数理解析研究所講究録
790 巻 1992 年 263-268

非線形項書き換えシステムのE－非オーバラップ性について

張 磊　　大山口 通夫+　　伊藤 英則

名古屋工業大学　　三重大学+

1．はじめに

これまで，項書き換えシステム（TRS）に関する研究は主に，線形或いは停止性を有する場合について行われてきたが，最近では，非線形かつ非停止の場合についても，新しい結果が幾つか報告されるようになってきた[3, 5, 6]．文献[3]では，E－非オーバラップの概念が提案され，E－非オーバラップ性を満たすTRSにおいては，各項の正規形は唯一であることが示された．

本研究では，このE－非オーバラップ性について考察し，ある制限されたTRSのクラスでは，“ω－非オーバラップ性質を持てば，E－非オーバラップ性質を持つ”という定理を証明することによって，E－非オーバラップ性を保証する，一つの判定可能な十分条件を与える．この結果の系として，非線形かつ非停止のTRSが合流性(Church-Rosser)を満たすための一つの判定可能な十分条件を示す．

2．諸定義

TRSに関する基本的な定義については文献[1-4]を参照されたい．非線形TRSとは，書換え規則の左辺に同じ変数の出現が二度以上に許されるTRSである．非線形TRSが制限的とは，右線形であり，かつ，規則の左辺に同じ変数がたかだか二度出現するときである．ここで，TRSが右線形とは，規則の右辺に出現する変数はたかだか一度出現するときである．

項書き換えシステムRの規則のペア$\alpha_i \to \beta_i$と$\alpha_j \to \beta_j$がE－オーバラップであるとは文脈C[]，非変数の項M及び代入θが存在して，$\alpha_i \equiv C[M]$，$\theta(\alpha_j) =_R \theta(M)$が成立するときである．但し，項のトップを書き換えることはない．

TRS　Rは<E, n>－オーバラップであるというのは，規則$\alpha_i \to \beta_i$，$\alpha_j \to \beta_j \in R$，文脈C[]，非変数の項Mと代入$\theta$が存在して，次の条件が成立するときである．（但し，$i \neq j$，又はC≠[]）$\alpha_i \equiv C[M]$，かつ，$\theta(\alpha_j) =_R \theta(M)$である．但し，項のトップ書き換えることはなく，書き換えの回数がn以下である．このようなペア<α_j, M>を<E, n>－オーバラップペアと呼ぶ．

項書き換えシステムRがE－非オーバラップシステムであるとは，自明な場合(つまり：$i = j \wedge C[] = \Box$)を除いて，任意の二つの規則がE－オーバラップでないときである．

TRS　Rの規則の組がω－オーバラップ性質を満たすとは無限項の代入を許して，オーバラップするときである．ω－非オーバラップシステムとはTRS　Rの任意の二つの規則はその自明のケースを除いて，ω－オーバラップではないシステムである．

本稿では，制限的な非線形TRSがE－非オーバラップ性を満たすための十分条件を与える．なお，TRSがE－非オーバラップ性を満たすかどうか判定する問題は，一般に非可解な問題であることが知られている([3])．

3．主定理

[定理1] TRS　Rは制限的であるとすると，そのとき，Rはω－非オーバラップ性質をもつならば，E－非オーバラップ性質を持つ．

この証明は背理法による．即ち，もし，ある<E, n>－オーバラップペアが存在するならば，ω－非オーバラップ性質を満たさないことを示すことによって，矛盾を導く．

矛盾を導くために，次節で導入するω－Unifyアルゴリズムを必要とする．そして，このω－Unifyアルゴリズムが次の二つの条件(1)と(2)を満たすことを示す．

(1)．ω－Unifyアルゴリズムはω－単一化アルゴリズムである．

(2)．ある<E, n>－オーバラップペアに対して，ω－Unifyアルゴリズムは成功する（即ち，ω－単一化に成功する）．

以上の(1)と(2)より，TRS　Rがω－非オーバラップ性質をもつならば，<E, n>－オーバラップペアの存在が否定される．ゆえに，定理1が成立する．

4．ω－Unifyアルゴリズム

このアルゴリズムに使われる定義と記法を簡単に述べる．

[定義] 変数と項の組からなる全集合をP(V, T)とする．但し，変数と項の集合をそれぞれVとTとする．

[記法] 集合$B \subseteq P(V, T)$に関するインサート操作は以下のように定義される．

$Insert(B, \{(x_1, P_1); \ldots; (x_n, P_n)\}) \equiv \{(y, \sigma(Q)) \mid (y, Q) \in B \cup \{(x_1, P_1); \ldots; (x_n, P_n)\}$

但し，写像$\sigma = [P_1/x_1, \ldots, P_n/x_n]$，$(x_i, P_i) \in P(V, T)$

[定義] 項M，Nの変数の出現の中の極小出現の集合$V_{M,N}$は以下

On the E-nonoverlapping Property of Nonlinear Term Rewriting Systems
Lei Zhang, Michio Oyamaguchi+ and Hidenori Itoh
Nagoya Institute of Technology
+ Mie University

のように定義される．$V_{mis}(M,N)\equiv min(O_v(M)\cup O_v(N))$
ここで，$O_v(M)=\{u|M/u\in V\}$ 即ち，変数出現の集合である．
［記法］操作Front(M,N)は項M，Nのフロンティアを取る．
フロンティアを二つに分けて，Aは変数のペアの集合であり，Sはその以外のペアの集合である．
$Front(M,N)\equiv\{(M/u,N/u)\mid u\in V_{mis}(M,N)\}$
$C(M,N)\equiv V\times V\cap Front(M,N)\subset A$ (Aは変数のペアの集合である)
$S(M,N)\equiv Front(M,N)-C(M,N)$
$\equiv\{(M_j,N_j)\mid M/u_j=M_j\in V\wedge N/u_j=N_j\in T;u_j\in V_{mis}(M,N)\}$

［定義］操作Check(M,N)は項M，Nの非変数部分の整合性をチェックする．つまり，以下のように定義される．
$M[u\leftarrow\Omega\mid u\in V_{mis}(M,N)]=N[u\leftarrow\Omega\mid u\in V_{mis}(M,N)]$であれば，
$Check(M,N)\equiv true$ である．
$M[u\leftarrow\Omega\mid u\in V_{mis}(M,N)]\neq N[u\leftarrow\Omega\mid u\in V_{mis}(M,N)]$であれば，
$Check(M,N)\equiv failure$ である．

［記法］操作New(x)はxに新しい変数名を割り当てる手続きである．ここで，新しい変数名とはその時点でプログラムの中で使われていない変数である．

［記法］操作 max(M,N) と min(M,N) は以下のである．
max(M,N) = M (N∈V のとき)
or = N (M∈V のとき)
min(M,N) = M (N∈T のとき)
or = N (M∈T のとき)

アルゴリズム ω－Unify (M, N)；
```
begin
(I)if Check(M,N)=FALSE then return(failure);
(II)Eq:=S(M,N);
    A:=C(M,N);
    B:=φ;
(III)while (∃(x,P),(x,Q)∈B) do
    begin
      B:={B-{(x,P),(x,Q)}}∪{(x,P[uj←max(M/uj,N/uj)|
          uj∈Vmis(P,Q)])}
      B:=Insert(B,Front(P,Q))
    end;
(IV)while (∃(x,P)∈B,(x,Q)∈Eq ) do
    begin
    if Check(P,Q)=FALSE then return(failure);
      Eq:=(Eq-{(x,Q)})∪S(P,Q);
      A:=A∪C(P,Q)
    end;
(V)if (∃(x,P),(x,Q)∈Eq;P≅Q) then
    begin
    if V(P)∪V(Q)=φ then return(failure);
    if Check(P,Q)=FALSE then return(failure);
      B:=Insert(B,{(x,P[uj←xj|uj∈Vmis(P,Q);New(xj),
                                  1≦j≦|Vmis(P,Q)|])});
      Eq:={Eq-{(x,P),(x,Q)}}
      ADD(xj,P/uj) ; (1≦j≦|Vmis(P,Q)|)
      A:=A∪{(xj,Q/uj)|1≦j≦|Vmis(P,Q)|}
      goto (III)
    end;
(VI)if (∃x⇢Ay,(x,P)∈B,(y,Q)∈Eq;x≠y) then
    begin
    if Check(P,Q)=FALSE then return(failure);
    if V(P)∪V(Q)=φ then
      begin
        B:=Insert(B,{y,P});
        Eq:=Eq-{y,Q}
      else begin
            B:=Insert(B,{(y,Q[uj←yj|uj∈Vmis(P,Q);
                    New(yj),1≦j≦|Vmis(P,Q)|])});
            Eq:=Eq-{(y,Q)}
            ADD(yj,Q/uj|1≦j≦|Vmis(P,Q)|)
          end
      goto (III)
    end;
(VII)if (∃x↔Az⇢Ay,(x,P)∈B,(y,Q)∈B,P≠Q) then
    begin
      if Check(P,Q)=FALSE then return(failure);
        A:=A-{(x,z)};
        B:=Insert(B,{(z,P[uj←zj|uj∈Vmis(P,Q);New(zj),
                                  1≦j≦|Vmis(P,Q)|])});
        ADD(zj,P/uj|1≦j≦|Vmis(P,Q)|)
    if (∃x↔y,(x,P)∈B,(y,Q)∈B,P≠Q) then
    begin
      if Check(P,Q)=FALSE then return(failure);
        A:=A-{(x,y)};
        ADD(min(P/uj,Q/uj),max(P/uj,Q/uj))
                        (1≦j≦|Vmis(P,Q)|)
        goto (III)
     end;
(VIII)if (∃x⇢Ay,(x,P)∈Eq,(y,Q)∈Eq,x≠y) then
    begin
      if V(P)∪V(Q)=φ∧P≠Q then return(failure);
      if V(P)∪V(Q)=φ∧P=Q then
        B:=Insert(B,{(x,P),(y,Q)});
        Eq:=Eq-{(x,P),(y,Q)};
      if V(P)∪V(Q)≠φ then
        begin
          if Check(P,Q)=FALSE then return(failure);
          B:=Insert(B,{(x,P[uj←aj|uj∈Vmis(P,Q);New(aj),
                                  1≦j≦|Vmis(P,Q)|]),
                      (y,Q[uj←bj|uj∈Vmis(P,Q);New(bj),
                                  1≦j≦|Vmis(P,Q)|])});
          Eq:=(Eq-{(x,P),(y,Q)});
          ADD(aj,P/uj|1≦j≦|Vmis(P,Q)|)
          ADD(bj,Q/uj|1≦j≦|Vmis(P,Q)|)
```

```
        end;
        goto (III)
    end
(IX)return(success)
end.
```

プロセスADDは以下である．

```
procedure ADD(var, term)
begin
  If term∈V then
    A:=A∪{(var, term)};
  else
    Eq:=Eq∪{(var, term)};
end.
```

ここで，このアルゴリズムはω－単一化アルゴリズムになっている．即ち，（M，N）がω－単一化が可能であれば，値としてsuccessを返し，そうでなければ，failure を返すということは以下の補題1より明かである．

また，これからのすべての補題は以下の前題によるものである．

［前提］：

項書き換えシステムRが<E，n>非オーバラップであり，そして，<E，n＋1>オーバラップである．アルゴリズムω－ｕｎｉｆｙの入力としては<E，n＋1>オーバラップペア$<\theta(\alpha_1), O'(M)>$である．

$(\alpha_1 \to \beta_1, \alpha_1 \to \beta_1 \in R \ \ \alpha_1/u = M \& \forall\ u \in O(\alpha_1) \wedge O(M))$

［補題1］：

ω－Ｕｎｉｆｙアルゴリズムはω－単一化アルゴリズムである．

<1>.<E，n＋1>－オーバラップペア$<\theta(\alpha_1), \theta(M)>$が単一化できれば，アルゴリズムω－Ｕｎｉｆｙの出力Ｅｑ(ｔ) ∪ Ａ(ｔ)∪Ｂ(ｔ)も単一化できる．

<2>.<1>の逆も成立する

<3>.アルゴリズムω－Ｕｎｉｆｙの出力Ｅｑ(t)∪Ａ(t)∪Ｂ(t)も単一化できる．

<4>.アルゴリズムω－Ｕｎｉｆｙは停止する．

［定義］：アルゴリズムω－Ｕｎｉｆｙのワン・ステップとは，このアルゴリズム(Ⅰ)，(Ⅱ)，(Ⅲ)，(Ⅳ)，(Ⅴ)，(Ⅵ)，(Ⅶ)，(Ⅷ) のどれか一回実行される事である．その中，ループになっている(Ⅲ)と(Ⅳ)に対して，ループ一回廻ることをワン・ステップであるとする．

アルゴリズムの中の集合Ｅｑ，Ａ，Ｂに対して，ｔステップ後の内容をそれぞれＥｑ(t)，Ａ(t)，Ｂ(t)と書く．

［証明］：

<1>の証明：

この証明はアルゴリズムω－Ｕｎｉｆｙのワン・ステップづつの出力がω－単一化アルゴリズムの解になることによる．

<E，n＋1>－オーバラップペア$<\theta(\alpha_1), \theta(M)>$が単一化できる前題によって，このアルゴリズムの全てのチェックの所で失敗しないである．（実際，失敗しないのは補題3に証明する．）以下，アルゴリズムω－Ｕｎｉｆｙのステップ数ｔに関する帰納法で証明する．

①．ｔ＝2の時，<E，n＋1>－ｵｰﾊﾞｰﾗｯﾌﾟﾍﾟｱ$<O(\alpha_1), O(M)>$のω－単一化の解が集合ＳとＣによって，保存されている．

②．ｽﾃｯﾌﾟ数くｔの時，ω－単一化アルゴリズムの解がＥｑ(ｔ-1)∪Ａ(ｔ-1)∪Ｂ(ｔ-1)に保存されている事を仮定する．

ｔｽﾃｯﾌﾟのとき考える．

(Ⅲ)から(Ⅷ)まで，同じく，入力のフロンティア部分に対して，操作を行って，入力のフロンティア部分がＢ(t)，Ａ(t)，Ｅｑ(t)に移した．帰納の仮定によって，このアルゴリズムが停止するとき，解がＥｑ∪Ａ∪Ｂに保存される．

つまり：Ｅｑ∪Ａ∪Ｂはω－単一化出来る．

<2>の証明：

<1>により，分かったのはこのアルゴリズムの操作はただフロンティア部分で行う．新しく導入されるのは変数だけである．このフロンティア部分からなる集合のω－単一化，いつも成功するので，それらのフロンティア部分の親の方も成功しないといけない．
従って，入力も単一化できる．

<3>の証明：

ここで，背理法を用いるため，アルゴリズムω－Ｕｎｉｆｙの出力集合Ｅｑ(ｔ)∪Ａ(ｔ)∪Ｂ(ｔ)は単一化不可能である事を仮定する．この場合，∃（ｘ，S_1），（ｘ，S_2）∈Ｅｑ∪Ａ∪Ｂ，出現ｕ∈Ｏ(S_1)∪Ｏ(S_2)，root(S_1/u)≠root(S_2/u)．もし，S_1∈ＶまたはS_2∈Ｖなら，（ｘ，S_1）と（ｘ，S_2）が単一化が出来てしまうので，（ｘ，S_1）と（ｘ，S_2）∈Ｅｑ∪Ｂだけ討論すればよい．

①．（ｘ，S_1）∈Ｂ，（ｘ，S_2）∈Ｂ

こういう対を存在する限り，アルゴリズムω－ｕｎｉｆｙの(Ⅲ)が動き続ける，つまり：このケースが有り得ない．

②．（ｘ，S_1）∈Ｂ，（ｘ，S_2）∈Ｅｑ

こういう対を存在するなら，アルゴリズムω－ｕｎｉｆｙの(Ⅵ)でのchockの所で失敗してしまい，failuroを返す．しかし，補題3によって，これがあり得ない．

③．（ｘ，S_1）∈Ｅｑ，（ｘ，S_2）∈Ｅｑ

こういう対を存在するなら，アルゴリズムω－ｕｎｉｆｙの(Ⅴ)でのchockの所で失敗してしまい，failuroを返す．しかし，補題3によって，これがあり得ない．
従って，最終形Ｅｑ∪Ａ∪Ｂが単一化可能である．

<4>の証明：

任意の時刻のサイズＬ(ｔ)＝Σ|θ(ｘ)|＋|O(Ｐ)|
（(ｘ，Ｐ)∈Ｅｑ∪Ａ）に関する帰納法で停止性を示そう．

①．ｔ≦2の時，入力$\theta(\alpha_1)$とθ(Ｍ)が(Ⅱ)でＥｑとＡに分けられる．当然に：

$|\theta(\alpha_1)|+|\theta(M)| > |Eq(0)|+|A(0)|$

入力のサイズが減った．

②．ｔ＝ｉの時，$L(t_{i-1}) > L(t_i)$であり，つまり：

$|Eq(t_{i-1})|+|A(t_{i-1})| > |Eq(t_i)|+|A(t_i)|$である事を仮定する．

$t = t_{i+1}$の時，(Ⅲ)から(Ⅷ)まで順番に述べる．

(Ⅲ). このループの全ての操作がEqとAと関係がないから，このループが無限に続く事がないのを示せばよい．
この証明はB(t)の左辺のPとQの深さD(P)とD(Q)に関する帰納法で行う：
(x, P), (x, Q)∈B(t)に対して，
max(D(B(t)))=n ⇒ (Ⅲ)が停止することを示す．
max(D(B(t)))=max{θ(D(x)), θ(D(P))|(x,P)∈B} かつ，xを左としてもつBの要素が二個以上である．
1>. max(D(B(t)))=1の時，つまり：∃(x,P),(x,Q)∈B,
PとQはそれぞれ
$f(x_1, x_2, \ldots, x_q)$と$f(y_1, y_2, \ldots, y_q)$ $(x_k \in V, y_k \in V, k=1, 2, \ldots)$
の形である．従って：
Front(P,Q)={$(x_1, y_1), \ldots, (x_q, y_q)$}
つまり，PとQの深さが一つ減った．
2>. max(D(B(t)))<nの時，PとQの深さが減っていたと仮定する．
max(D(B(t)))=nの時を考えよう．
フロンティアを取る事によって，上記のようなPとQが少なくとも一つ減る事が明かである．
なお，インサート操作により，Front(P,Q)のある要素が何回にコピーされても，深さの減ったものに限られている．帰納の仮定により，こんな項の深さも減っていたので，有限のD(P)とD(Q)に対して，いつかD(P)またはD(Q)=1になり，ループが抜ける．

(Ⅵ). $L(t_i) = |\theta(x)| + |\theta(Q)|$
$L(t_{i+1}) = |\mathrm{Front}(P,Q)| = |\{(P_k, Q_k) | k=1, 2, \ldots\}|$
$|\theta(P)| + |\theta(Q)| > |\mathrm{Front}(P,Q)|$
∵ $\theta(x) = \theta(P)$ (補題3による)
$|\theta(x)| + |\theta(Q)| > |\mathrm{Front}(P,Q)|$
つまり：$L(t_i) > L(t_{i+1})$

(Ⅴ). $L(t_i) = 2|\theta(x)| + |\theta(P)| + |\theta(Q)|$
$L(t_{i+1}) = \Sigma(2|\theta(x_j)| + |\theta(N_j)| + |\theta(M_j)|)$
$|\theta(P)| + |\theta(Q)| > |\mathrm{Front}(P,Q)|$
∵ $\theta(x_j) = \theta(x)/u_j$
$\{\theta(N_j) + \theta(M_j)\} \subset \{\theta(P) + \theta(Q)\}$
従って：$L(t_i) > L(t_{i+1})$

(Ⅵ). $L(t_i) = |\theta(y)| + |\theta(Q)|$
1>. V(P)∪V(Q)=φの時，
Eq:=Eq−{(y, Q)} ，従って：$L(t_i) > L(t_{i+1})$
2>. V(P)∪V(Q)≠φの時，
$L(t_{i+1}) = \Sigma|\theta(y_j)| + |\theta(Q/u_j)|$
$|\theta(P)| + |\theta(Q)| > |\mathrm{Front}(P,Q)|$
∵ $\theta(y_j) = \theta(y)/u_j$，従って：$L(t_i) > L(t_{i+1})$

(Ⅶ). (x, P)∈B, (y, Q)∈Bなので，この操作はBの内部で行うが，AとEqのサイズが増えない事を示そう．
1>. $x \overset{=}{\sim}_A z \overset{=}{\sim}_A y$の時：
A:=A−{(x, z)} により，
Aのサイズが$|\theta(x)| + |\theta(z)|$を減らした．
その後の操作により，増える分は$\Sigma(|\theta(z_j)| + |\theta(P/u_j)|)$
∵ $|\theta(x)| = |\theta(P)| > \Sigma|\theta(P/u_j)|$
Bに入れた操作により，$|\theta(z)| = |\theta(P)| > \Sigma|\theta(z_j)|$
従って：$L(t_i) > L(t_{i+1})$
2>. $x \overset{=}{\sim}_A y$の時：この場合はほぼ1>と同じから，省略する．

(Ⅷ)：$\exists x \overset{=}{\sim}_A y$, (x,P)∈Eq, (y,Q)∈Eq, x≠y
1>. V(P)∪V(Q)=φの時，
Eq:=Eq−{(x, P), (y, Q)}
従って：$L(t_i) > L(t_{i+1})$
2>. V(P)∪V(Q)≠φの時，
Eq:=Eq−{(x, P), (y, Q)} でサイズが減った．
その後の操作により，増える分は
$\Sigma(|\theta(Q/u_j)| + |\theta(P/u_j)| + |\theta(a_j)| + |\theta(b_j)|)$
Bに入れた操作により，t_{i+1}の時：
∵ $|\theta(x)| = |\theta(P)| > \Sigma|\theta(a_j)|$
$|\theta(y)| = |\theta(Q)| > \Sigma|\theta(b_j)|$
$|\theta(P)| > \Sigma|\theta(P/u_j)|$
$|\theta(Q)| > \Sigma|\theta(Q/u_j)|$
従って：$L(t_i) > L(t_{i+1})$
以上の審議を纏めて，t_{i+1}の時にも，$L(t_i) > L(t_{i+1})$が成立するから，このアルゴリズムが停止する．■

次に，3節の条件(2)を証明するために，ω−非ｵｰﾊﾞﾗｯﾌﾟの前提条件の元で，<E, n>ｵｰﾊﾞﾗｯﾌﾟペア<α_i, M>の具体例<$\theta(\alpha_i)$, θ(M)>のうち，nが最小であるようなペアをアルゴリズムω−Unifyに入力すると，ω−Unifyが成功することを示すことが出来る．条件(1)よりアルゴリズムω−unify中の整合性を検査する手続きcheckがいつも真であることを示せばよい．
その証明は以下いくつの補題によって与える。

[補題2]：
任意の部分リデックス・ペア:<$\theta(\alpha)$, θ'(M)>に対して，
$\theta(\alpha) \overset{=}{\sim} \theta'$(M)ならば，その系列における任意のリデックス出現uに対して，$\exists v \in V_{at}(\alpha, M)$，u≧vである事を成立する．
（リダクションがα，Mの共通部分のリデックスの集合に含まれていないである．）

[定義]：
リデックス・ペア<$\theta(\alpha)$, θ'(M)>とは，αは書き換え規則左辺であり，Mは書き換え規則の左辺或いはその部分項のペアである．

[証明]：
この証明は項のリダクションのリデックスがα, Mの共通部分リデックスの集合に含まれている事を仮定して，矛盾を導く．
<E, n+1>オーバラップの条件により：
$\alpha_i \to \beta_i, \alpha_j \to \beta_j \in R$；代入$\theta$, θ'；$\alpha_j/u = M \notin V$

$\varepsilon \in O_{\iota,v}$, $T[O(\alpha_\iota) =_R O'(M)]$;
$\#T[O(\alpha_\iota) =_R O'(M)] \leqq n+1$
リダクションのリデックスがα，Mの共通部分のリデックスの集合に含まれている仮定によって，
$\#T[O(\alpha_\iota)/v =_R O'(M)/v] = p \leqq n$であるようなvを選ぶ事ができる．$v \in (O(\alpha_\iota) - O_v(\alpha_\iota)) \wedge (O(M) - O_v(M))$ かつ $v' \in (O(\alpha_\iota) - O_v(\alpha_\iota)) \wedge (O(M) - O_v(M))$，$v > v'$．
この時，RがくE，n>非オーバラップである条件によって，
規則$\alpha_k \to \beta_k \in R$，代入$\sigma$，$\sigma'$；
$T[\theta(\alpha_\iota)/v =_R \theta'(\alpha_\iota)/v] \supseteq T'$
$T' \equiv T[\sigma(\beta_k) \leftarrow \sigma(\alpha_k)] \cdot T[\sigma(\alpha_k) =_R \sigma'(\alpha_k)] \cdot T[\sigma'(\alpha_k) \to \sigma'(\beta]$
（ここで，$\sigma(\beta_k) \leftarrow \sigma(\alpha_k)$は$O(\alpha) \overset{*}{\to} O'(M)$のリダクション序列の中の最も左のものである．）
$\#T[\sigma(\alpha_k) =_R \sigma'(\alpha_k)] \leqq n$とくE，n>非オーバラップ条件より，削除補題([1])を適用して，$\sigma =_R \sigma'$を導いた．
さらに，$\sigma(\beta_k) =_R \sigma'(\beta_k)$を得た．
$T'' \equiv T[\sigma(\beta_k) = \sigma'(\beta_k)]$とすれば，$\#T' > \#T''$であり，$T'$が$T''$で置き換える事とTRS Rが右線形であることにより$\#T[\theta(\alpha_\iota)/v =_R \theta'(\alpha_\iota)/v] < p$になり，これで矛盾を導いた．■

[補題3]：
tステップ後の集合A(t)，Eq(t)，B(t)に対して，以下の事を成り立つ．
<1>. $\forall(x, P) \in B(t)$に対して，$\theta(x) = \theta(P)$
<2>. $(x, P) \in B(t) \Rightarrow \forall(y, Q) \in B(t)$に対して，
$x \notin V(Q)$．
（即ち，B(t)のある要素の左辺に出現している変数は他の任意の要素の右辺に出現しない．）
<3>. $\forall(M, N) \in \Lambda(t)$，Eq(t)に対して，$M \overset{*}{\to} N$である．
<4>. t+1ステップにおいて，このアルゴリズムがfailureを返す事がない．

[証明]：
この証明はアルゴリズムω－unifyの実行ステップに関る帰納法を用いる．
①. t<3の時，$B = \phi$である，<1>,<2>が成立する．
t=1の時，補題2によって，<3>,<4>も成立する．
②. $\forall(x, P) \in B(t)$，A(t)，Eq(t)に対して，
<1>,<2>,<3>,<4>が成立するである事を仮定する．
t+1の時を討論する．
以下は(III)(IV)(V)(VI)(VII)(VIII)の順番に証明する．
(III)の場合：$(\exists(x,P),(x,Q) \in B(t))$
$B = \{B(t) - \{(x,P),(x,Q)\}\} \cup \{(x, P[u_j \leftarrow N_j] \mid P/u_j \equiv M_j, u_j \in Node(P) \wedge Node(Q)])\}$
$B(t+1) = Insert(B, Front(P,Q))$
$Front(P,Q) = \{(P/u, Q/u) \mid u \in O(P) \cap O(Q) \wedge ((P/u \in V) \vee (Q/u \in V))\}$
$(x, P) \in B(t)$，$(x, Q) \in B(t)$なので，帰納の仮定を用いて，
$\theta(x) = \theta(P)$と$\theta(x) = \theta(Q)$
及び$x \notin V(P)$と$x \notin V(Q)$であるから：
$\theta(P) = \theta(Q)$，だから，$\theta(P/u) = \theta(Q/u)$
つまり：$\theta(x)/u_j = O(x_j) = O(N_j)$
従って：
$O(x) = O(P[u_j \leftarrow N_j] \mid \ldots])$ つまり $O(x) = O(P)$
$x \notin V(P(t+1))$と$x \notin V(Q(t+1))$である事が分かった．
この(III)でΛとEqについての操作がなかった．
(IV)の場合：$(\exists(x,P) \in B(t), (x,Q) \in Eq(t))$
Bに関する操作がないから，B(t+1)が成立する．
帰納の仮定によって，Check(P,Q)=TURE と$x \overset{*}{\to} Q$である．
$Eq := (Eq - \{(x,Q)\}) \cup S(P,Q)$
$\Lambda := \Lambda \cup C(P,Q)$
補題2より，xとQの間のリダクションはFront(P,Q)により，保たれてる．従って，$\forall(M,N) \in Front(P,Q)$，$M \overset{*}{\to} N$
(V)の場合：$(\exists(x,P),(x,Q) \in Eq(t); P \simeq Q)$
$B(t+1) = Insert(B(t), \{(x, P[u_j \leftarrow x_j] \mid P/u_j, Q/u_j \equiv M_j \vee N_j, u_j \in V_{min}(P,Q); New(x_j), 1 \leqq j \leqq |V_{min}(P,Q)|])\}$
$\because \quad O(P[u_j \leftarrow x_j] \mid \ldots]) = P[u_j \leftarrow O(x_j) \mid \ldots])$
$O(Q[u_j \leftarrow x_j \mid \ldots]) = Q[u_j \leftarrow O(x_j) \mid \ldots])$
従って，$\theta(x) =_R P[u_j \leftarrow O(x_j) \mid \ldots])$を示せばよい．その為に
$\theta(x)$と$\theta(P)$，$\theta(Q)$の間のリダクション出現$u_j \in V_{min}(O(x), \theta(P)) \wedge V_{min}(\theta(x), \theta(Q))$を証明して，つまり：リダクションが$\theta(x)$と$\theta(Q)$の共通部分のリデックスの集合に含まれていない事である．
これを証明するためにTRSが制限的である条件を必要とすることを利用する．
帰納法で証明するために，リダクションがフロンティア出現以外で行う事を仮定する．即ち：
$v \in O(O(P)), u \in V_{min}(O(x), O(P)), s.t.\ v < u$．
規則$\alpha_k \to \beta_k \in R$
制限条件によって，xの個数が二個(right linear)であるから：
$O'(x) = O(x)[v \leftarrow \sigma(\beta_k)]$
$O'(x) = O(x)[v \leftarrow \sigma'(\beta_k)]$
項書き換えシステムRがくE，n+1>オーバラップであるから，
入力項$O(\alpha_j) =_R O(M)$のリダクション・ステップ数：
$\#T[O(\alpha_j) =_R O(M)] \leqq n+1$
従って：
$\theta'(\alpha_j) = \alpha_j[w \leftarrow O'(x) \mid \alpha_j/w = x]$
$\theta'(M) = M[w' \leftarrow \theta'(x) \mid M/w' = x]$ によって，
$\#T[\theta'(\alpha_j) =_R \theta'(M)] \leqq n$ になり，
これは矛盾であるから，リダクションがそれぞれのフロンティア出現以上で行う事が明かとなる：
$O(x) = O(P)$ と $O(x) = O(Q)$である．
そして，Pのフロンティアの変数を全部新しい変数に換えられたから，$x \notin V(P)$である事も明かとなる．
帰納の仮定によって，$x \overset{*}{\to} P$，$x \overset{*}{\to} Q$だから，補題2によって，
Check(P,Q)=TUREである．そして，(IV)で示したように
$(M,N) \in Front(P,Q)$，$M \overset{*}{\to} N$
(VI)の場合：$(\exists x \overset{*}{\to}_A y, (x,P) \in B(t), (y,Q) \in Eq(t); x \neq y)$
$V(P) \cup V(Q) = \phi$ならば，
$\theta(x) = \theta(P) = \theta(Q) = \theta(y)$である．
$V(P) \cup V(Q) \neq \phi$の時，
$B(t+1) = Insert(B(t), \{(y, Q[u_j \leftarrow y_j] \mid P/u_j, Q/u_j \equiv M_j \vee N_j,$

$u_j \in V_{mtn}(P,Q);New(y_j),1 \leq j \leq |V_{mtn}(P,Q)|]))$

リダクション系列：$\theta(Q) \cdot\!\!\rightarrow \theta(y) \cdot\!\!\rightarrow \theta(x) = \theta(P)$において，制限条件により，変数yの数が高々の二個出あるから，$\theta(y)$からのリダクションがあれば，一つ$\theta(Q)$へ向いて，もう一方$\theta(x)$つまり$\theta(P)$に向く．従って，これと(V)番ほぼ同じであるから，そのような方法で，

$\theta(y)/u_j = \theta(y_j) = \theta(x_j) = \theta(x)/u_j$が明かとなる．

従って，$\theta(y) = \theta(Q)$とy∈V(Q)である．

帰納の仮定によって，$x=P, y \cdot\!\!\rightarrow Q, x \cdot\!\!\rightarrow y$だから，補題2によって，

Check(P,Q)=TURBである．そして，(IV)で示したように

$(M,N) \in Front(P,Q)$，$M \cdot\!\!\rightarrow N$

(VII)．の場合：$(\exists x \leftrightarrow_A z \cdot\!\!\rightarrow_A y, (x,P) \in B(t), (y,Q) \in B(t), P \neq Q)$

$B(t+1) := Insert(B(t), \{(z, P[u_j \leftarrow z_j] | P/u_j, Q/u_j \equiv M_j \vee N_j,$

$u_j \in V_{mtn}(P,Q);New(z_j),1 \leq j \leq |V_{mtn}(P,Q)|]))$

帰納の仮定によって，tの時：

$\theta(x) = \theta(P)$，$\theta(y) = \theta(Q)$である．

つまり：$\theta(P) = \theta(x) \leftrightarrows \theta(z) \cdot\!\!\rightarrow \theta(y) = \theta(Q)$

制限条件により，変数zの数が高々の二個出あるから，$\theta(z)$からのリダクションがあれば，一つ$\theta(y)$つまり$\theta(Q)$へ向いて，もう一方$\theta(x)$つまり$\theta(P)$に向く．従って：

以下のようなvが存在する事を仮定する．

$v \in (O(\theta(P)) - V_{mtn}(\theta(x), \theta(P))) \wedge (O(\theta(Q)) - V_{mtn}(\theta(y), \theta(Q)))$；代入$\sigma, \sigma'$；規則$\alpha_k \rightarrow \beta_k$

$\theta(x) = \theta(z)[v \leftarrow \sigma(\beta_k)]$

$\theta'(x) = \theta(z)[v \leftarrow \sigma'(\beta_k)]$

TRS　Rが<E，n+1>オーバラップであるから，

入力項$\theta(\alpha_j) =_R \theta(M)$の$\#T[\theta(\alpha_j) =_R \theta(M)] \leq n+1$

$\theta'(\alpha_j) = \alpha_j[w \leftarrow \theta(x) | \alpha_j/w = z]$

$\theta'(M) = M[w' \leftarrow \theta'(x) | M/w' = z]$　によって，

$\#T[\theta'(\alpha_j) =_R \theta'(M)] \leq n$　になり，

これは矛盾であるから，リダクションがそれぞれのフロンティア出現以上で行う事が明かとなる：

$\theta(z) =_R \theta(P)$　である．

そして，Pのフロンティアの変数を全部新しい変数に換えられたから，zをV(P)である事も明かとなる．

帰納の仮定によって，$x=P, y=Q, x \leftrightarrow z \cdot\!\!\rightarrow y$だから，補題2によって，Check(P,Q)=TUREである．そして，(IV)で示したように

$(M,N) \in Front(P,Q)$，$M \cdot\!\!\rightarrow N$

(VIII)の場合：$(\exists x \cdot\!\!\rightarrow_A y, (x,P) \in Eq(t), (y,Q) \in Eq(t), x \neq y)$

$V(P) \cup V(Q) = \phi$の時，$P \leftrightarrows Q$ならば，

$\theta(x) \cdot\!\!\rightarrow \theta(P)$と$\theta(y) \cdot\!\!\rightarrow \theta(Q)$より，

$\theta(x) = \theta(P) = P$　と　$\theta(y) = \theta(Q) = Q$

が得られた．

$V(P) \cup V(Q) \neq \phi$の時，

$B(t+1) = Insert(B(t), \{(x, P[u_j \leftarrow a_j] | P/u_j, Q/u_j \equiv M_j \vee N_j$

$u_j \in V_{mtn}(P,Q);New(a_j),1 \leq j \leq |V_{mtn}(P,Q)|],$

$(y, Q[u_j \leftarrow b_j | P/u_j, Q/u_j = M_j \vee N_j, u_j \in V_{mtn}$

$(P,Q);New(b_j),1 \leq j \leq |V_{mtn}(P,Q)|]))$

帰納の仮定によって，tの時：

$\theta(x) \cdot\!\!\rightarrow \theta(P)$，$\theta(y) \cdot\!\!\rightarrow \theta(Q)$である．

つまり：$\theta(P) \cdot\!\!\rightarrow \theta(x) \cdot\!\!\rightarrow_U \theta(y) \cdot\!\!\rightarrow \theta(Q)$

制限条件により変数x，yの数が高々の二個出あるから，$\theta(y)$からのリダクションがあれば，一つ$\theta(Q)$へ向いて，もう一方$\theta(x)$つまり$\theta(P)$に向く．$\theta(x)$からのリダクションがあれば，一つ$\theta(P)$へ向いて，もう一方$\theta(y)$つまり$\theta(Q)$に向く．従って，これと(V)番ほぼ同じであるから，そのような方法を二回適用すると，以下の事になる：

$\theta(a_j) = \theta(x/u_j)$と$\theta(b_j) = \theta(y/u_j)$により，

$\theta(x) = \theta(P)$と$\theta(y) = \theta(Q)$とxをV(P)とyをV(Q)である事が分かった．

帰納の仮定によって，$x \cdot\!\!\rightarrow P, y \cdot\!\!\rightarrow Q, x \cdot\!\!\rightarrow y$だから，補題2によって，Check(P,Q)=TUREである．そして，(IV)で示したように

$(M,N) \in Front(P,Q)$，$M \cdot\!\!\rightarrow N$

これで，この補題の証明を終了する．■

5．唯一正規性と合流性

TRSがω-非ｵｰﾊﾞﾗｯﾌﾟであるかどうか判定する問題は可解であるから，定理1より，E-非ｵｰﾊﾞﾗｯﾌﾟ性を保証する判定可能な一つの十分条件を与えることが出来た．この十分条件は文献[3]の結果より，唯一正規性を保証するための十分条件でもある．

また，本研究で与えた定理より，非線形かつ非停止のTRSの合流性を保証する判定可能な十分条件を与えることが出来る．即ち，合流性に関する次の結果が知られている．

[定理2]（[5]，[6]）

TRSがE-非ｵｰﾊﾞﾗｯﾌﾟかつ右定数項システムであるならば，合流性を持つ．

従って，この定理2と本論文の定理1より，次の系が得られる．

[系] TRSがω-非ｵｰﾊﾞﾗｯﾌﾟシステムかつ制限的な右定数項システムであるならば，合流性を持つ．

参考文献

[1] 坂井公：Knuth-Bendixの完備化手続きとその応用，コンピューターソフトウェア，(pp 2-22) 1987,1.

[2] 二木厚吉，外山芳人：項書き換え型計算モデルとその応用，情報処理，Vol.24,No.2,pp.133-146,1983.

[3] Mizuhito OGAWA and Satoshi ONO: On the Uniquely Converging Property of Nonlinear Term Rewriting System, ソフトウェア基礎論　1989,5.(pp 61-70)

[4] Huet, G.: Confluent reductions: abstract properties and applications to term rewriting systems, J. ACM 27-4, pp. 797-821 (Oct. 1980).

[5] Oyamaguchi,M. : Some results on decision problems for right-ground term-rewriting systems, Proc. of Toyohashi Symposium on Theoretical Computer Science 1990.(pp 41-42)

[6] Oyamaguchi,M. : On the confluence problem for right-ground term-rewriting systems,unpublished manuscript.

[7] 張　磊　：項書き換えシステムに関する研究，三重大学修士論文，１９９０．

[8] 張磊 大山口通夫 伊藤英則　：非線形項書き換えシステムの唯一正規形の性質，第４３回情報処理学会全国大会講演論文集（３），pp.75-76.