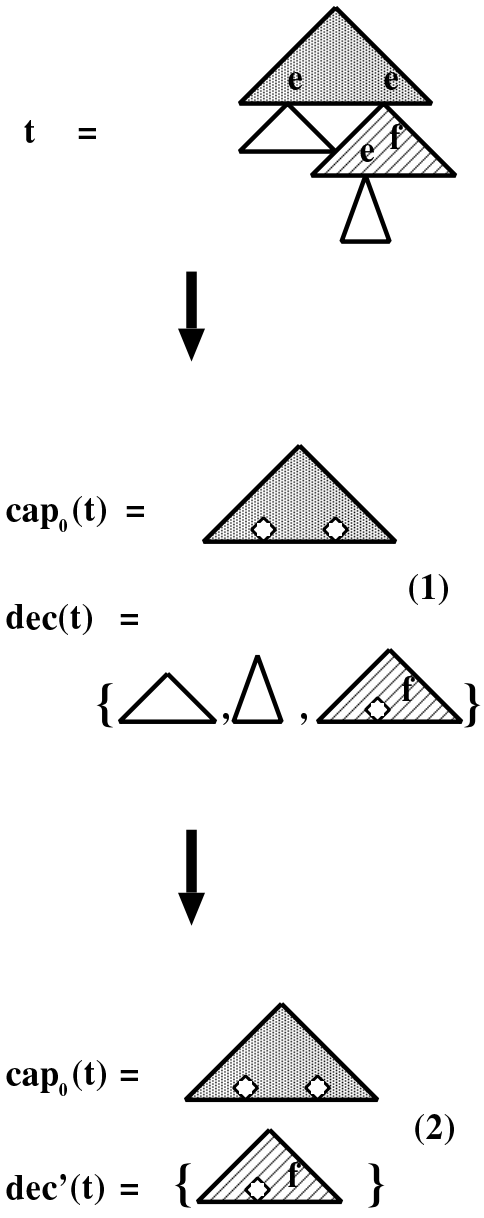

図 3　(1) 一般消去法，(2) 改良一般消去法
Fig. 3　(1)current,(2)improvement

4. ま と め

本研究で，一般消去法を拡張した改良一般消去法を提案し，その正当性を示した．改良一般消去法は，関数定義記号の出現に注目することで不必要な規則を減らし，従来の一般消去法の拡張になっている．また，本稿で正当性を示すために用いた依存対に基づく証明は，従来の意味的な証明 [5] に比べて簡潔になっている．しかし，関数定義記号を消去する際には，依存対による証明が適用できず，従来の一般消去法をそのまま用いている．今後の課題としては，改良一般消去法を関数定義記号を消去する場合でも適用できるように拡張すること，また意味的な変換手法である意味ラベリング [7] などを依存対によって解析することで，さらに決定可能な停止性の範囲を広げることである．

謝辞　本研究の一部は，文部省科学研究費 10139214, 10680346 の援助を受けている．

文　献

[1] T.Arts, "Automatically proving termination and innermost normalisation of term rewriting systems,"

PhD thesis, Universiteit Utrecht ,1997.

[2] T.Arts,J.Giesl, "Termination of term rewriting using dependency pairs," To appear in *Theoretical Computer Science*.

[3] F.Baader,T.Nipkow, "Term rewriting and all that," Cambridge University Press, 1998.

[4] M.C.F.Ferreira, "Termination of term rewriting -Well-foundedness, totality and transformations," PhD thesis, Universiteit Utrecht, 1995.

[5] M.C.F.Ferreira,H.Zantema, "Dummy elimination: Making termination easier," *LNCS* 965 ,pp.243-252, 1995.

[6] H.Zantema, "Termination of term rewriting: Interpretation and type elimination," *Journal of Symbolic Computation* 17 pp.23-50, 1994.

[7] H.Zantema: "Termination of term rewriting by semantic labelling". *Fundamenta Informatica* 24 pp.89-105, 1995.