修 士 論 文

消去法による項書換え系の停止性

指導教官　外山芳人 教授

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

中村正樹

1999 年 2 月 15 日

要 旨

項書換え系（TRS）において停止性は重要な性質のひとつである．これまでに様々な停止性判定のアルゴリズムが提案されてきている．しかし，一般に停止性判定は決定不能な問題であるため，それらのアルゴリズムが適用できない TRS が存在する．TRS の変換とは，停止性判定アルゴリズムを直接適用できない TRS を適用範囲内の TRS に変換することで停止性を示す手法である．本研究では，変換手法のひとつである消去法を解析し，従来よりも改良された消去法を提案する．

目 次

第 1 章

序論

1.1 背景と目的

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分野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証明を効率的に実現するための基礎理論である．

TRS は，無限の書換え列を持たないときに停止性があるという．計算モデルとして TRS を見るとき，停止性は解の存在を保証する重要な性質のひとつである．定理自動証明に有用な完備化手続きにおいても停止性は重要な役割を果たす．しかし，残念ながら TRS が停止性を持つかどうかという問題は決定不能であることが知られている．そのため，停止性を保証する十分条件に関する研究が活発に行なわれている [BN98]．

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．意味的な手法では，項を整礎な順序を持つ集合上に解釈する．書換えを集合上の順序に対応させることで，整礎性によって停止性を保証する [FZ96, MZ96, Zan94]．全ての停止性を持つ TRS に対して，停止性を保証する整礎な順序集合が存在することが分かっている．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な整礎な順序を定義することで停止性を示す [Der87, Ste95]．構文的手法では再帰経路順序による順序付けが代表的である．再帰経路順序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．しかし，停止性判定は決定不能な問題であるため，再帰経路順序で停止性示すことのできる TRS は限られている．実際，再帰経路順序は，項よりもその部分項が真に小さくなる単純化順序になっているため適用範囲が限られている．

TRS の変換は，停止性判定の有効な手法のひとつである [Fer95, FZ95, MOZ96, Zan94, Zan95]．変換は，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は単純化順序で規則に順序付けできないので再帰経路順序によって停止性を示すことができない．

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

実際に左辺が右辺よりも真に大きくなるような単純化順序があると仮定すると，単純化順序の性質より，$f(f(x)) > f(f(x))$ となってしまい矛盾する．変換手法のひとつである置換消去法によって，TRS R は，関数記号 g の出現を新たな定数記号 $\diamond$ に置き換えることで TRS $\mathcal{E}(R)$ に変換される．

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

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．TRS $\mathcal{E}(R)$ は，書換えによって関数記号 f の数が減っていくので停止性を持つことは明らかである．実際，再帰経路順序によって TRS $\mathcal{E}(R)$ の停止性を示すことができる．置換消去法による変換は，規則の数が有限ならば自動で行なうことができる．したがって，再帰経路順序のような停止性判定手法と組み合わせることによって決定可能な停止性のクラスを広げることができる．

一方，近年提案された依存対は，TRS の無限書換え列の解析に有効な概念である [Art97, AG97a, AG97b, AG99]．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，構文的な変換手法である消去法の改良を試みる [NKT99, NT98a, NT98b]．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．依存対を用いた正当性の証明では，弱書換え順序が本質的な役割を果たす．弱書換え順序の構成法として，切り落とし法を定義し，変換の正当性の証明手法を一般化する [KT98]．

1.2 構成

本稿は以下のように構成されている．次章では，準備として TRS の基礎的な説明を行なう．３章では，従来知られている代表的な TRS の停止性判定法について構文的手法，意

味的手法のそれぞれを紹介し，さらに停止性の間の階層関係について述べる．４章では，TRS の変換手法の１つ，置換消去法の定義を述べる．置換消去法の正当性の示すために依存対の概念を紹介する．依存対による正当性の証明を通して，置換消去法が改良可能であることを示す．５章では，弱書換え順序を構文的に構成する手法として，切り落とし法を紹介する．切り落とし法を用いることで変換手法の正当性の証明手法を一般化する．６章で，その他の変換手法として分配消去法を紹介する．分配則を含まない場合の分配消去法について改良を施し，正当性を切り落とし法を用いて示す．さらに，置換消去法と分配消去法の部分項の扱い方を融合させた一般消去法について述べ，同様の改良を行なう．最後に７章で，本研究の成果をまとめ，今後の課題を述べる．

第 2 章

準備

項書換え系（TRS）は，等式推論を効率良く実現するための基礎理論である．TRS は，項から項への規則の集合 R で表される．規則は，等式論理における公理に相当する．規則の集合 R によって定められる書換え関係によって計算が実現される．この章では，TRS と書換え関係を定義し，例を用いてどのように TRS が計算を行なうかを示す．この章における表記は，文献 [BN98, Fer95] を参考にした．

2.1 項

関数記号 $f, g, \ldots$ の集合を F，変数 $x, y, z, \ldots$ の可算無限集合を V とする．但し，$F \cap V = \emptyset$．また，各関数記号には引数が定められているとし，$arity(f) \in \mathbf{N}$ で関数記号 f の引数を表す．引数 0 の関数記号は**定数**と呼ぶ．本論文では，特に関数記号の集合 Fが有限集合の場合を扱う．

項 $s, t, u, \ldots$ の集合 $T(F, V)$ は，以下を満たす最小の集合である．

- $x \in T(F,V)$ if $x \in V$
- $f(t_1, \ldots, t_n) \in T(F,V)$ if $t_1, \ldots, t_n \in T(F,V), f \in F, arity(f) = n$

特に，すべての変数が高々 1 回しか出現しない項を**線形項**と呼ぶ．

特別な項として，**文脈**を定義する．文脈とは，F に属さない特別な定数 □ を只１つだけ含む項である．文脈 $C[\ ]$ の □ の出現を項 t で置き換えて得られる項を $C[t]$ で表す．任意の文脈 $C[\ ]$ に対して，項 t を項 $C[t]$ の**部分項**であるという．また，定数 □ だけからなる文脈を**自明**であるという．自明でない文脈 $C[\ ]$ に対しては，項 t を項 $C[t]$ の**真部分項**

であるという．変数V から項$T(F,V)$ への写像θ を**代入**と呼ぶ．但し，有限個を除く変数 $x \in V$ について $\theta(x) = x$ である．代入θ は，次のように項から項への写像に自然に拡張される．

$$\theta(f(t_1, \ldots, t_n)) = f(\theta(t_1), \ldots, \theta(t_n))$$

代入θ によって得られる項$\theta(t)$ を $t\theta$ で表す．

例 2.1.1 項t，文脈$C[\ ]$，代入θをそれぞれ以下のように定義する．

$$\begin{aligned} t &= f(g(x,y,z),x), \\ C[\ ] &= h(z,\Box), \\ \theta(x) &= f(a,x) \\ \theta(z) &= x. \end{aligned}$$

このとき，項$C[t]$, $t\theta$ は以下のようになる（図 2.1）．

$$\begin{aligned} C[t] &= h(z, f(g(x,y,z),x)) \\ t\theta &= f(g(f(a,x),y,x), f(a,x)). \end{aligned}$$

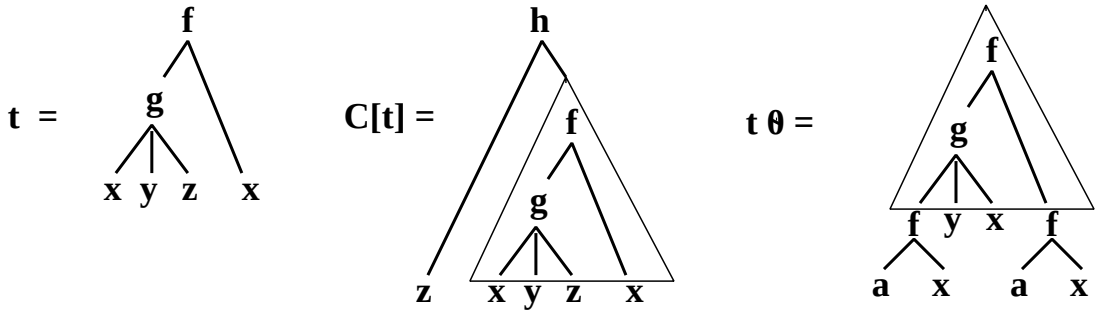


図 2.1: 項，文脈，代入

2.2 項書換え系

項t に出現する変数の集合を $Var(t)$ とする．項の対 $(l,r) \in T(F,V) \times T(F,V)$ が以下の条件を満たすとき，**書換え規則**という．

$$l \notin V$$
$$Var(r) \subset Var(l)$$

以下では，書換え規則 (l, r) を単に **規則**と呼び，$l \to r$ で記述する．

例 2.2.1 以下は，項の対であるが規則ではない．

$$\begin{array}{rcll} x & \to & f(x) & \text{左辺が変数,} \\ f(x) & \to & g(x,y) & \text{変数 } y \text{ が左辺に出現していない.} \end{array}$$

項書換え系（TRS）は，関数記号の集合 F と変数の集合 V ，規則の集合 R によって，(F, V, R) で表される．特に混乱のない限り TRS は規則の集合 R で略記する．すべての規則の右辺（左辺）が線形項であるような TRS を**右線形（左線形）**であるという．

TRS R による**書換え関係** $\underset{R}{\to}$ を以下で定義する．但し，θ は代入，$C[\ ]$ は文脈とする（図 2.2）．

定義 2.2.2 [BN98]

$$s \underset{R}{\to} t \overset{\text{def}}{\iff} \exists\theta, \exists C[\ ], \exists l \to r \in R, \quad s = C[l\theta], t = C[r\theta]).$$

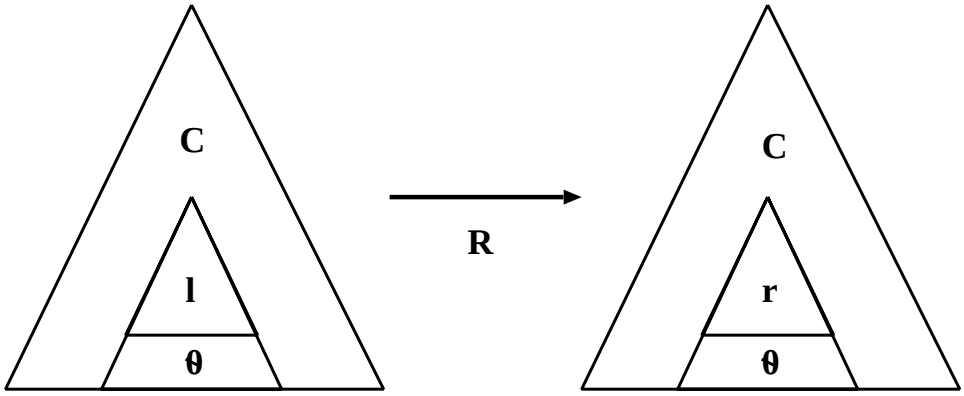


図 2.2: TRS R による書換え関係

書換え $s \underset{R}{\to} t$ に対して，項 s の部分項 $l\theta$ を**リデックス**という．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．項 $t_1, t_2, t_3, \ldots$ が各 $i \geq 1$ に対して，$t_i \underset{R}{\to} t_{i+1}$ の関係にあるとき，項の列 $t_1 \underset{R}{\to} t_2 \underset{R}{\to} t_3 \underset{R}{\to} \ldots$ を**書換え列**という．特に，無限個の項からなる書換え列を**無限書換え列**という．

例 2.2.3 論理式の真偽を判定するための TRS R_1 を考える．

$$R_1 = \left\{ \begin{array}{rcl} and(x,x) & \to & x \\ and(x,F) & \to & F \\ and(F,x) & \to & F \\ not(T) & \to & F \\ not(F) & \to & T \end{array} \right.$$

TRS R_1 の規則は，それぞれ論理式の基本的な等式

$$x \wedge x = x, \quad x \wedge F = F, \quad F \wedge x = F, \quad \neg T = F, \quad \neg F = T$$

に対応している．

TRS R_1 により，以下のような書換え列が存在する．

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

これは論理式 $\neg(\neg F \wedge F)$ が真であることを意味している．

第 3 章

停止性

TRS R が**停止性**を持つとは，無限書換え列 $t_1 \underset{R}{\rightarrow} t_2 \underset{R}{\rightarrow} \cdots$ が存在しないことである．TRS を計算モデルとして見たとき，停止性を持った TRS は，計算の道筋によらず解の存在が保証された計算モデルである．TRS が停止性を持つかどうかの判定は決定不能な問題であることが知られている [BN98]．したがって，TRS が停止性を持つための十分条件に関する研究がさかんに行なわれている．この章では，従来知られている代表的な停止性判定法について説明する．

3.1 書換え順序

TRS の停止性は一般に項の集合 $T(F, V)$ 上の順序を用いて示される．以下で順序は，狭義の半順序，すなわち推移的で非反射的な二項関係をさす．また，二項関係 $\geq = (> \cup =)$ とする．順序 $>$ が**整礎**であるとは無限下降列 $a_1 > a_2 > a_3 > \cdots$ が存在しないことである．

整礎な項の集合上の順序 $>$ が存在して，TRS R によって $s \underset{R}{\rightarrow} t$ と書換えられたとき $s > t$ となるならば，整礎性より TRS R は停止性を持つ．一般に TRS R によって，$s \underset{R}{\rightarrow} t$ と書換えられるような項 s, t は無限に存在するため，これだけでは停止性判定は難しい．以下では，TRS R の規則だけを調べることで停止性を保証するような順序を定義する．

定義 3.1.1 [BN98] 以下を満たす項の集合上の順序 $>$ を**書換え順序**という．

1. 整礎である．

2. 任意の文脈 $C[\,]$ に対して，$s > t$ ならば $C[s] > C[t]$ ．

3. 任意の代入 θ に対して，$s > t$ ならば $s\theta > t\theta$ ．

順序 $>$ は，2. の条件を満たしているとき**文脈に閉じている**といい，3. のとき**代入に閉じている**という．また，項の集合上の擬順序 $\gtrsim$ が，文脈，代入に閉じていて，順序 $>= (\gtrsim \setminus \lesssim)$ [1] が，整礎で代入に閉じているとき，擬順序 $\gtrsim$ を**弱書換え順序**という [Art97]．但し，擬順序とは，反射的で推移的な二項関係である．

書換え順序は，与えられた TRS の規則を調べるだけで停止性を保証することができる．

定義 3.1.2 [BN98] 二項関係 $\rhd$ が TRS R の任意の規則 $l \to r \in R$ に対して $l \rhd r$ となるとき，二項関係 $\rhd$ は TRS R と**両立**するという．

定理 3.1.3 [BN98] TRS R と両立する書換え順序 $>$ が存在することと TRS R が停止性を持つことは同値である．

証明 TRS R と両立する書換え順序を $>$ とする．定義 2.2.2 より，$s \xrightarrow[R]{} t$ ならば $s = C[l\theta]$, $t = C[r\theta]$ と書ける．両立性より $l > r$，定義 3.1.1 の 3. より $l\theta > r\theta$，2. より $C[l\theta] > C[r\theta]$．すなわち $s > t$ である．1. より TRS R は停止性を持つ．逆に TRS R が停止性を持つとすると，二項関係 $\xrightarrow[R]{+}$ は書換え順序であり，明らかに TRS R と両立する． □

一方，弱書換え順序 $\gtrsim$ は，狭義部分 $>= (\gtrsim \setminus \lesssim)$ が TRS R と両立しても停止性を保証しない．狭義部分 $>$ は，整礎であるが単調性を持っていないため，狭義部分 $>$ が両立しても，書換え $s \xrightarrow[R]{} t$ に対して $s > t$ は保証されないからである．

例 3.1.4 規則 $a \to f(a)$ のみからなる TRS を考える．擬順序 $\gtrsim$ を任意の項 s, t に対して，

$$\begin{aligned} f(s) &\sim f(t), \\ a &> f(t). \end{aligned}$$

となるように定義する．擬順序 $\gtrsim$ は弱書換え順序であり，狭義部分 $>$ は，規則 $a \to f(a)$ と両立する．しかし，明らかに無限書換え列

$$a \xrightarrow[R]{} f(a) \xrightarrow[R]{} f(f(a)) \xrightarrow[R]{} f(f(f(a))) \xrightarrow[R]{} \cdots$$

[1] 擬順序 $\gtrsim$ に対して，順序 $>= (\gtrsim \setminus \lesssim)$ を**狭義部分**，同値関係 $\sim= (\gtrsim \cap \lesssim)$ を**同値部分**と呼ぶ．

が存在する．弱書換え順序で見ると，以下のように同値部分∼で無限列が存在している．

$$a > f(a) \sim f(f(a)) \sim f(f(f(a))) \sim \cdots.$$

書換え順序を定義する方法には大きく分けて構文的手法と意味的手法の２通りがある．前者の構文的手法では，関数記号の集合上に定義された優先順位から書換え順序を定義する．次節でその代表である再帰経路順序について詳しい定義を述べる．再帰経路順序による二項間の順序付けは決定可能であるため実装に適している．しかし，再帰経路順序によって定義される書換え順序は，単純化順序という制限された順序になっているため停止性を示すことのできる TRS は限られている．

定義 3.1.5 [MZ96] 項の集合上の順序 $>$ が**単純化順序**である$\stackrel{\text{def}}{\Longleftrightarrow}$ 任意の項 t と任意の自明でない文脈 $C[\ ]$ に対して，$C[t] > t$ である．

後者の意味的手法では，項をある整礎な順序を持つ集合の要素に解釈することで書換え順序を定義する．項の解釈は，F 上の各関数記号ごとに定義された写像によって行なわれるため F-代数と呼ばれる．すべての停止性を持つ TRS に対して，停止性を保証するような F-代数が存在することが知られている．しかし，TRS の停止性を保証する F-代数は構成的ではないため，意味的手法による停止性判定は実装に向かない．F-代数については次々節で詳しく述べることにする．

3.2 再帰経路順序

再帰経路順序は，関数記号の集合上に定義された優先順位を拡張することで得られる項の集合上の順序である．再帰経路順序は書換え順序になっているため，与えられた TRS の規則を調べるだけで停止性を示すことができる．再帰経路順序による二項間の順序付けは決定的であるので，規則の数が有限ならば再帰経路順序による停止性判定法は決定的である．

再帰経路順序の定義の前に，まず多重集合順序を定義する．**多重集合**とは，要素の有限個の重複を許した集合である．例えば，自然数を要素とする多重集合 $\{1,1,2,3\}$ は，多重集合 $\{1,2,3\}$ とは区別される．多重集合 X, Y に対して，$X - Y$ は，X 上の各要素の出現回数から Y 上のその要素の出現回数を引いたもので定義する．但し，X に出現しない Y の

要素は$X-Y$でも出現しないものとする．例えば，$\{a,a,b,b,b,c\}-\{a,b,b,d\}=\{a,b,c\}$となる．

定義 3.2.1 [BN98] 集合A上の順序を$>$とする．**多重集合順序** $>^{mul}$ は，集合Aの多重集合上の順序として以下のように定義される．

$$X >^{mul} Y \stackrel{\text{def}}{\Longleftrightarrow} X \neq Y \text{ and } \forall y \in Y - X, \exists x \in X - Y, x > y.$$

但し，X, YはA上の多重集合．

定理 3.2.2 [BN98] 順序 $>$が整礎であることと多重集合順序$>^{mul}$が整礎であることは同値である．

優先順位 $\trianglerighteq$ は，関数記号の集合F上の擬順序である．優先順位 $\trianglerighteq$ に対して，同値部分を $\sim$，狭義部分を $\triangleright$ とする．

再帰経路順序 $>_{rpo}$ は以下のように定義される．

定義 3.2.3 [Fer95] 関数記号の集合F上の優先順位を$\trianglerighteq$とし 項をs,t とする．

$$\begin{aligned} s >_{rpo} t \stackrel{\text{def}}{\Longleftrightarrow}\quad & s \equiv f(s_1,\ldots,s_m) \text{ and} \\ & 1.\quad t \equiv g(t_1,\ldots,t_n) \text{ and } \forall i, s >_{rpo} t_i \text{ and} \\ & \qquad f \triangleright g \text{ or} \\ & \qquad f \sim g \text{ and } \{s_1,\ldots,s_m\} >_{rpo}^{mul} \{t_1,\ldots,t_n\} \text{ or} \\ & 2.\quad \exists i, s_i >_{rpo} t \text{ or } s_i = t. \end{aligned}$$

優先順位が整礎ならば，再帰経路順序 $>_{rpo}$ は書換え順序になっており，さらに単純化順序にもなっている．証明は，文献 [Der87] を参照されたい．定義より明らかに，項s,tが与えられたとき $s >_{rpo} t$ であるかどうかは決定可能な問題である．以下のような停止性判定の手続きが考えられる．

1. TRS Rに現れる関数記号上に整礎な優先順位を決める．
2. すべての規則$l \to r$について$l >_{rpo} r$ならば，TRS Rは停止性を持つ．
3. そうでないならば，異なる優先順位で繰り返す．

与えられたTRS R の規則の数が有限ならば，2. のステップは決定的である．また関数記号の数も有限なので優先順位の数も有限となり，この手続きは決定的となる．

例 3.2.4 例 2.2.3 の TRS R_1を考える．

$$R_1 = \left\{ \begin{array}{rcl} and(x,x) & \rightarrow & x \\ and(x,F) & \rightarrow & F \\ and(F,x) & \rightarrow & F \\ not(T) & \rightarrow & F \\ not(F) & \rightarrow & T \end{array} \right.$$

優先順位を $and \rhd F, not \rhd T, not \rhd F$ を満たすように定義すると，TRS R_1 は，再帰経路順序で停止性を示すことができる．

各規則について，左辺の最外の関数記号が右辺に出現するどの関数記号よりも真に大きくなるように優先順位を定義できるならば，その TRS は再帰経路順序によって停止性を示すことができる．

再帰経路順序による停止性判定は実装に適しているが，単純化順序に制限されるため適用できる TRS は限られている．

例 3.2.5 以下のような TRS を考える．

$$R_2 = \{f(f(x)) \rightarrow f(g(f(x)))\}$$

TRS R_2は，再帰経路順序で停止性が示せない．何故なら，もし $f(f(x)) >_{rpo} f(g(f(x)))$ となるように再帰経路順序が定義できたとすると，単純化順序の性質より，$g(f(x)) >_{rpo} f(x)$．また書換え順序の性質（文脈に閉じている）から $f(g(f(x))) >_{rpo} f(f(x))$ となり順序の定義に矛盾する．

再帰経路順序は，多重集合順序による定義の他に辞書式順序をもとに定義されることもある．また，関数記号に状態を定義することで，多重集合順序か辞書式順序かを選択できるように汎用性を持たせた定義もある [Fer95]．再帰経路順序以外にも，決定性のある順序がいくつか提案されているが，ほとんどが単純化順序に準じている [Ste95]．単純化順序によって停止性を示すことのできる TRS を**単純停止性**を持つという．

3.3 F-代数

F-代数 $(A, >)$ は，集合 A とその上の順序 $>$ の対であり，各関数記号 $f \in F, arity(f) = n$ について写像 $f_A : \overbrace{A \times \cdots \times A}^{n} \to A$ が定義されている．写像 f_A を関数記号 f の**解釈**と呼ぶ．

集合 A 上の任意の**割り当て** $\sigma : V \to A$ と各関数記号の解釈により，項 t は以下のように集合 A の要素 $[\![t, \sigma]\!]_A$ に解釈される．

$$\begin{array}{lcll} [\![x, \sigma]\!]_A & = & \sigma(x) & \text{if } x \in V \\ [\![f(t_1, \ldots, t_n), \sigma]\!]_A & = & f_A([\![t_1, \sigma]\!]_A, \ldots, [\![t_n, \sigma]\!]_A) & \text{if } t_1, \ldots, t_n \in T(F, V), f \in F \end{array}$$

F-代数 $(A, >)$ を用いて，順序 $>_A$ を以下のように定義する．

定義 3.3.1 [Zan94] F-代数 $(A, >)$ とするとき，

$$s >_A t \overset{\text{def}}{\iff} \forall \sigma : V \to A, [\![s, \sigma]\!]_A > [\![t, \sigma]\!]_A.$$

次に，順序 $>_A$ が書換え順序になるように F-代数の性質をいくつか定義する．

定義 3.3.2 [Zan94] F-代数 $(A, >)$ について以下を定義する．

1. 整礎である．$\overset{\text{def}}{\iff}$ 順序 $>$ が整礎である．
2. TRS R と両立する．$\overset{\text{def}}{\iff}$ 順序 $>_A$ が TRS R と両立する．
3. **単調**である．$\overset{\text{def}}{\iff} a > b$ ならば $\forall f \in F, f_A(\ldots, a, \ldots) > f_A(\ldots, b, \ldots)$.
4. **弱単調**である．$\overset{\text{def}}{\iff} a \geq b$ ならば $\forall f \in F, f_A(\ldots, a, \ldots) \geq f_A(\ldots, b, \ldots)$.

補題 3.3.3 [Zan94] F-代数 $(A, >)$，代入 $\theta : T(F, V) \to T(F, V)$，集合 A 上の割り当て $\sigma : V \to A$ とする，割り当て $\rho : V \to A$ を $\rho(x) = [\![\theta(x), \sigma]\!]_A$ と定義すると以下が成り立つ．

$$[\![t\theta, \sigma]\!]_A = [\![t, \rho]\!]_A$$

証明

項の構造に関する帰納法による．変数 x に対しては，

$$\begin{aligned}[\![x\theta, \sigma]\!]_A &= [\![\theta(x), \sigma]\!]_A \\ &= \rho(x) \\ &= [\![x, \rho]\!]_A\end{aligned}$$

次に関数記号 $f \in F, arity(f) = n$ とする．帰納法の仮定より，項 $t_1, \ldots, t_n$に対しては補題が成り立つ．

$$\begin{aligned}[\![f(t_1, \ldots, t_n)\theta, \sigma]\!]_A &= [\![f(t_1\theta, \ldots, t_n\theta), \sigma]\!]_A \\ &= f_A([\![t_1\theta, \sigma]\!]_A, \ldots, [\![t_n\theta, \sigma]\!]_A) \\ &= f_A([\![t_1, \rho]\!]_A, \ldots, [\![t_n, \rho]\!]_A) \\ &= [\![f(t_1, \ldots, t_n), \rho]\!]_A\end{aligned}$$

□

定理 3.3.4 [Zan94] F-代数が整礎で単調ならば，順序$>_A$は書換え順序である．

証明 F-代数を $(A, >)$ とする．集合 A 上の順序>が整礎であることから順序$>_A$が整礎であることは明らか．$s >_A t$ と仮定する．任意の文脈 $C[\ ]$ に対して $C[s] > C[t]$ であることを，文脈 $C[\ ]$ の構造に関する帰納法で示す．文脈 $C[\ ]$ が自明のとき，すなわち $C[\ ] = \Box$, は明らか．文脈 $C[\ ]$ で $C[s] > C[t]$ が成り立つと仮定し，任意の関数記号 $f \in F$とする．このとき任意の割り当てσに対して，

$$\begin{aligned}[\![f(\ldots, C[s], \ldots), \sigma]\!]_A &= f_A(\ldots, [\![C[s], \sigma]\!]_A, \ldots) \\ &> f_A(\ldots, [\![C[t], \sigma]\!]_A, \ldots) \quad \text{（単調性より）} \\ &= [\![f(\ldots, C[t], \ldots), \sigma]\!]_A\end{aligned}$$

任意の代入θとする．このとき任意の割り当てσに対して，補題3.3.3の割り当てρが存在し，

$$[\![s\theta, \sigma]\!]_A = [\![s, \rho]\!]_A > [\![t, \rho]\!]_A = [\![t\theta, \sigma]\!]_A.$$
$$(s >_A t \text{ より})$$

□

F-代数 $(A, >)$ に対して，擬順序 $\gtrsim_A$を以下で定義する．

$$s \gtrsim_A t \overset{\text{def}}{\iff} \forall \sigma : V \to A, [\![s, \sigma]\!]_A \geq [\![t, \sigma]\!]_A.$$

擬順序 $\gtrsim_A$の狭義部分は $>_A$である（$>_A = (\ \gtrsim_A \setminus \lesssim_A)$）．次の定理が成り立つ．

定理 3.3.5 [NKT99] F-代数が整礎で弱単調ならば，順序 $\gtrsim_A$は弱書換え順序である．

証明 定理 3.3.4 同様． □

定理 3.3.6 [Zan94] TRS R と両立する整礎で単調な F-代数が存在することと TRS R が停止性を持つことは同値である．

証明 F-代数が存在すると仮定する．定理 3.3.4より，順序$>_A$は書換え順序である．両立性より，TRS Rは停止性を持つ（定理 3.1.3）．逆に，TRS R が停止性を持つならば，定理 3.1.3より TRS Rと両立する書換え順序$>$が存在する．定義より，書換え順序は整礎で単調である．したがって TRS R と両立し整礎で単調な F-代数 $(T(F,V),>)$ が存在する．
□

例 3.3.7 再び，例 2.2.3 の TRS R_1を考える．

$$R_1 = \left\{ \begin{array}{rcl} and(x,x) & \to & x \\ and(x,F) & \to & F \\ and(F,x) & \to & F \\ not(T) & \to & F \\ not(F) & \to & T \end{array} \right.$$

F-代数 $(N,>)$ として自然数とその上の通常の順序を考える．各関数記号に次のような解釈を与える．

$$\begin{array}{rcl} and_A(x,y) & = & x+y+1 \\ not_A(x) & = & x+1 \\ T_A & = & 0 \\ F_A & = & 0 \end{array}$$

この F-代数 $(N,>)$ は，整礎で単調かつ TRS R_1と両立するので，定理 3.3.6より，TRS R_1の停止性を保証する．

停止性を持つすべての TRS に対して，その停止性を保証する F-代数が存在するため，F-代数による停止性判定は非常に強力である．しかし，定理 3.3.6は，F-代数の存在を保証するが，構成法を与えるわけではない．したがって，F-代数による停止性判定法は，人間の手による部分が多く実装には向かない．

3.4 停止性の階層

単純化順序によって停止性を示すことのできる TRS を単純停止性を持つと言った．この節では，単純停止性に F-代数を用いた意味的な位置付けを行なう．さらに，F-代数の性質から全停止性を定義し，停止性，単純停止性，全停止性の間の包含関係を調べる．

F-代数について，以下の性質を定義する．

定義 3.4.1 [Zan94] 単調な F-代数 $(A, >)$ とする．任意の $f \in F$について，

$$f_A(a_1, \ldots, a_n) \geq a_i$$

となるとき，F-代数 $(A, >)$ は単純であるという．但し，$a_1, \ldots, a_n \in A, 1 \leq i \leq n$.

TRS $Emb(F)$ を以下で定義する．以下では，TRS R は，TRS (F, V, R) とする．

定義 3.4.2 [Zan94]

$$Emb(F) = \{f(x_1, \ldots, x_n) \to x_i \mid f \in F, arity(f) = n \geq 1, 1 \leq i \leq n\}$$

単純停止性について同値な条件を挙げる．

定理 3.4.3 [Zan94] 以下は同値である．

(1) TRS Rが単純停止性を持つ．

(2) TRS Rと両立する単調で単純で整礎な F-代数 $(A, >)$ が存在する．

(3) TRS $R \cup Emb(F)$ が停止性を持つ．

証明 (1)⇒(2).

単純停止性を持つことから，TRS Rと両立する書換え順序>が存在して単純化順序である．単純化順序>は，項の集合 $T(F, V)$ 上の順序として明らかに単純である．よって TRS Rと両立する単調で単純で整礎な F-代数 $(T(F, V), >)$ が存在する．

(2)⇒(3).

(2) の F-代数 $(A, >)$ から以下の F-代数 $(B, >)$ を定義する．

$$\begin{aligned} B &= A \times N \\ (a, k) > (a', k') &\overset{\text{def}}{\Longleftrightarrow} a > a' \text{ or } (a = a', k > k') \\ f_B((a_1, k_1), \ldots, (a_n, k_n)) &= (f_A(a_1, \ldots, a_n), 1 + \textstyle\sum_n k_i). \end{aligned}$$

明らかに，F-代数 $(B, >)$ は，単調で整礎かつ TRS Rと両立する．また TRS $Emb(F)$ とも両立する．したがって，定理 3.3.6より TRS $R \cup Emb(F)$ は停止性を持つ．

(3)⇒(1)．

TRS $R \cup Emb(F)$ と両立する書換え順序を>とする．書換え順序>は，TRS $Emb(F)$ と両立することから，任意の関数記号 $f \in F$に対して，

$$f(\ldots, t, \ldots) > t$$

である．したがって任意の自明でない文脈 $C[\ ]$，項 t に対して，

$$C[t] > t$$

は自明． □

順序が全域的である F-代数を用いて単純停止性よりも更に条件を強めた全停止性が定義できる．全域的な順序とは，任意の元 $a, b \in A$ に対して，$a > b, a = b, b > a$ のいずれかが成り立つ集合 A 上の順序>である．

定義 3.4.4 [Zan94] TRS R が**全停止性**を持つ．$\stackrel{\text{def}}{\Longleftrightarrow}$ TRS R と両立する整礎で単調な F-代数 $(A, >)$ が存在し，集合 A 上の順序>は全域的である．

全停止性と単純停止性について次の包含関係がある．

定理 3.4.5 [Zan94] TRS R が全停止性を持っているならば，TRS R は単純停止性を持っている．

証明 定義より，TRS R と両立する整礎で単調な F-代数 $(A, >)$ が存在し，集合 A 上の順序>は全域的である．順序>が単純でないと仮定すると，ある関数記号 $f \in F$，要素 $a_i \in A$ に対して，$f_A(\ldots, a_i, \ldots) \geq a_i$でない．すると順序>は全域的であることから，$a_i > f_A(\ldots, a_i, \ldots)$ が成り立つ．単調性より以下の無限下降列が存在し，整礎性に矛盾．

$$\begin{aligned} a_i &> f_A(\ldots, a_i, \ldots) \\ &> f_A(\ldots, f_A(\ldots, a_i, \ldots), \ldots) \\ &> f_A(\ldots, f_A(\ldots, f_A(\ldots, a_i, \ldots), \ldots), \ldots) \\ &> \cdots \end{aligned}$$

□

停止性に関して以下の包含関係が成り立つ．関係 $A \Rightarrow B$は，TRS が A を持つならば，B を持つことを意味する．

$$\text{全停止性} \Rightarrow \text{単純停止性} \Rightarrow \text{停止性}$$

最後に再帰経路順序と全停止性の関係を示す．証明は，文献 [Fer95] を参照されたい．

定理 3.4.6 [Fer95] 優先順位が全域的なとき，再帰経路順序によって停止性を示すことのできる TRS は全停止性を持っている．

第 4 章

置換消去法

すでに見たように，停止性を持つにも関わらず既存の停止性判定アルゴリズム（再帰経路順序など）では適用範囲外である TRS が存在する．TRS の変換は，停止性判定が困難な TRS の停止性を判定するために提案された手法である．変換は，複雑な構造の TRS をより単純な構造の TRS に変換し，得られた TRS の停止性などの性質を調べることで，元の TRS の停止性を保証する．本研究では，決定性のある停止性判定アルゴリズムに興味がある．したがって，変換手法のうち変換自体が決定的な手続きである消去法を扱う．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された 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や分配消去法，一般消去法が提案されている．この章では，置換消去法の定義と性質を述べる．また停止性判定に有効な概念である依存対を紹介し，置換消去法の正当性を依存対を用いることで示す．依存対による証明を通して置換消去法が改良可能であることを示す．

4.1 定義

置換消去法は，停止性判定に不必要と思われる関数記号を新たな定数関数記号 $\diamond$ に置き換えることで，停止性判定が容易な TRS に変換する．

置換消去法の定義のために，項 t の**キャップ部分** $cap(t)$ 及び**分解部分** $dec(t)$ を定義する．但し，関数記号の集合 F，消去する関数記号 $e \in F$，新たな定数関数記号 $\diamond \notin F, arity(\diamond) = 0$ とし，関数記号の集合 $F_\diamond = (F - \{e\}) \cup \{\diamond\}$ とする．

定義 4.1.1 [FZ95] $cap : T(F,V) \to T(F_\diamond, V)$

$$cap(t) = \begin{cases} t & \text{if } t \in V \\ f(cap(t_1), \ldots, cap(t_n)) & \text{if } t = f(t_1, \ldots, t_n), f \neq e \\ \diamond & \text{if } t = e(t_1, \ldots, t_n) \end{cases}$$

定義 4.1.2 [FZ95] $dec : T(F,V) \to \mathcal{P}(T(F_\diamond, V))$

$$dec(t) = \begin{cases} \emptyset & \text{if } t \in V \\ \bigcup_n dec(t_i) & \text{if } t = f(t_1, \ldots, t_n), f \neq e \\ \bigcup_n (dec(t_i) \cup \{cap(t_i)\}) & \text{if } t = e(t_1, \ldots, t_n) \end{cases}$$

キャップ部分は，消去する関数記号 e から始まる部分項を定数 $\diamond$ で置き換える関数である．また，分解部分は，消去する関数記号 e から始まるすべての部分項 $e(t_1, \ldots, t_n)$ に対して集合 $\{cap(t_1), \ldots, cap(t_n)\}$ の和集合を返す関数である (図 4.1)．

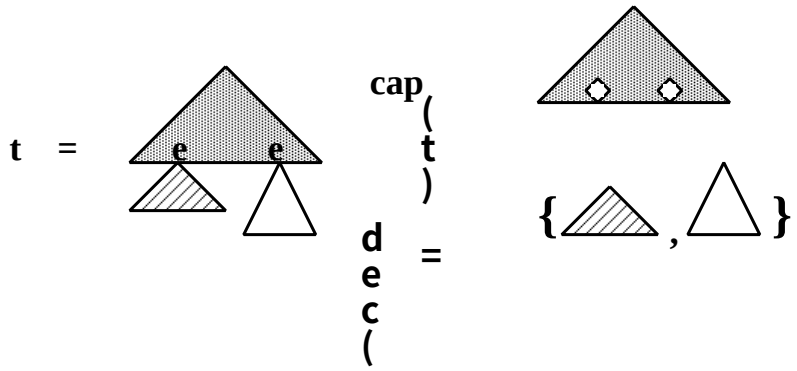


図 4.1: キャップ部分，分解部分

例 4.1.3 項 $t = f(e(a, g(e(x,y))))$ に対して，消去する関数記号を e とする．項 t のキャップ部分，分解部分は以下の通りである．

$$\begin{aligned} cap(t) &= f(\diamond) \\ dec(t) &= \{a, g(\diamond), x, y\} \end{aligned}$$

キャップ部分，分解部分により，TRS R の関数記号 e を置換消去して得られる TRS $\mathcal{DM}(R)$ は以下のように定義される[1]．

[1] 正確には，TRS (F,V,R) から TRS $(F_\diamond, V, \mathcal{DM}(R))$ への変換である．文献 [FZ95] では，消去する関数記号 e の出現は右辺のみという制約があったが，ここでは，左辺にも関数記号 e の出現を許す文献 [Fer95] における定義を用いる．

定義 4.1.4 [Fer95]

$$\mathcal{DM}(R) = \{cap(l) \to r' \mid l \to r \in R, r' \in dec(r) \cup \{cap(r)\}\}$$

置換消去法は，与えられた TRS Rのすべての規則について，左辺をそのキャップ部分，右辺をそのキャップ部分もしくは分解部分の要素となるように変換する．TRS R と置換消去法によって得られた TRS $\mathcal{DM}(R)$ との間に以下のような停止性に関する包含関係がある．証明は次々節で行なう．

定理 4.1.5 [Fer95] TRS $\mathcal{DM}(R)$ が停止性を持つならば，TRS Rは停止性を持つ．

一般に，TRS R から TRS R' を得るような変換を考えるとき，TRS R' が停止性を持つならば TRS R も停止性を持つという関係を，変換が**正当**であると呼ぶことにする．置換消去法の正当性は，次節で紹介する依存対の性質を用いて示すことができる．

例 4.1.6 例 3.2.5の TRS R_2を考える．

$$R_2 = \{f(f(x)) \to f(g(f(x)))\}$$

TRS R_2 は，単純化順序で停止性を示すことができなかった．しかし，置換消去法を用いることで，単純停止性を持つ TRS に変換することができる．消去する関数記号を gとする．

$$\mathcal{DM}(R_2) = \left\{ \begin{array}{lcl} f(f(x)) & \to & f(\diamond) \\ f(f(x)) & \to & f(x) \end{array} \right.$$

TRS $\mathcal{DM}(R_2)$ が停止性を持つことは自明である．実際に，$f \rhd \diamond$ とすれば再帰経路順序と両立する．したがって，定理 4.1.5より TRS R_2は停止性を持つ．

4.2 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．

関数記号の名前変えとして $F^{\#} = F \cup \{f^{\#} | f \in F\}$ を新たに導入する．$t = f(t_1, \ldots, t_n)$ のとき，$t^{\#} = f^{\#}(t_1, \ldots, t_n)$ と略記する．

また，**関数定義記号**の集合を $D_R = \{root(l) | l \to r \in R\}$ とする．$root(t)$ は，項 t の根と呼び，$root(f(\ldots)) = f$ で定義する．

定義 4.2.1 [AG99] TRS R の**依存対**の集合 $DP(R)$ を以下で定義する．

$$DP(R) = \{\langle s^\#, t^\# \rangle \mid root(t) \in D_R, s \to C[t] \in R\}$$

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 TRS として取り扱う場合がある．

定義 4.2.2 [AG99] 変数を共有しない依存対 $\langle u_1^\#, v_1^\# \rangle, \langle u_2^\#, v_2^\# \rangle, \langle u_3^\#, v_3^\# \rangle, \cdots$ に対して，ある代入θが存在して，すべての $i \geq 1$ に対して，$v_i^\# \theta \xrightarrow[R]{*} u_{i+1}^\# \theta$ を満たすとき，依存対の列

$$\langle u_1^\#, v_1^\# \rangle \langle u_2^\#, v_2^\# \rangle \langle u_3^\#, v_3^\# \rangle \cdots$$

を**依存対列**という．特に無限個の依存対の列を**無限依存対列**という．

依存対列に関して以下の定理が成り立つ．

定理 4.2.3 [AG99] 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．

証明 TRS Rとする．まず，無限依存対列 $\langle u_1^\#, v_1^\# \rangle \langle u_2^\#, v_2^\# \rangle \langle u_3^\#, v_3^\# \rangle \cdots$ が存在すると仮定する．定義より，各依存対 $\langle u_i^\#, v_i^\# \rangle$ に対してある文脈 $C_i[\ ]$ が存在して，規則 $u_i \to C_i[v_i] \in R$ である．また，ある代入θが存在して各依存対間に $v_i^\# \theta \xrightarrow[R]{*} u_{i+1}^\# \theta$ の関係がある．各 $C_i[\ ]$ に対して，$C_i'[\ ] = C_i[\ ]\theta$とすると，無限書換え列

$$u_1\theta \xrightarrow[R]{} C_1'[v_1\theta] \xrightarrow[R]{*} C_1'[u_2\theta] \xrightarrow[R]{} C_1'[C_2'[v_2\theta]] \xrightarrow[R]{*} C_1'[C_2'[u_2\theta]] \xrightarrow[R]{} \cdots$$

が存在する．逆に，TRS R に無限書換え列が存在すると仮定する．無限書換え列のうち，サイズが最小の列を１つ選び，$t_1 \xrightarrow[R]{} t_2 \xrightarrow[R]{} t_3 \xrightarrow[R]{} \ldots$ とする．但し，サイズが最小とは，無限列の先頭の項 t_1に真部分項から始まる無限列が存在しないという意味である．

列の中で最初にリデックスとなる項を t_i $(i \geq 1)$ とする．もし存在しなければ，項 t_1 の真部分項から始まる無限書換え列が存在し，最小性に反する．規則 $l \to r$が存在して $t_i \equiv l\theta \xrightarrow[R]{} r\theta \equiv t_{i+1}$と書ける．次に，項 t_{i+1}の部分項から始まる無限書換え列のうち最小の列を $t_1' \xrightarrow[R]{} t_2' \xrightarrow[R]{} t_3' \xrightarrow[R]{} \ldots$ とする．再び最初にリデックスとなる項を t_j' $(j \geq 1)$ とする．このとき，項 t_j'の根 $root(t_j)$ は関数定義記号であるので，項 t_1'の根も関数定義記号である．それぞれ $u_1 = l$，$v_1\theta = t_1'$とする．このとき列 $t_1 \xrightarrow[R]{} t_2 \xrightarrow[R]{} t_3 \xrightarrow[R]{} \cdots$の最小性より $v_1 \notin V$である．したがって，TRS R には依存対 $\langle u_1^\#, v_1^\# \rangle$ が存在する．

$$
\begin{array}{lllll}
t_1 \xrightarrow[R]{} \cdots \xrightarrow[R]{} & t_i & \xrightarrow[R]{} & t_{i+1} & \xrightarrow[R]{} \cdots \\
& \| & & \| & \\
& l\theta & & r\theta & \\
& \| & & \| & \\
& u_1\theta & & C[t'_1] & \equiv C[v_1\theta]
\end{array}
$$

$$
t'_1 \xrightarrow[R]{} \cdots \xrightarrow[R]{} t'_j \xrightarrow[R]{} t'_{j+1} \xrightarrow[R]{} \cdots
$$

同様の操作を繰り返すと，無限個の依存対

$$\langle u_1^\#, v_1^\# \rangle, \langle u_2^\#, v_2^\# \rangle, \langle u_3^\#, v_3^\# \rangle, \ldots$$

が得られる．依存対列は，変数を共有しないように定義されている．各ステップごとに代入θの定義域を互いに素となるように取ることで，それぞれの代入θを自然に足し合わせた代入θ'が定義できる．代入θ'に対して，無限依存対列

$$\langle u_1^\#, v_1^\# \rangle \langle u_2^\#, v_2^\# \rangle \langle u_3^\#, v_3^\# \rangle \cdots$$

が存在する． □

定理 4.2.3より明らかに次の定理が成り立つ.

定理 4.2.4 [AG99] TRS R に対して，

$$
\begin{array}{ll}
\forall l \to r \in R, & l \gtrsim r, \\
\forall \langle u^\#, v^\# \rangle \in DP(R), & u^\# > v^\#
\end{array}
$$

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（$>= (\gtrsim \setminus \lesssim)$）.

証明 定理 4.2.4の条件を満たす弱書換え順序を $\gtrsim$とする．TRS R が停止性を持たないと仮定すると，定理 4.2.3より，無限依存対列

$$\langle u_1^\#, v_1^\# \rangle \langle u_2^\#, v_2^\# \rangle \langle u_3^\#, v_3^\# \rangle \cdots$$

が存在する．すると，弱書換え順序 $\gtrsim$の狭義部分 $>$ に無限下降列

$$u_1^\# > v_1^\# \gtrsim u_2^\# > v_2^\# \gtrsim u_3^\# > v_3^\# \gtrsim \cdots$$

が存在し，整礎性に矛盾．□

３章で述べたように TRS の停止性判定は，書換え順序によって規則に順序付けを行なう手法が通常である．しかし，定理 4.2.4により，弱書換え順序によって規則と依存対に順序付けをすることで停止性を示すことができる．弱書換え順序は，書換え順序に比べて制限の弱い順序であるため設計が容易である．

また TRS R の依存対の集合 $DP(R)$ を TRS とみなすと，TRS Rと TRS $DP(R)$ の和集合からなる TRS $R \cup DP(R)$ に関して次の性質が成り立つ．

補題 4.2.5 [AG99, KT98] TRS Rが停止性を持つことと TRS $R \cup DP(R)$ が停止性を持つことは同値である．

証明 TRS $R \cup DP(R)$ が停止性を持つならば TRS R が停止性を持つことは自明．

TRS R が停止性を持つと仮定する．TRS R に関する項 t の名前変えした関数記号に関する深さ $|t|$ を以下で定義する．

$$\begin{array}{rcll} |x| & = & 0 & \text{if } x \in V \\ |f(t_1, \ldots, t_n)| & = & \max\{|t_i|\} & \text{if } f \in F \\ |f^{\#}(t_1, \ldots, t_n)| & = & 1 + \max\{|t_i|\} & \text{if } f^{\#} \in F^{\#} \setminus F \end{array}$$

1. まず TRS $R \cup DP(R)$ に項 $t^{\#}$ から始まる無限書換え列が存在しないことを深さ $|t^{\#}|$ に関する帰納法で示す．以下では，TRS $R \cup DP(R)$ の書換え関係を $\xrightarrow[R']{}$ で略記する．

   深さ $|t^{\#}| = 0$ となる項 $t^{\#}$ は存在しない．したがって無限書換え列も存在しない．

   項 $f_1^{\#}(t_1, \ldots, t_n)$ から始まる無限書換え列を考える．帰納法の仮定より，各項 t_iの部分項で部分項 $s_{i_j}^{\#}$から始まる無限書換え列は存在しない．各項 t_iのサイズ最大の部分項 $s_{i_j}^{\#}$をすべて新たなあるひとつの変数 yで置き換えた項を u_iとする．項 t_iに対して，項 $s_{i_j}^{\#}$ がサイズ最大とは，自明でない文脈 $C[\,]$ に対して $s_{i_k}^{\#} = C[s_{i_j}^{\#}]$ となる部分項 $s_{i_k}^{\#}$が存在しないことである．すると，項 $f_1^{\#}(u_1, \ldots, u_n)$ から始まる無限書換え列が存在する．何故なら，項 $s_{i_j}^{\#}$をリデックスとして書換えることのできる規則は，$u^{\#} \to v^{\#} \in DP(R)$ であり，左辺 $u^{\#}$，右辺 $v^{\#}$には，根以外に名前変えした関数記号は出現しない．項 $s_{i_j}^{\#}$を書換えることで得られる項 $s'^{\#}_{i_j}$は根が名前変えされている項である．よって，$C[s_{i_j}^{\#}] \xrightarrow[R']{*} C'[s'^{\#}_{i_j}]$ と書換えられたなら，$C[x] \xrightarrow[R']{*} C'[x]$ という書換えが存在する．項 $f_1^{\#}(t_1, \ldots, t_n)$ から始まる無限書換え列が存在することと，部分項

$s_{i_j}^{\#}$から始まる無限書換え列は存在しないことから，項 $f_1^{\#}(u_1,\ldots,u_n)$ から始まる無限書換え列が存在する．TRS R が停止性を持つことから無限書換え列は以下のようになる．

$$f_1^{\#}(u_1,\ldots,u_n) \xrightarrow[R']{*} f_1^{\#}(v_1,\ldots,v_n) \xrightarrow[R']{} f_2^{\#}(w_1,\ldots,w_m) \xrightarrow[R']{} \cdots$$

さらに，各 u_i には $F^{\#}-F$の関数記号は存在しないことより，次のように書ける．

$$f_1^{\#}(u_1,\ldots,u_n) \xrightarrow[R]{*} f_1^{\#}(v_1,\ldots,v_n) \xrightarrow[DP(R)]{} f_2^{\#}(w_1,\ldots,w_m) \xrightarrow[R']{} \cdots$$

したがって，規則 $l_1^{\#} \to r_1^{\#} \in DP(R)$ が存在して，

$$l_1^{\#}\theta \equiv f_1^{\#}(v_1,\ldots,v_n) \xrightarrow[DP(R)]{} f_2^{\#}(w_1,\ldots,w_m) \equiv r_1^{\#}\theta.$$

項 $f_2^{\#}(w_1,\ldots,w_m)$ から始まる無限書換え列についても同様に繰り返すと，TRS R の無限依存対列

$$\langle l_1^{\#}, r_1^{\#}\rangle\langle l_2^{\#}, r_2^{\#}\rangle\langle l_3^{\#}, r_3^{\#}\rangle\cdots$$

が存在して，TRS R の停止性に矛盾する．

2. TRS $R \cup DP(R)$ に項 $t \in T(F^{\#},V)$ から始まる無限書換え列がないことを示す．項 t の部分項で根が $F^{\#}-F$である部分項を $s_i^{\#}$ とする．1. より項 $s_i^{\#}$ から始まる無限書換え列は存在しない．項 t のサイズ最大の項 $s_i^{\#}$ をすべて新たな変数 yで置き換えた項を t'とする．1. と同様に，項t'から始まる無限書換え列が存在するはずであるが，これは TRS R の停止性に矛盾する． □

4.3 正当性

本研究により，置換消去法の正当性（定理 4.1.5）が依存対の性質を用いることで簡単に示すことができた [NT98a]．

置換消去法の正当性の証明の前に，キャップ部分，分解部分について以下の性質が成り立つことを示す．

補題 4.3.1 任意の項 u, 文脈 $C[\ ]$ に対して，ある文脈 $D[\ ]$ が存在して，

$$D[cap(u)] \in dec(C[u]) \cup \{cap(C[u])\}.$$

証明 文脈 $C[\,]$ の構造に関する帰納法によって示す．消去する関数記号 e とする．文脈 $C[\,] = \Box$ とすると，文脈 $D[\,] = \Box$ が存在して，

$$D[cap(u)] = cap(u) = cap(C[u]).$$

関数記号 $f \in F$，文脈 $C[\,] = f(\ldots, C'[\,], \ldots)$ とする．

帰納法の仮定より，ある文脈 $D'[\,]$ が存在して，$D'[cap(u)] \in dec(C'[u]) \cup \{cap(C'[u])\}$．

1. 関数記号 $f \neq e$ の場合．定義より，

$$\begin{aligned} dec(C'[u]) &\subset dec(C[u]) \\ cap(C[u]) &= f(\ldots, cap(C'[u]), \ldots) \end{aligned}$$

$D'[cap(u)] \in dec(C'[u])$ ならば，文脈 $D[\,] = D'[\,]$ として，

$$D[cap(u)] \in dec(C'[u]).$$

$D'[cap(u)] = cap(C'[u])$ ならば，文脈 $D[\,] = f(\ldots, D'[\,], \ldots)$ とすれば，

$$D[cap(u)] = f(\ldots, D'[cap(u)], \ldots) = f(\ldots, cap(C'[u]), \ldots) = cap(C[u]).$$

したがって，ある文脈 $D[\,]$ が存在して，

$$D[cap(u)] \in dec(C[u]) \cup \{cap(C[u])\}.$$

2. 関数記号 $f = e$ の場合．定義より

$$dec(C'[u]) \cup \{cap(C'[u])\} \subset dec(C[u])$$

文脈 $D[\,] = D'[\,]$ とすれば，

$$D[cap(u)] \in dec(C[u]).$$

□

補題 4.3.1 より，TRS R を置換消去する際，規則 $l \to r \in R$ の右辺 r の部分項 u は，変換後の TRS $\mathcal{DM}(R)$ の右辺に部分項 $cap(u)$ として保存されることが分かる．

置換消去法の正当性を示す．

定理 4.1.5 [Fer95] TRS $\mathcal{DM}(R)$ が停止性を持つならば，TRS Rは停止性を持つ．

証明

仮定より，TRS $\mathcal{DM}(R)$ は停止性を持つ．したがって補題 4.2.5より TRS $\mathcal{DM}(R) \cup DP(\mathcal{DM}(R))$ は停止性を持つ．定理 3.3.6より TRS $\mathcal{DM}(R) \cup DP(\mathcal{DM}(R))$ と両立する整礎かつ単調な $F^{\#}_{\diamond}$-代数 $(A, >)$ が存在する．

この $F^{\#}_{\diamond}$-代数 $(A, >)$ を元に TRS $R \cup DP(R)$ と両立する $F^{\#}$-代数 $(A, >)$ を定義する．今，$F^{\#}_{\diamond}$-代数 $(A, >)$ には消去した関数記号 e の解釈が存在しない．よって写像 e_Aを以下のような定数関数として定義する．

$$e_A(\ldots) = \diamond_A$$

もし $e \in D_R$の場合は，同様に

$$e^{\#}_A(\ldots) = \diamond^{\#}_A$$

で定義する．

関数記号 e（および $e^{\#}$）の解釈を定義した $F^{\#}$-代数 $(A, >)$ は，集合 A，順序 $>$ は変わらないので整礎性を保っている．

$F^{\#}$-代数 $(A, >)$ は弱単調である．関数記号 $f \neq e$ に対しては，写像 f_Aの単調性より自明．関数記号 e に対しては，$a \geq b$ に対して，写像 e_Aの定義より $e_A(\ldots, a, \ldots) = e_A(\ldots, b, \ldots)$．

最後に，$F^{\#}$-代数 $(A, >)$ が TRS $R \cup DP(R)$ と両立することを示す．関数記号 e の解釈より，任意の項 $t \in T(F^{\#}, V)$，割り当てσについて次の関係が成り立つ．

$$[\![t, \sigma]\!]_A = [\![cap(t), \sigma]\!]_A.$$

任意の規則 $l \to r \in R$に対して，置換消去法の定義より，$cap(l) \to cap(r) \in \mathcal{DM}(R)$ が存在する．

任意の規則 $l^{\#} \to u^{\#} \in DP(R)$ に対しては，ある文脈 $C[\ ]$ が存在して，規則 $l \to C[u] \in R$である．補題 4.3.1より，規則 $cap(l) \to D[cap(u)] \in \mathcal{DM}(R)$．項 u の根が関数定義記号であることに注意すると，規則 $cap(l^{\#}) \to cap(u^{\#}) \in DP(\mathcal{DM}(R))$ が存在する．

以上より，規則 $l \to r \in R \cup DP(R)$ に対して，規則 $cap(l) \to cap(r) \in \mathcal{DM}(R) \cup DP(\mathcal{DM}(R))$ が存在する．

$F^{\#}$-代数 $(A, >)$ 上で，任意の割り当てσに対して，

$$[\![l, \sigma]\!]_A = [\![cap(l), \sigma]\!]_A > [\![cap(r), \sigma]\!]_A = [\![r, \sigma]\!]_A.$$

したがって，$l >_A r$である．定理 3.3.6より，整礎で弱単調な $F^{\#}$-代数 $(A, >)$ に対して，順序 $\gtrsim_A$は弱書換え順序である．よって定理 4.2.4より，TRS R は停止性を持つ． □

4.4 改良置換消去法

前節の正当性の証明より，置換消去法によって得られる TRS から冗長な規則を取り除くことができることが分かる．補題 4.3.1によって，TRS R の任意の右辺の部分項が TRS $\mathcal{DM}(R)$ の右辺に保存されていることが示された．しかし依存対を用いた正当性の証明においては，TRS R の右辺の部分項のうち，関数定義記号から始まる部分項さえ保存されていればよい．本研究により，冗長な規則を取り除いた改良置換消去法を定義し，その正当性を示す [NT98a]．

置換消去法の分解部分を以下のように定義しなおす．

定義 4.4.1 [NT98a]

$$dec'(t) = dec(t) \setminus T(F - D_R, V)$$

新たに定義しなおした分解部分 $dec'(t)$ は，元の分解部分 $dec(t)$ から関数定義記号が出現しない項を取り除いたものである (図 4.2)．

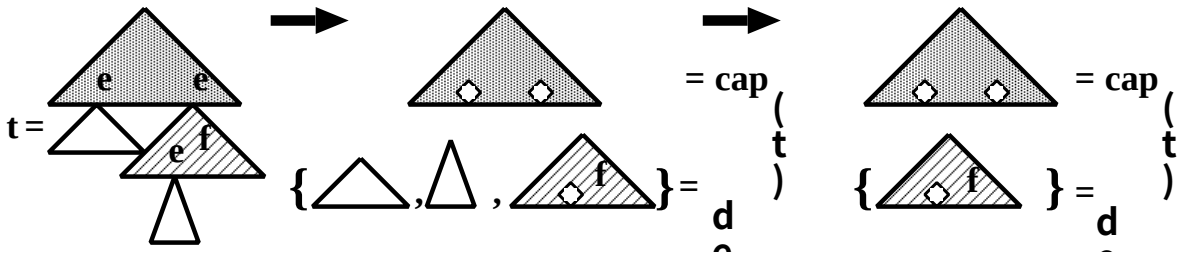


図 4.2: 分解部分の改良 ($f \in D_R$)

新たな分解部分 $dec'(t)$ により，**改良置換消去法**を以下で定義する．

定義 4.4.2 [NT98a]

$$\mathcal{DM}'(R) = \{cap(l) \to r' \mid l \to r \in R, r' \in dec'(r) \cup \{cap(r)\}\}$$

改良置換消去法についても正当性が成り立つ．

定理 4.4.3 [NT98a] TRS $\mathcal{DM}'(R)$ が停止性を持つならば，TRS Rは停止性を持つ．

次の補題 4.4.4により，証明は置換消去法の正当性と同様に示せる．

補題 4.4.4 根が関数定義記号である項u，文脈$C[\ ]$に対して，ある文脈$D[\ ]$が存在して，

$$D[cap(u)] \in dec'(C[u]) \cup \{cap(C[u])\}.$$

証明

1. $root(u) \neq e$ の場合．補題 4.3.1より，

   $$D[cap(u)] \in dec(C[u]) \cup \{cap(C[u])\}.$$

   項 $D[cap(u)]$ は関数定義記号を含む．分解部分 $dec'(u)$ の定義より，

   $$D[cap(u)] \in dec'(C[u]).$$

2. $root(u) = e$ の場合．

   ある文脈 $D[\ ]$ が存在して，$cap(C[u]) = D[\diamond]$ と書ける．

   $$D[cap(u)] = D[\diamond] = cap(C[u]).$$

□

改良置換消去法が正当な変換であることは確かめられた．次に置換消去法を真に改良していることを確かめる．

TRS R に対して，従来の置換消去法によって得られる TRS $\mathcal{DM}(R)$ と改良置換消去法によって得られる TRS $\mathcal{DM}'(R)$ の間には包含関係 $\mathcal{DM}'(R) \subset \mathcal{DM}(R)$．が成り立つ．したがって，TRS $\mathcal{DM}(R)$ で停止性が示せるならば TRS $\mathcal{DM}'(R)$ でも停止性が示せる．さらに，改良したことによって以前の置換消去法では停止性を示せなかった例が存在する．したがって，改良置換消去法は真の拡張と言える．

例 4.4.5 TRS R_3を考える．

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

消去する関数記号を g として置換消去法で変換すると以下のようになる．

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

TRS $\mathcal{DM}(R_3)$ は明らかに停止性を持っていない．

$$f(a) \underset{R_3}{\to} f(b) \underset{R_3}{\to} f(a) \underset{R_3}{\to} \cdots$$

一方，改良置換消去法で変換すると以下のようになる．

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

改良置換消去法では，分解部分 $dec'(g(a))$ には関数定義記号が出現しない項 a は含まれないので，規則 $b \to a$ が省かれる．

TRS $\mathcal{E}'(R_3)$ は明らかに停止性を持つ．実際，$a \rhd b \rhd \diamond$ の優先順位で，再帰経路順序によって順序付けができる．よって，定理 4.4.3 より TRS R_3は停止性を持つ．

第 5 章

切り落とし法

依存対を用いた置換消去法の正当性の証明では，依存対の性質，特に定理 4.2.4が本質的な活躍をした．定理 4.2.4では，停止性判定において，通常用いられる書換え順序ではなく，より条件の弱い弱書換え順序を用いた．弱書換え順序は，書換え順序に比べて設計が容易である．前章の置換消去法の正当性の証明において，F-代数 $(A, >)$ から生成される弱書換え順序$>_A$の設計は，関数記号 $e \in F$の解釈を定数関数によって定義している．

$$e_A(\ldots) = \text{定数}$$

本来，F-代数によって TRS の停止性を保証する場合には，このような解釈は不適当である．何故なら，単調性

$$a > b \Rightarrow f(\ldots, a, \ldots) > f(\ldots, b, \ldots)$$

が成り立たなくなるからである．実際，どのような a, b に対しても，$e_A(\ldots, a, \ldots) = e_A(\ldots, b, \ldots)$ である．しかし，すでに見たように弱単調性が成り立つため，順序$>_A$は弱書換え順序を満たしている．

前章では，弱書換え順序を F-代数を用いて定義したが，この章では，弱書換え順序を構文的に生成する切り落とし関数を定義する．切り落とし関数を用いた弱書換え順序の構成を切り落とし法と呼ぶ [KT98]．

切り落とし法を用いて変換手法の正当性の証明を一般化する．

5.1 切り落とし関数

定義 5.1.1 [KT98] 関数記号 $f \in F, arity(f) = n$ に対して，**切り落とし関数**πは，状態 $\pi(f)$ として自然数の集合 $\{i_1, \ldots, i_m\}$，または，自然数 i を取る．但し，$m \geq 0, 1 \leq i_1 < \cdots < i_m \leq n$，$1 \leq i \leq n$.

切り落とし関数πを項から項への関数に拡張する．

定義 5.1.2 [KT98]

$$\begin{array}{rcll} \pi(x) & = & x & \text{if } x \in V \\ \pi(f(t_1, \ldots, t_n)) & = & \left\{ \begin{array}{ll} f(\pi(t_{i_1}), \ldots, \pi(t_{i_m})) & \text{if } \pi(f) = \{i_1, \ldots, i_m\} \\ \pi(t_i) & \text{if } \pi(f) = i \end{array} \right. \end{array}$$

項の集合 T，項の対の集合 R についても自然に拡張する．

$$\begin{array}{rcl} \pi(T) & = & \{ \quad \pi(t) \quad | \quad t \in T \quad \} \\ \pi(R) & = & \{ \quad (\pi(l), \pi(r)) \quad | \quad (l, r) \in R \quad \} \end{array}$$

切り落とし関数によって得られる項では，関数記号 $f \in F$の引数 $arity(f)$ が異なるか，もしくは f が存在しないことがある．関数記号 Fに対して，関数記号 F_πを以下で定義する．

$$\begin{array}{c} F_\pi = F - \{f \in F \quad | \, \pi(f) = i \in N\} \\ arity(f) = m \qquad \text{if } f \in F_\pi, \pi(f) = \{i_1, \ldots, i_m\} \end{array}$$

このとき，$\pi(T(F, V)) = T(F_\pi, V)$ である．項の集合$T(F_\pi, V)$ 上に順序>が定義されているとするとき，項の集合 $T(F, V)$ 上の順序$>_\pi$，および擬順序 $\gtrsim_\pi$を以下で定義する．

定義 5.1.3 [KT98]

$$\begin{array}{rcl} s >_\pi t & \stackrel{\text{def}}{\Longleftrightarrow} & \pi(s) > \pi(t) \\ s \gtrsim_\pi t & \stackrel{\text{def}}{\Longleftrightarrow} & \pi(s) \geq \pi(t) \end{array}$$

順序$>_\pi$，擬順序 $\gtrsim_\pi$を切り落とし関数π によって順序>から生成された順序，擬順序と呼ぶ．明らかに，順序$>_\pi$は擬順序 $\gtrsim_\pi$の狭義部分，すなわち，$>_\pi = (\gtrsim_\pi \setminus \lesssim_\pi)$ である．

二項関係 $\gtrsim_\pi$について以下の定理が成り立つ．

定理 5.1.4 [KT98] 切り落とし関数πによって書換え順序>から生成された擬順序 $\gtrsim_\pi$は弱書換え順序である.

証明 書換え順序>を項の集合 $T(F_\pi, V)$ 上の二項関係とする.

集合 A_πを項の集合$T(F_\pi, V)$, 順序>を与えられた書換え順序とする F_π-代数 $(A_\pi, >)$ を考える.

明らかに F_π-代数 $(A_\pi, >)$ は単調かつ整礎である.

F_π-代数 $(A_\pi, >)$ を用いて, 以下の F-代数 $(A, >)$ を定義する.

$$\begin{array}{rcll} A & = & A_\pi & \\ f_A(x_1, \ldots, x_n) & = & f_{A_\pi}(x_{i_1}, \ldots, x_{i_m}) & \text{if} \quad \pi(f) = \{i_1, \ldots i_m\} \\ f_A(x_1, \ldots, x_n) & = & x_i & \text{if} \quad \pi(f) = i \end{array}$$

定義より明らかに以下が成り立つ.

$$\forall \theta, [\![\pi(t), \theta]\!]_{A\pi} = [\![t, \theta]\!]_A$$

順序>は書換え順序であるので, F-代数 $(A, >)$ が整礎であることは自明.

F-代数 $(A, >)$ の弱単調性を示す. 以下, 写像 f_Aの第 i 引数に a を適用した結果を $f_A(\ldots, a, \ldots)_i$と書くことにする. 任意の関数記号 fについて,

$\pi(f) = \{i_1, \ldots, i_m\}$ のとき,

$$\begin{array}{rcl} a \geq b & \Rightarrow & f_{A_\pi}(\ldots, a, \ldots) \geq f_{A_\pi}(\ldots, b, \ldots) \\ & \Rightarrow & \left\{ \begin{array}{rcll} f_A(\ldots, a, \ldots)_i & \geq & f_A(\ldots, b, \ldots)_i & \text{if} \quad i \in \{1, \ldots, i_m\} \\ f_A(\ldots, a, \ldots)_i & = & f_A(\ldots, b, \ldots)_i & \text{if} \quad i \notin \{1, \ldots, i_m\} \end{array} \right. \\ & \Rightarrow & f_A(\ldots, a, \ldots) \geq f_A(\ldots, b, \ldots) \end{array}$$

$\pi(f) = i$ のとき,

$$\begin{array}{rcl} a \geq b & \Rightarrow & \left\{ \begin{array}{rcll} f_A(\ldots, a, \ldots)_i & \geq & f_A(\ldots, b, \ldots)_i & \\ f_A(\ldots, a, \ldots)_j & = & f_A(\ldots, b, \ldots)_j & \text{if} \quad j \neq i \end{array} \right. \\ & \Rightarrow & f_A(\ldots, a, \ldots) \geq f_A(\ldots, b, \ldots) \end{array}$$

よって, F-代数 $(A, >)$ は弱単調である. 定理3.3.5より, 擬順序 $\gtrsim_A$は弱書換え順序である. 最後に $\gtrsim_A = \gtrsim_\pi$を確かめる.

$$
\begin{aligned}
s \gtrsim_\pi t &\iff \pi(s) \geq \pi(t) \\
&\iff \forall\theta, \pi(s)\theta \geq \pi(t)\theta \\
&\iff \forall\theta, [\![\pi(s), \theta]\!]_{A\pi} \geq [\![\pi(t), \theta]\!]_{A\pi} \\
&\iff \forall\theta, [\![s, \theta]\!]_A \geq [\![t, \theta]\!]_A \\
&\iff s \gtrsim_A t
\end{aligned}
$$

□

5.2 変換の正当性

切り落とし関数 π を用いることで，変換の正当性は，次の関係が成り立てば保証される．

定理 5.2.1 切り落とし関数 π が存在して包含関係

$$\pi(R \cup DP(R)) \subset R' \cup DP(R')$$

が成り立つとき，TRS R' が停止性を持つならば，TRS R は停止性を持つ．

証明 TRS R'が停止性を持つことから，補題 4.2.5より，TRS $R' \cup DP(R')$ と両立する書換え順序>が存在する．切り落とし関数 πによって書換え順序>から生成される弱書換え順序 $\gtrsim_\pi$は，TRS $R \cup DP(R)$ と両立する．

$$
\begin{aligned}
l \to r \in R \cup DP(R) &\Longrightarrow \pi(l) \to \pi(r) \in \pi(R \cup DP(R)) \\
&\Longrightarrow \pi(l) \to \pi(r) \in R' \cup DP(R') \\
&\Longrightarrow \pi(l) > \pi(r) \\
&\Longrightarrow l >_\pi r
\end{aligned}
$$

よって，定理 4.2.4より TRS Rは停止性を持つ． □

置換消去法で消去する関数記号 e に対して，切り落とし関数πを以下で定義する．

$$
\begin{aligned}
\pi(e) &= \emptyset \\
\pi(f) &= \{1, \ldots, arity(f)\} \quad \text{if} \quad f \neq e
\end{aligned}
$$

また置換消去法で現れる定数 $\diamond$ は，定数 $e \in F_\pi$と同一視する．関数記号 $e \in F_\pi$は関数記号 $e \in F$とは別の関数記号であるので問題はない．すると，$cap(t) = \pi(t)$ である．前章で

確かめた通り，$l \to r \in R \cup DP(R)$ に対して，$cap(l) \to cap(r) \in \mathcal{DM}(R) \cup DP(\mathcal{DM}(R))$ が存在する．

前章で行なった置換消去法の改良は，$\mathcal{DM}(R)$ から包含関係が成り立つのために不必要な規則を取り除くことで定義した．

次章では，置換消去法以外の変換手法を紹介するが，それらの変換手法についても，条件付きではあるが上記の包含関係が成り立つ．

第 6 章

消去法

この章では，すでに紹介した置換消去法以外の消去法である分配消去法，一般消去法を紹介する．これらの消去法は，置換消去法と同様に，ある関数記号を消去することでより構造の単純な TRS に変換する．置換消去法と異なるのは，消去した関数記号の直下の部分項の扱いである．置換消去法においては，消去した関数記号の直下の部分項は，分解部分に保存されて，それ自体が規則の右辺を構成した．分配消去法では，部分項は，残余部分と呼ばれる集合に保存される．残余部分は，消去した関数記号の出現を直下の部分項と置き換えて得られる．一般消去法は，消去する関数記号に状態を定義することで，部分項の扱いを分解，残余から選択できる変換手法である．一般消去法は，状態の取り方によって置換消去法にも分配消去法にもなり，分解，残余を混在させることによってどちらの消去法でも扱えなかった TRS にも適用可能となる [Fer95]．分配消去法，一般消去法について，切り落とし法による正当性の証明を行なう．また，置換消去法と同様に，改良可能であることを示す [NKT99, NT98b]．

6.1 分配消去法

分配消去法は，分配則と呼ばれる規則を消去する変換手法である．変換の際，分配則を形成する関数記号を消去するが，分配消去法においては，消去する関数記号は，直下の部分項によって置き換えられる [Zan94]．

定義 6.1.1 [Zan94] 関数記号 $a, arity(a) \geq 1$ とし，文脈 $C[\ ]$ には関数記号 a が出現しないとき，規則

$$C[a(x_1, \ldots, x_n)] \to a(C[x_1], \ldots, C[x_n])$$

を関数記号 a に関する**分配則**と呼ぶ.

関数記号 a に関して項 t の**残余部分** $E_a(t)$ を定義する.

定義 6.1.2 [Zan94] $E_a : T(F, V) \to \mathcal{P}(T(F - \{e\}, V))$,

$$E_a(t) = \begin{cases} \{t\} & \text{if} \quad t \in V \\ \{f(u_1, \ldots, u_m) \mid u_i \in E_a(t_i), 1 \leq i \leq m\} & \text{if} \quad t = f(t_1, \ldots, t_m), f \neq a \\ \bigcup_n E_a(t_i) & \text{if} \quad t = a(t_1, \ldots, t_n) \end{cases}$$

TRS R から 関数記号 a に関する分配則を除いたものを TRS R_aとする.

$$R_a = R - \{C[a(x_1, \ldots, x_n)] \to a(C[x_1], \ldots, C[x_n])\}$$

残余部分によって，**分配消去法**は以下のように定義される.

定義 6.1.3 [Zan94]

$$\mathcal{DS}(R) = \{l \to r' \mid l \to r \in R_a, r' \in E_a(r)\}$$

分配消去法は，分配則がない TRS についても適用できる．その場合，置換消去法と同様に関数記号を消去する変換手法となる．しかし，置換消去法と違い，消去する関数記号はその直下の部分項の残余部分によって置き換えられる.

例 6.1.4 次の TRS R_4 を関数記号+を消去することで TRS $\mathcal{DS}(R_4)$ に変換する.

$$R_4 = \begin{cases} \times(+(x, y), z) \to +(\times(x, z), \times(y, z)) & (1) \\ \times(s(x), s(y)) \to s(+(+(\times(x, y), x), y)) & (2) \end{cases}$$

規則 (1) は，+に関する分配則になっている．規則 (2) の右辺の残余部分，および変換して得られる TRS $\mathcal{DS}(R_4)$ は，以下のようになる.

$$E_+(s(+(+(\times(x, y), x), y))) \quad = \quad \{ \quad s(\times(x, y)), \quad s(x), \quad s(y) \quad \}$$

$$\mathcal{DS}(R_4) \quad = \quad \begin{cases} \times(s(x), s(y)) & \to & s(\times(x, y), y) \\ \times(s(x), s(y)) & \to & s(x) \\ \times(s(x), s(y)) & \to & s(y) \end{cases}$$

分配消去法について，以下の定理が成り立つ．

定理 6.1.5 [Zan94] TRS R，消去する関数記号 a とし，分配則以外の規則には左辺に関数記号 a は出現しないとする．

1. TRS $\mathcal{DS}(R)$ が全停止性を持つことと，TRS R が全停止性を持つことは同値．
2. TRS $\mathcal{DS}(R)$ が右線形ならば，TRS $\mathcal{DS}(R)$ が単純停止性を持つことと，TRS R が単純停止性を持つことは同値．
3. TRS $\mathcal{DS}(R)$ が右線形かつ停止性を持つならば，TRS R は停止性を持つ．

左辺に関数記号 a が現れない TRS，すなわち分配則がない TRS ，については右線形の条件を外すことができる [MOZ96]．

定理 6.1.6 [MOZ96] TRS R とし，左辺には関数記号 a は出現しないとする．このとき，TRS $\mathcal{DS}(R)$ が停止性を持つならば，TRS R は停止性を持つ．

証明は，以下で行なう改良分配消去法の証明から自明に導かれる．分配則がない場合について，分配消去法を以下のように改良する．

定義 6.1.7 消去する関数記号を a として，切り払い関数πを以下で定義する．但し，$1 \leq i \leq arity(a)$ とする．

$$\begin{aligned} \pi(a) &= i \\ \pi(f) &= \{1, \ldots, arity(f)\} \quad \text{if} \quad f \neq a \end{aligned}$$

改良分配消去法を以下で定義する．

$$\begin{aligned} E'_a(t) &= E_a(t) \setminus T(F - D_R, V) \\ \mathcal{DS}'(R) &= \{l \to r' \mid l \to r \in R_a, r' \in E'_a(r) \cup \{\pi(r)\}\} \end{aligned}$$

任意の項 t について，明らかに$\pi(t) \in E_a(t)$ である．したがって，$E'_a(t) \subset E_a(t)$ が成り立ち，左辺に消去する関数記号 a が現れない TRS R については，$\mathcal{DS}'(R) \subset \mathcal{DS}(R)$ である．

次の補題が成り立つ．

補題 6.1.8 任意の項 u，文脈 $C[\ ]$ に対して，ある文脈 $D[\ ]$ が存在して，

$$D[\pi(u)] \in E_a(C[u])$$

証明 文脈 $C[\ ]$ の帰納法で示す．文脈 $C[\ ] = \Box$ とすると，文脈 $D[\ ] = \Box$ が存在して，

$$D[\pi(u)] = \pi(u) = \pi(C[u]).$$

関数記号 $f \in F$，文脈 $C[\ ] = f(t_1, \ldots, t_{i-1}, C'[\ ], t_{i+1}, \ldots, t_n)$ とする．
帰納法の仮定より，ある文脈 $D'[\ ]$ が存在して，$D'[\pi(u)] \in E_a(C'[u]) \cup \{\pi(C'[u])\}$．

1. 関数記号 $f \neq e$ の場合．定義より，

$$E_a(C[u]) = \{f(u_1, \ldots, u_{i-1}, C'[u], u_{i+1}, \ldots, u_n) \mid u_j \in E_a(t_j), j \neq i\}$$

任意の項 t_i について，$\pi(t_i) \in E_a(t_i)$ である．また，帰納法の仮定より，$D'[\pi(u)] \in \pi(C'[u])$．したがって，

$$D[\ ] = f(\pi(t_1), \ldots, \pi(t_{i-1}), D'[\ ], \pi(t_{i+1}), \ldots, \pi(t_n))$$

とすれば，

$$f(\pi(t_1), \ldots, \pi(t_{i-1}), D'[\pi(u)], \pi(t_{i+1}), \ldots, \pi(t_n)) \in E_a(C[u]).$$

2. 関数記号 $f = e$ の場合．定義より

$$E_a(C'[u]) \subset E_a(C[u])$$

文脈 $D[\ ] = D'[\ ]$ とすれば，

$$D[cap(u)] \in dec(C[u]).$$

□

補題 6.1.9 関数記号 a が左辺に現れないとき，根が関数定義記号である項 u，文脈 $C[\ ]$ に対して，ある文脈 $D[\ ]$ が存在して，

$$D[\pi(u)] \in E'_a(C[u]).$$

証明 補題 6.1.8より，

$$D[\pi(u)] \in E_a(C[u]).$$

仮定より，消去する関数記号 a は規則の左辺に現れない．したがって，根が関数定義記号であることから，$root(u) \neq a$ である．このとき，項 $D[\pi(u)]$ には関数定義記号が出現する．よって，

$$D[\pi(u)] \in E'_a(C[u]).$$

□

定理 6.1.10 消去する関数記号 a が規則の左辺に現れないとき，TRS $\mathcal{DS}'(R)$ が停止性を持つならば，TRS R は停止性を持つ．

証明 定義より $l \to \pi(r) \in \mathcal{DS}'(R)$ である．左辺 l には関数記号 a が出現しないので，明らかに $l = \pi(l)$ である．よって，

$$\pi(R) \subset \mathcal{DS}'(R).$$

また，補題 6.1.9より，

$$\pi(DP(R)) \subset DP(\mathcal{DS}'(R)).$$

以上より，包含関係

$$\pi(R \cup DP(R)) \subset \mathcal{DS}'(R) \cup DP(\mathcal{DS}'(R)).$$

が成り立ち，定理 5.2.1より，TRS R は停止性を持つ． □

左辺に消去する関数記号 a が出現しないときは，包含関係 $\mathcal{DS}'(R) \subset \mathcal{DS}(R)$ が成り立つ．よって，定理 6.1.10から定理 6.1.6は自明である．

6.2 一般消去法

置換消去法と分配消去法は適当な関数記号を消去させることで変換を行なう手法であった．関数記号を消去することで失われた部分項は，置換消去法においては分解部分 $dec(t)$ で保存され，分配消去法においては残余部分 $E_a(t)$ で保存されていた．

最後に紹介する一般消去法[1]は，置換消去法と分配消去法を融合させた変換手法である．一般消去法は，消去する関数記号に状態関数を定義し，失われた部分項の保存を分解，残余から選ぶことができる．状態関数は，部分項それぞれについて分解，残余を混在させて定義できるため，一般消去法は，置換消去法，分配消去法を含むだけでなく，どちらでも停止性を示せなかった TRS についても適用可能である [Fer95]．

定義 6.2.1 [Fer95] **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．

状態関数において，集合 Iは残余する部分項に対応する．

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） [Fer95]．

定義 6.2.2 [Fer95] $cap_i : T(F, V) \to T(F, V)$,

$$cap_i(t) = \begin{cases} t & \text{if } t \in V \\ f(t_1, \ldots, t_n) & \text{if } t = f(t_1, \ldots, t_n) \\ cap_i(t_i) & \text{if } t = e(t_1, \ldots, t_n), i \neq 0 \\ \diamond & \text{if } t = e(t_1, \ldots, t_n), i = 0 \end{cases}$$

但し，$\diamond$ は新たな定数とする．

定義 6.2.3 [Fer95] $E, E_i : T(F, V) \to \mathcal{P}(T(F, V))$,

$$E_i(t) = \begin{cases} \{t\} & \text{if } t \in V \\ \{f(u_1, \ldots, u_n) \mid u_k \in E_i(t_k)\} & \text{if } t = f(t_1, \ldots, t_n), f \neq e \\ E(t_i) & \text{if } t = e(t_1, \ldots, t_n) \end{cases}$$

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

[1]"general dummy elimination" [Fer95].

定義 6.2.4 [Fer95] $dec : T(F,V) \to \mathcal{P}(T(F,V))$,

$$dec(t) \;=\; \begin{cases} \phi & \text{if} \quad t \in V \\ \bigcup_n dec(t_i) & \text{if} \quad t = f(t_1,\ldots,t_n), f \neq e \\ \bigcup_n dec(t_i) \cup \bigcup_{k \notin I} E(t_k) & \text{if} \quad t = e(t_1,\ldots,t_n) \end{cases}$$

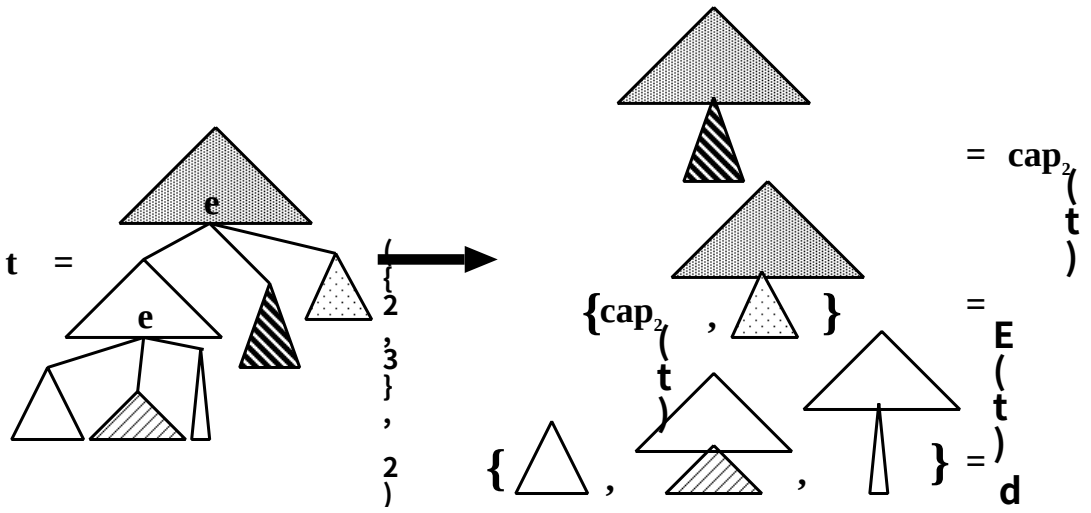


図 6.1: $\tau(e) = (\{2,3\},2)$ でのキャップ部分 $cap_2(t)$，残余部分 $E(t)$，分解部分 $dec(t)$

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 [Fer95].

定義 6.2.5 [Fer95] $\tau(e) = (I,i)$ のとき，

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

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の要素に対応する引数の部分項を残余し，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 $i \neq 0$ ならば e の直下で i 番目の部分項に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 $\diamond$ に，$I \neq \emptyset$ ならば Iの要素に対応する e の直下の部分項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_iの残余部分 $E(t_i)$ の和集合である（図 6.1）．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 $(\tau(e) = (\emptyset, 0))$，分配則を含まない分配消去法は，全て残余したもの $(\tau(e) = (\{1, 2, ..., n\}, i))$ に対応する．

一般消去法について次の定理が成り立つ．

定理 6.2.6 [Fer95] $\mathcal{E}(R)$ が停止性を持つならば，R は停止性を持つ．

証明は，文献 [Fer95] を参照されたい．次節では，特に消去する関数記号 e が関数定義記号でない場合，すなわち $e \notin D_R$，について改良を行なう．

例 6.2.7 一般消去法により，以下の TRS R_5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 $\mathcal{E}(R_5)$ は以下のようになる．

$$R_5 = \{f(x,x) \to f(g(a,x), g(b,x))\}$$

$$\begin{aligned} cap_i(r) &= f(a,b) \\ E(r) &= \{f(a,b)\} \\ dec(r) &= \{x\} \\ \mathcal{E}(R_5) &= \begin{cases} f(x,x) \to f(a,b) \\ f(x,x) \to x \end{cases} \end{aligned}$$

TRS $\mathcal{E}(R_5)$ は明らかに停止性を持つので定理 6.2.6 より R_5は停止性を持つ．

TRS R_5 は置換消去法，分配消去法のどちらでも停止性のある TRS に変換することができない例である．

$$\mathcal{DE}(R_5) = \begin{cases} f(x,x) \to f(\diamond, \diamond) \\ f(x,x) \to a \\ f(x,x) \to b \\ f(x,x) \to x \end{cases}$$

$$\mathcal{DS}(R_5) = \begin{cases} f(x,x) \to f(a,b) \\ f(x,x) \to f(a,x) \\ f(x,x) \to f(x,b) \\ f(x,x) \to f(x,x) \end{cases}$$

6.3 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 6.3.1

$$\begin{aligned} E'(t) &= E(t) \setminus T(F - D_R, V) \\ dec'(t) &= dec(t) \setminus T(F - D_R, V) \end{aligned}$$

定義 6.3.2 消去する関数記号 $e \notin D_R$で，$\tau(e) = (I, i)$ のとき，

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

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

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．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．改良一般消去法の正当性が成り立つことを示す．

定理 6.3.3 [NKT99, NT98b] $\mathcal{E}'(R)$ 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 6.2.6 より Rは停止性を持つ．以下では，$e \notin D_R$と仮定する．状態関数$\tau(e) = (I, i)$ に対して，切り落とし関数πを以下で定義する．

$$\pi(e) = \begin{cases} \emptyset & i = 0 \\ i & i \neq 0. \end{cases}$$

すると，$cap_i(t) = \pi(t)$ である．したがって，

$$\pi(R) \subset \mathcal{E}(R).$$

また補題 4.3.1，補題 6.1.8 及び，補題 4.4.4，補題 6.1.9の証明と同様にして以下が成り立つ．根が関数定義記号の項 u，任意の文脈 $C[\ ]$ に対して，ある文脈 $D[\ ]$ が存在して，

$$D[\pi(u)] \in \{\pi(C[u])\} \cup E'(C[u]) \cup dec'(C[u]).$$

よって依存対の集合に関しても，

$$\pi(DP(R)) \subset DP(\mathcal{E}'(R)).$$

以上より，

$$\pi(R \cup DP(R)) \subset \mathcal{E}'(R) \cup DP(\mathcal{E}'(R)).$$

補題 4.2.5より，TRS $\mathcal{E}'(R) \cup DP(\mathcal{E}'(R))$ は停止性を持つ．よって，定理 5.2.1より TRS R は停止性を持つ． □

明らかに，$\mathcal{E}'(R) \subseteq \mathcal{E}(R)$ が成り立つので，$\mathcal{E}(R)$ 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 6.2.6の証明にもなっている．

例 6.3.4 ４章の例題 4.4.5の TRS R_3は，置換消去法では停止性のある TRS に変換できない例であった．TRS R_3は，一般消去法でも停止性を持つ TRS に変換できない．

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

実際，どのような状態関数に対しても，変換後の規則に $b \to a$ が加えられる．しかし，改良一般消去法において，関数記号 g を 状態 $\tau(g) = (\emptyset, 0)$ で消去すると，改良置換消去法と同様に以下の TRS $\mathcal{E}'(R_3)$ を得る．

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

すでに見たように，TRS $\mathcal{E}'(R_3)$ は停止性を持つ．よって，定理 6.3.3より TRS R_3は停止性を持つことが分かる．したがって，改良一般消去法は従来の一般消去法の真の拡張になっている．

第 7 章

結論

7.1 成果

本研究では，置換消去法，及び分配消去法，一般消去法（以下，まとめて消去法と呼ぶ）の変換の正当性を依存対を用いることで証明した．また，証明を通して各消去法の改良に成功した．改良された消去法で変換して得られる TRS は従来の消去法によって得られる TRS から冗長な規則を取り除くことで定義したため，従来の消去法によって停止性を示すことのできる TRS は，改良した消去法にも適用可能である．さらに，従来の消去法では停止性のある TRS に変換できなかった TRS についても改良した消去法によって停止性を示すことできた．

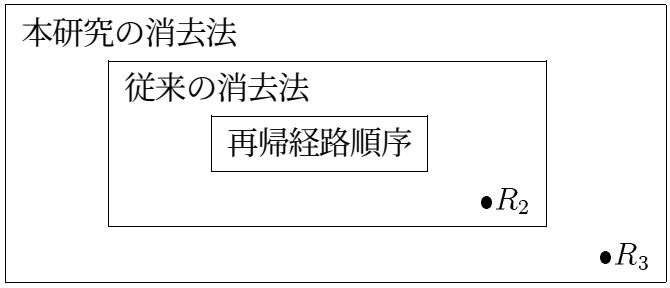
消去法の動機のひとつに，決定性のある停止性のクラスを広げることがある．決定性のある停止性のクラスとは，ある決定可能な停止性判定法によって停止性を示すことのできる TRS のクラスである．再帰経路順序による停止性判定は決定可能である．再帰経路順序では停止性を示すことのできなかった TRS R_2 は，置換消去法によって変換することで，再帰経路順序によって停止性を示すことのできる TRS $\mathcal{DM}(R_2)$ に変換できた．

$$
\begin{aligned}
R_2 &= \left\{ f(f(x)) \rightarrow f(g(f(x))) \right\} \\
\mathcal{DM}(R_2) &= \left\{ \begin{array}{lcl} f(f(x)) & \rightarrow & f(\diamond) \\ f(f(x)) & \rightarrow & f(x) \end{array} \right.
\end{aligned}
$$

消去法による変換は，規則の数が有限ならば決定可能な手続きである．したがって，与えられた TRS を消去法によって変換し，変換後の TRS の停止性を再帰経路順序によって判定をするという手続きは決定可能である．

今回改良した消去法は，従来の消去法では適用できなかった TRS R_3 を再帰経路順序によって停止性を示すことができる TRS $\mathcal{E}'(R_3)$ に変換することができた．すなわち，決定性のある停止性のクラスを広げることができたことになる（下表）．本研究によって，TRS の停止性判定の実装に対して十分意義のある結果を得たと言える．

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



7.2 今後の課題

置換消去法については，特に条件もなく改良することができたが，一般消去法は，関数定義記号を消去する場合には改良できなかった．何故なら，関数定義記号を消去してしまうと，それに伴い変換した先の関数定義記号が変化するためである．変換後の TRS では，関数定義記号が減るだけでなく，場合によっては元々無かった関数定義記号が出現することもある．したがって，変換後の TRS の依存対だけでは，元の TRS の依存対の間に順序付けができない．関数定義記号を消去する場合の改良は，今後の課題のひとつである．同様に，分配消去法における分配則がある場合の改良も今後の課題とする．

依存対による変換手法の解析は，切り落とし法を用いることで一般化された．本論文では，ある特別な切り落とし関数（$\pi(e) = \emptyset, \pi(e) = i$）を適用することで，消去法の正当性を示したが，一般の切り落とし関数を用いることで，変換手法を定義することもできる [KT98]．今後は，消去法以外の変換手法に対しても依存対による解析を試みることで，改良もしくは新たな変換手法の提案を行なう予定である．

謝辞

本論文の作成および研究生活全般に対して，多大な御指導を頂いた外山芳人教授，鈴木太朗助手，草刈圭一朗氏，活発な議論の場を与えて下さった外山研究室の人々とそれ以外の人々に感謝する．また，有形無形の援助で学生生活を支えてくれた家族に謝意を表する．

参考文献

[Art97] T.Arts, Automatically proving termination and innermost normalisation of term rewriting systems, PhD thesis, Universiteit Utrecht ,1997.

[AG97a] T.Arts,J.Giesl, Automatically proving termination where simplification orderings fail, In M.Dauchet, editor, *Proceeding of the 22nd International Colloquium on Trees in Algebra and Programming, CAAP'97 Lecture Notes in Computer Science* 1214 ,pp.261-272, 1997.

[AG97b] T.Arts,J.Giesl, Proving innermost normalization automatically, In *Proceeding of the 8th International Conference on Rewriting Techniques and Applications, RTA-97, Lecture Notes in Computer Science* 1232 ,pp.157-171, 1997.

[AG99] T.Arts,J.Giesl, Termination of term rewriting using dependency pairs, To appear in *Theoretical Computer Science.*

[BN98] F.Baader,T.Nipkow, Term rewriting and all that, Cambridge University Press, 1998.

[Der87] N.Dershowitz, Termination of rewriting, *Jounal of Symbolic Computation* 3(1), pp.69-116,1987.

[Fer95] M.C.F.Ferreira, Termination of term rewriting -Well-foundedness, totality and transformations, PhD thesis,Universiteit Utrecht, 1995.

[FZ95] M.C.F.Ferreira,H.Zantema, Dummy elimination: Making termination easier, In H.Reichel, editor, *Proceeding of the 10th International Conference on Fundamentals*

of Computation Theory, FCT'95, Lecture Notes in Computer Science 965 ,pp.243-252, 1995.

[FZ96] M.C.F.Ferreira, H.Zantema, Total Termination of Term Rewriting, *Applicable Algebra in Engineering, Communication and Computing* 7, pp.133-162, 1996.

[KT98] 草刈 圭一朗，外山 芳人， 切り落とし法に基づく項書換え系の停止性判定， 信学技法，Vol.98，No.432，電子情報通信学会技術研究報告 COMP98-57，pp.49-56，1998.

[MOZ96] Aart Middeldorp, Hitoshi Ohsaki, Hans Zantema, Transforming Termination by Self-Labelling, In M.A.McRobbie and J.K.Slaney, editors, *Proceeding of the 13th International Conference on Automated Deduciton, CADE-13, Lecture Notes in Artificial Intelligence* 1104,pp.373-387, 1996.

[MZ96] Aart Middeldorp, Hans Zantema, Simple Termination of Rewrite Systems, *Theoretical Computer Science* 175(1), pp. 127-158, 1997.

[NKT99] 中村 正樹，草刈 圭一朗，外山 芳人， 消去法による項書換え系の停止性判定について，電子情報通信学会 和文論文誌 D-I，投稿中，1998.

[NT98a] 中村 正樹，外山 芳人， 置換消去法による項書換え系の停止性判定について，電気関係学会北陸支部連合大会 講演論文集，p258，1998.

[NT98b] 中村 正樹，外山 芳人， 消去法による項書換え系の停止性判定について，信学技法，Vol.98，No.432，電子情報通信学会技術研究報告 COMP98-58，pp.57-64，1998.

[Ste95] J.Steinbach, Simplification Orderings - History of Results, *Fundamenta Informaticae* 24,pp.47-87, 1995.

[Zan94] H.Zantema, Termination of term rewriting: Interpretation and type elimination, *Journal of Symbolic Computation* 17,pp.23-50, 1994.

[Zan95] H.Zantema, Termination of term rewriting by semantic labelling, *Fundamenta Informatica* 24, pp.89-105, 1995.