社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

# 等式付き書換え系の等式数を削減する変換

三浦 浩一† 西田 直樹†† 酒井 正彦†† 坂部 俊樹†† 草刈 圭一朗††

†,†† 名古屋大学大学院情報科学研究科
〒 464-8603 名古屋市千種区不老町
E-mail: †miura@trs.cm.is.nagoya-u.ac.jp, ††{nishida,sakai,sakabe,kusakari}@is.nagoya-u.ac.jp

**あらまし** 与えられた等式集合を法として計算する等式付き書換え計算では，等式集合の語問題が決定不能であるため，一般には，与えられた項から 1 ステップで到達可能なすべての項を決定することができない．本稿では，与えられた等式付き書換え系を等価変換する手続きを与える．提案する手続きでは，等式の一部を規則化し，さらにそれらと重なる規則に対してそれらの等式を法として等価な左辺を持つ規則を追加する．元の等式集合から等式数を削減し，語問題が決定可能となる等式を残すことができれば，等式付き書換え系における 1 ステップの計算で到達可能なすべての項が決定可能となる．本変換は，等式付き書換え系の計算系列において各等式の適用順序が可換であるシステムを対象とする．また，本手法をプロセス計算の一つであるアンビエント計算に適用し，2 つの単純な構造等価関係を規則化に成功した例を示す．

**キーワード** 項書換え系，E 書換え計算，アンビエント計算，構造等価性

# Transformation of Equational Rewriting Systems for Removing some Equations

Koichi MIURA†, Naoki NISHIDA††, Masahiko SAKAI††,
Toshiki SAKABE††, and Keiichiro KUSAKARI††

†,†† Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-8603 Japan
E-mail: †miura@trs.cm.is.nagoya-u.ac.jp, ††{nishida,sakai,sakabe,kusakari}@is.nagoya-u.ac.jp

**Abstract** In equational rewriting, which is rewriting modulo equations, the set of all direct successors from a given term is in generally undecidable because the word problem for equations is undecidable. In this paper, we give an equivalent transformation of equational rewriting systems. The procedure converts some of the given equations to corresponding rewrite rules, and simultaneously adds rewrite rules whose left-hand sides are equivalent modulo equations with the left-hand sides of some of the given rules. Succeeded to reduce the number of equation so that the word problem for the remaining equation is decidable, the set of all direct successors of a given term becomes decidable. The procedure requires that input equations are commutative to the other equations. We also give a successful example of the procedure that is for ambient calculus.

**Key words** term rewriting system, rewriting modulo equations, ambient calculus, structural congruence

## 1. はじめに

等式付き書換え系は，与えられた等式を公理として書換え計算を進める計算モデルであり，様々な分野で用いられている．その計算には，対象とする項が与えられた等式集合 $E$ のもとで等価な関係である項の集合を決定する必要があるが，一般の等式集合に対してはそうした判定を有限時間で効率的に行うことはできない．そのため，計算機上で取り扱うことができない．一方で，等式集合を結合律や交換律などの特定の等式に限定した場合には，それら等式を法とした書換え計算で到達できる項の集合が決定可能であることはよく知られている．

プロセス計算では一般に，プロセスの遷移関係と構造等価関係を与え，プロセスの動作を決める．書換え型計算のモデルである項書換え系の分野では，様々な研究成果がある．これを利

用してプロセスの検証を行うには，プロセスを項で表現し，遷移関係と構造等価関係を書換え系でモデル化する必要がある．それらの関係の意味から前者を書換え規則で，後者を等式集合で与えることで定まる等式付き書換え系は自然なモデル化である．しかし，等式集合の語問題が決定不能であるため，検証を行うには不便である．一方，等式集合を両向き規則とみなすことで項書換え系がモデル化できるが，停止性をむやみに破壊する．実際には構造等価関係は，結合律，交換律，束縛変数の名前替え，$0+x=x$ のような単純な関係しかない．結合律と交換律による語問題は決定可能であるので，それ以外の等式を規則に吸収させることで，検証に適したモデル化が期待できる．

本稿では，等式付き書換え系の等式数を削減する等価変換手続きを与える．幾つかの等式に方向付けをし，代わりに元の等式と等価な書換え関係を持つ書換え規則を追加する提案する手続きでは，等式の一部を規則化し，さらにそれらと重なる規則に対してそれらの等式を法として等価な左辺を持つ規則を追加する．この変換を用いて等式数を削減し、等式集合の語問題が決定可能となる等式のみを持つ等式付書換え系に変換できれば，その 1 ステップの計算結果を決定することができる．特に，等式集合を結合律，交換律にすることができれば，AC 書換え計算などに関する多くの既存手法を用いることができる．本変換手続きは，与えられた等式付き書換え系の計算系列において等式の適用順序が可換であるシステムを対象とする．

また，本変換手続きをアンビエント計算へ適用した例を示す．アンビエント計算における構造等価関係のうち，2 つの構造等価関係に対して一方向に方向付け，等式付き書換え関係の等価性を保持する遷移規則を追加する．

## 2. 準　　備

本稿は項書換え系の一般的な記法に従う [1,2]．

関数記号の集合を $\mathcal{F}$，変数の可算無限集合を $\mathcal{X}$ とする．関数記号 $f$ の引数の個数を $arity(f)$ で表す．$\mathcal{F}$ と $\mathcal{X}$ から生成されるすべての項の集合を $\mathcal{T}(\mathcal{F},\mathcal{X})$ と記す．また，2 つの項 $s$ と $t$ が等しいことを $s \equiv t$ と記し，項 $t$ に含まれる変数の集合を $\mathcal{V}ar(t)$ と表す．

項 $t$ における位置の集合を $\mathcal{O}(t)$ と記す．位置 $\pi,\tau \in \mathcal{O}(t)$ に対して，$\pi\pi' = \tau$ を満たす $\pi' \in \mathcal{O}(t)$ が存在するとき，$\pi \leqq \tau$ と書く．特に $\pi' \neq \varepsilon$ のとき，$\pi < \tau$ と記す．$\mathtt{top}(t)$ は項 $t$ の先頭 (位置 $\varepsilon$) の記号を表す．

文脈 $C[\,]$ の位置 $p$ に出現するホール $\Box$ を項 $t$ で置き換えることによって得られる項を $C[t]_p$ と記す．なお $p$ を省略してもよい．項 $u$ が $t$ の部分項であるとき，$t \trianglerighteq u$ と記す．特に $\varepsilon < p$ であるとき，$u$ は $t$ の真部分項であるといい，$t \triangleright u$ と記す．また，位置 $\pi \in \mathcal{O}(t)$ における部分項 $u$ を $u \equiv t|_\pi$ と記す．

代入 $\sigma$ の定義域，値域は，それぞれ $\mathcal{D}om(\sigma)$, $\mathcal{R}an(\sigma)$ で表す．値域に現れる変数の集合を $\mathcal{VR}an(\sigma)$ で表す．$\mathcal{D}om(\sigma) = \{x_1,\cdots,x_n\}$ であり，かつ $\sigma(x_i) = u_i$ のとき，$\sigma$ を $\{x_1 \mapsto u_1,\cdots,x_n \mapsto u_n\}$ と記す．代入 $\sigma$ の項 $t$ への適用 $\sigma(t)$ を $t\sigma$ と記す．代入 $\sigma,\theta$ の合成 $\sigma\theta$ は $t\sigma\theta = \theta(\sigma(t))$ と定義される．$\sigma$ の定義域を $X \subseteq \mathcal{X}$ に制限した代入 $\sigma|_X$ は $\{x \mapsto x\sigma \mid x \in \mathcal{D}om(\sigma) \cap X\}$ である．代入 $\sigma,\theta$ に対して $\sigma\delta = \theta$ となる代入 $\delta$ が存在するとき，$\sigma \lesssim \theta$ と記す．項 $t\sigma$ を $t$ の具体項と呼ぶ．項 $s,t$ が互いの具体項であるとき，$s$ と $t$ は名前替えであるという．すべての $x \in \mathcal{D}om(\sigma)$ について $\sigma(x) \in \mathcal{T}(\mathcal{F})$ のとき，$\sigma$ を基礎代入と呼ぶ．項 $s$ と $t$ の単一化子とは，$s\sigma \equiv t\sigma$ を満たす代入 $\sigma$ である．またこのとき，$s$ と $t$ は単一化可能であるという．$s$ と $t$ の単一化子を $\sigma = uni(s,t)$ と記す．項 $s$ と $t$ の最汎単一化子は，$s$ と $t$ の単一化子 $\sigma$ であり，$s$ と $t$ の任意の単一化子 $\theta$ に対して $\sigma \lesssim \theta$ を満たすものと定義する．最汎単一化子は名前替えを法として唯一である．

書換え規則とは，$l \notin \mathcal{X}$ を満たす項 $l,r$ の対 $(l,r)$ であり，$l \to r$ と記す．項 $l,r$ をそれぞれ左辺，右辺と呼ぶ．書換え規則は他の規則と区別できるラベル $\rho$ を用いて，$\rho : l \to r$ と書くこともある．書換え規則 $l \to r$，$l' \to r'$ は関数記号の位置 $\pi \in \mathcal{O}_\mathcal{F}(t)$ について $l|_\pi$ と $l'$ が単一化可能であるならば,（$\pi$ において）重なりを持つという．ただし，同一の規則については $\pi \neq \varepsilon$ に限る．

項書換え系 (TRS) は，全ての書換え規則 $l \to r$ において $\mathcal{V}ar(l) \supseteq \mathcal{V}ar(r)$ を満たす書換え規則の集合 $R$ である．とくに，$l \notin \mathcal{X}$，$\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$ のどちらかを満たさない TRS を拡張項書換え系（eTRS）という．$R$ の書換え関係を $\to_R$ で表す．

書換え関係 $\to_R$ は，以下のように定義される二項関係である．

$$\to_R = \{(C[l\sigma]_p, C[r\sigma]_p) \mid l \to r \in R, C \in \mathcal{T}(\mathcal{F} \cup \{\Box\}, \mathcal{X})\}$$

文脈から $R$ が明らかな場合は $\to_R$ を単に $\to$ と記す．

等式集合 $E$ に基づく関係 $\to_E$ は，以下のように定義される二項関係である．

$$\to_E = \{(C[u\sigma]_\pi, C[v\sigma]_\pi) \mid u = v \in E, C \in \mathcal{T}(\mathcal{F} \cup \{\Box\}, \mathcal{X})\}$$

また，等式の方向を明示して $\to_{\{u\to v\}}$ とも記す．$\to_E$ の対称閉包を $\approx_E$ と記し，$\approx_E$ の反射推移閉包を $\sim_E$ と記す．

**定義 2.1** (等式付き書換え [3]) $R$ を TRS，$E$ を等式集合とする．$s,t$ を項とする．$s \sim_E s' \to_R t' \sim_E t$ を満たす項 $s',t'$ が存在するならば，項 $s$ から項 $t$ へ等式付き書換え可能であるといい，$s \to_{R/E} t$ と記す． □

項 $t$ 中のどの変数も 1 回しか現れないとき，$t$ は線形であるという．書換え規則の左辺が線形であるとき，左線形であるという．TRS $R$ に対して，$R$ に含まれるすべての規則が左線形であるとき，$R$ は左線形であるという．

擬順序 $\succsim$ とは反射的かつ推移的な二項関係である．$s \succsim t$ ならば $\forall C[\,].\, C[s] \succsim C[t]$ であるとき擬順序 $\succsim$ が文脈に閉じているという．$s \succsim t$ ならば $\forall \sigma.\, s\sigma \succsim t\sigma$ であるとき，擬順序 $\succsim$ が代入に閉じているという．（狭義の）半順序（strict order）$\succ$ は推移的かつ非反射的な二項関係である．関係 $\succ$ の無限減少列が存在しないとき，$\succ$ は整礎であるという．文脈と代入に閉じた擬順序 $\succsim$ を擬簡約化順序という．代入に閉じた整礎な半順序 $\succ$ と擬簡約化順序 $\succsim$ が $\succsim \circ \succ \subseteq \succ$ または $\succ \circ \succsim \subseteq \succ$ を満たすとき，$\succ$ を簡約化順序といい，順序の対 $(\succsim,\succ)$ を簡約化対という．

## 3. 等式付き書換え系の等式を規則化する変換

本節では，等式付き書換え系から等式数を削減する等価変換手続きを与える．等式付き書換え計算は項 $s$ の書換えに等式集合 $E$ を法とした単一化を行って計算を進めるが，一般に $E$ 単一化は決定不能である．そのため，等式付き書換え計算の $1$ ステップの計算で到達可能な項の集合を決定できない．また等式 $u = v \in E$ を $u \to v$，$v \to u$ の $2$ つの規則として $R$ に追加したのでは，明らかに $R$ は停止性を持たなくなる．そこで，等式集合に含まれる等式数を削減した等式付き書換え系への等価変換を提案する．与えられた等式付き書換え系から等式数を削減し，残った等式集合が $E$ 単一化が決定可能である等式付き書換え系へと変換ができれば，$1$ ステップの計算で到達可能なすべての項が決定可能となる．提案手法は，等式集合の一部を規則化し，それらの等式を法として等しい左辺を持つ書換え規則を追加するものである．提案する等価変換手続きが完全であるためには等式の適用順序が可換であるという条件を満たす必要がある．

具体的に等式を規則化する例として，TRS $R = \{g(f(f(x)) \to h(x)\}$，と等式 $f(a) = a$ を考える．このとき，項 $g(f(a))$ は，$g(f(a)) \approx g(f(f(a))) \to_R h(a)$ と書換えられる．等式 $f(a) = a$ を用いずに，$g(f(a))$ から $h(a)$ まで書換え可能にするために，$g(f(a)) \to h(a)$ という書換え規則を追加する．この追加規則は等式 $f(a) = a$ の左辺 $f(a)$ と規則の左辺の部分項 $f(x)$ が単一化可能であることから導くことができる．さらに，項 $g(a)$ は $g(a) \approx g(f(a)) \to_{R \cup \{g(f(a)) \to h(a)\}} h(a)$ と書換えられるので，同様に $g(b) \to h(a)$ という書換え規則を追加する．以上によって得られた TRS $R' = R \cup$ $\{g(f(a)) \to h(a), g(a) \to h(a)\}$ $\cup \{f(a) \to a\}$ は，元の等式付き書換え系と等価な計算を行うことができる．このように，項 $s$ から項 $t$ への書換え系列 $s \approx_{u=v} s' \to_{l \to r} t$ に対して $s \to_{\{l' \to r'\}} t$ に対応する規則を繰り返し追加する．

等式付き書換え系の等価変換手続きを図 $1$ に示す．手続きの概要を述べる．入力として TRS $R$，等式 $u = v$（ただし $u \succ v$，$\succ$ は簡約化順序）とする．手順 $2.1$ から $2.5$ の実行により書換え規則と等式を $1$ 回適用する書換え関係 $s \leftarrow_{u \to v} s' \to t$ に対し，$s \to_\rho t$ となる規則 $\rho_1$，$\rho_2$ を新たに追加する．集合 $R_i$ は等式を $i$ 回以下適用する書換え関係 $s \overset{\leqq i}{\leftarrow}_{u \to v} s' \to_{l \to r} t$ に対応する規則の集合 $\{l' \to r' \mid s \to_{l' \to r'} t\}$ である．集合 $S_i$ は，等式 $u = v$ との重なりを調べる候補の規則であり，手順 $2$ の $i$ 回目の操作で追加された規則の集合である．$S_i = \emptyset$，すなわち新たな規則が追加されなくなったら手続きを終了する．出力 $R_k$ の $k$ は手順 $2$ を繰り返した回数であり，$R_k$ は等式を $k$ 回以下適用する書換え関係に対応する規則の集合 $\{l' \to r' \mid s \overset{\leqq k}{\leftarrow}_{u \to v} s' \to_{l \to r} t,\ s \to_{l' \to r'} t\}$ である．

手順 $2$ における手続きについて述べる．まず，$S_i$ の中から一つ書換え規則 $l \to r$ を選択する（手順 $2.1$）．$R_{i+1}$，$S_{i+1}$ をそれぞれ初期化する（手順 $2.2$）．規則の左辺の部分項 $l|_\pi$（$\pi \geqq \varepsilon$）と等式の左辺 $u$ が単一化可能な場合（図 $2$），$v\sigma \approx u\sigma \equiv l|_\pi\sigma$

**入力** 規則集合 $R$，等式 $u = v$（ただし $u \succ v$）
**出力** 規則集合 $R_k$

1 $R_0 := R;\ S_0 := R;\ i := 0;$
2 $S_i = \emptyset$ になるまで以下を繰り返す:
　2.1 $l \to r \in S_i$ とする．
　2.2 $R_{i+1} := R_i;\ S_{i+1} := \emptyset$
　2.3 全ての $\sigma \in uni(l|_\pi, u)$（$\pi \geqq \varepsilon$）に対して
　　規則 $\rho_1 : l[v]_\pi\sigma \to r\sigma$ が $\rho_1 \notin \bigcup_{j=1}^{i} R_j$ ならば
　　$R_{i+1} := R_{i+1} \cup \{\rho_1\}$
　　$S_{i+1} := S_{i+1} \cup \{\rho_1\}$
　2.4 全ての $\sigma \in uni(l, u|_\pi)$（$\pi > \varepsilon$）に対して
　　規則 $\rho_2 : v\sigma \to u[r]_\pi\sigma$ が $\rho_2 \notin \bigcup_{j=1}^{i} R_j$ ならば
　　$R_{i+1} := R_{i+1} \cup \{\rho_2\}$
　　$S_{i+1} := S_{i+1} \cup \{\rho_2\}$
　2.5 $S_i := S_i \backslash \{l \to r\}$
3 $S_{i+1} = \emptyset$ ならば $R_k := R_i \cup \{u \to v\}$ として終了．そうでなければ $i := i + 1$ として手順 $2$ へ戻る．

図 1　等式を規則化する変換手続き

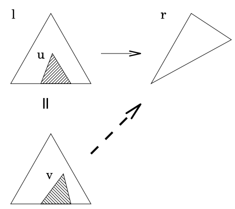


図 2　$uni(l|_\pi, u)$ の場合

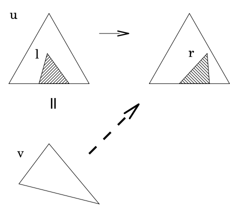


図 3　$uni(l, u|_{\pi'})$ の場合

であるので，$l[v]_\pi\sigma \leftarrow_{u \to v} l[u]_\pi\sigma \equiv l\sigma \to_{l \to r} r\sigma$ となる．よって書換え規則 $\rho_1 : l[v]_\pi\sigma \to r\sigma$ を $R_{i+1}, S_{i+1}$ に追加する（手順 $2.3$）．同様に規則の左辺 $l$ と等式の左辺の部分項 $u_\pi$（$\pi > \varepsilon$）が単一化可能な場合（図 $3$），$u|_\pi\sigma \equiv l\sigma$ であるので，$v\sigma \leftarrow_{u \to v} u\sigma \equiv u[l]_\pi\sigma \to_{l \to r} u[r]_\pi\sigma$ となる．よって書換え規則 $\rho_2 : v\sigma \to u[r]_\pi\sigma$ を $R_{i+1}, S_{i+1}$ に追加する（手順 $2.4$）．書換え規則 $l \to r$ は等式との重なりを調べる候補の集合 $S_i$ から除く（手順 $2.5$）．$S_i$ の全ての書換え規則について調べ終えたら，手順 $3$ へ進む．新たに追加された規則が無ければ出力を $R_k = R_i \cup \{u \to r\}$ として終了．そうでなければ手順 $2$ へ戻り，$S_{i+1}$ の規則と等式のとの重なりを調べる．

本手法は，等式集合 $E$ の全て等式を規則に変換するためのものではなく，特定の等式一つを規則化するものである．複数個の等式を規則化したい場合，手続きで規則が追加された TRS を入力して下記手続きを繰り返す必要がある．

**例 3.1**　TRS $R = \{g(f(f(x)) \to h(x)\}$，と等式 $f(a) = a$ に対して，上述の手続きにより等式を規則化する．

$R_0 := \{g(f(f(x)) \to h(x)\}$，$S_0 := \{g(f(f(x)) \to h(x)\}$ とする．

$R_1 := R_0$，$S_1 := \emptyset$ である．書換え規則 $g(f(f(x)) \to h(x) \in S_0$ を選択する．規則の左辺の部分項 $l|_{11}$ と等式の左辺 $u$ で単一化を行い，

$$\sigma = uni(l|_{11}, u) == uni(f(x), f(a))\{x \mapsto a\}$$

より

$$\begin{aligned} l[v]_\pi\sigma &\equiv g(f(a))\sigma &\equiv g(f(a)) \\ r\sigma &\equiv h(x)\sigma &\equiv h(a) \end{aligned}$$

となるので，$\rho_1 : g(f(a)) \to h(a)$ が $R_1$ に追加される．$R_1 := \{g(f(f(x))) \to h(x), g(f(a)) \to h(a)\}$．また，$S_1 := \{g(f(a)) \to h(a)\}$ となる．$S_0 = \emptyset$ となり手順 3 へ進むが，$S_1 \neq \emptyset$ であるため，手順 2 へ戻る．

書換え規則 $g(f(a)) \to h(a) \in S_1$ を選択する．規則の左辺の部分項 $l|_1$ と等式の左辺 $u$ で単一化を行い，

$$\sigma = uni(l|_1, u) == uni(f(a), f(a))\emptyset$$

より

$$\begin{aligned} l[v]_\pi\sigma &\equiv g(a)\sigma &\equiv g(a) \\ r\sigma &\equiv h(a)\sigma &\equiv h(a) \end{aligned}$$

となるので，$\rho_1 : g(a) \to h(a)$ が $R_2$ に追加される．$R_2 := \{g(f(f(x))) \to h(x), g(f(a)) \to h(a), g(a) \to h(a)\}$．また，$S_2 := \{g(a) \to h(a)\}$ となる．$S_1 = \emptyset$ となり手順 3 へ進むが，$S_2 \neq \emptyset$ であるため，手順 2 へ戻る．

書換え規則 $g(a) \to h(a) \in S_2$ を選択するが，規則の左辺と等式の左辺で単一化はできない．そのため規則は追加されず，$S_3 = \emptyset$．よって $R_2 = \{g(f(f(x))) \to h(x), g(f(a)) \to h(a), g(a) \to h(a)\} \cup \{f(a) \to a\}$ が得られる． □

本変換が健全であることは明らかである．よって完全であること，すなわち $\to_{R/(E\cup\{u=v\})} \subseteqq \sim_E \xrightarrow{*}_{u\to v} \to_{R_k\setminus\{u\to v\}} \sim_{u=v}$ であることを示す．

まず補題 3.2，3.3 によって手続きによって生成される規則の正しさを示す．

**補題 3.2** $s, t$ を項とし，$l \to r \in S_i$，$\rho_1 : l[v]_\pi\sigma \to r\sigma \in S_{i+1}$ とする．

$s \to^{\pi'}_{\{l\to r\}} t$ ならば，$s \to^{\tau}_{\{l\to r\}} s'$ かつ $\tau = \pi'\pi$ を満たす項 $s'$ に対して $s' \to^{\pi'}_{\rho_1} t$． □

**補題 3.3** $s, t$ を項とし，$l \to r \in S_i$，$\rho_2 : v\sigma \to u[r]_\pi\sigma \in S_{i+1}$ とする．

$s \to^{\pi'}_{\{l\to r\}} t$ ならば，$s \to^{\tau}_{\{l\to r\}} s'$ かつ $\pi' = \tau\pi$ を満たす項 $s'$ に対して $s' \to^{\tau}_{\rho_2} t$． □

手続き中の規則の作り方より，補題 3.2，3.3 が正しいことは明らかである．

以上の補題を用いて次の条件 C が成立するような $u = v$ と TRS $R$ に対して，本変換が完全であることが示せる．

**条件 C：** 等式 $u = v$，TRS $R$ に対して以下の 4 つの条件が成立する：

(i) $u, v$ はともに線形項である．

(ii) $\mathcal{V}ar(u) \subseteqq \mathcal{V}ar(v)$

(iii) $R$ は左線形 TRS である．

(iv) $v$ と $l$ は重なりを持たない．

**補題 3.4** $R_i$ を，$i$ 回手続きを繰り返して得られた TRS とする．等式 $u = v$ と $R_i$ は条件 C を満たすとする．

このとき任意の項 $s, t$ に対して，ある項 $s'$ が存在して $s \approx^{\tau}_{u=v} s' \to^{\pi}_{R_i} t$ ならば，ある項 $s''$ が存在して $s \to_{R_{i+1}} s'' \sim_{u\to v} t$ である． □

**系 3.5** 手続きを $i$ 回繰り返して得られた TRS $R_i$ とする．等式 $u = v$ と $R_i$ は条件 C を満たすとする．

このとき任意の項 $s, t$ に対して，ある項 $s'$ が存在して $s \sim^{\tau}_{u=v} s' \to^{\pi}_{R_i} t$ ならば，ある項 $s''$ が存在して $s \to_{R_k} s'' \sim_{u\to v} t$ である．

[証明] $s \overset{j\ \tau}{\approx}_{u=v} s' \to^{\pi}_{R_i} t$ と仮定し，$j$ に関する帰納法で証明する．

(I) $j = 1$ のとき $s \approx^{\tau}_{u=v} s' \to^{\pi}_{R_i} t$ であり， 補題 3.4 より，$s \to_{R_{i+1}} s'' \sim_{u\to v} t$．$R_i \subseteqq R_k$ より $s \to_{R_k} s'' \sim_{u\to v} t$．

(II) $j$ のとき成立を仮定する．このときある項 $s'''$ が存在し，$s \approx_{u=v} s''' \overset{j\ \tau}{\approx}_{u=v} s' \to^{\pi}_{R_i} t$ とする．帰納法の仮定より，$s \approx_{u=v} s''' \to_{R_{i+j}} s'' \sim_{u\to v} t$．さらに補題 3.4 より，$s \to_{R_{i+j+1}} s'' \sim_{u\to v}$ $R_{i+j+1} \subseteqq R_k$ より $s \to_{R_k} s'' \sim_{u\to v} t$．

□

**定理 3.6** $R$ を TRS，$E \cup \{u = v\}$ を等式集合とする．$R$ と $u = v$ は条件 C を満たすとする．本手続きに $R$ と $u = v$ を入力して得られた TRS を $R_k$ とする．

$\sim_{E\cup\{u=v\}} = \sim_E \circ \sim_{\{u=v\}}$ とする．

このとき任意の項 $s, t$ に対して，ある項 $s'$ が存在して $s \sim_{E\cup\{u=v\}} s' \to_R t$ ならば，ある $s'', t'$ が存在して $s \sim_E s'' \to_{R_k} t' \sim_{u\to v} t$．

[証明] $s \sim_{E\cup\{u=v\}} s' \to_R t$ と仮定する．$\sim_{E\cup\{u=v\}} = \sim_E \circ \sim_{\{u=v\}}$ より，$s \sim_E s'' \sim_{u\to v} s' \to_R t$．系 3.5 より，$s'' \to_{R_k} t' \sim_{u\to v} t$ である．ゆえに $s \sim_E s'' \to_{R_k} t' \sim_{u\to v} t$．

□

本手続きでは，入力，出力が eTRS となることを許す．例えば $R = \{f(x) \to r\}$（$r$ は任意の項）と等式 $g(f(x)) = x$ を入力すると，$\rho_2 : x \to g(r)$ という規則が追加され，eTRS となってしまう．しかし，本手続きの出力が eTRS ではなく，全ての規則が書換え規則であるならば，その出力結果は変換前の等式付き書換え系に対して等価な等式付き書換え計算を行う TRS である．eTRS が得られた場合は手続きは失敗とする．また，このような eTRS となる場合をを除くには，“$v \in \mathcal{X}$ ならば $u$ は $R_i$ の規則と重ならない” という条件の成立の可否を手続きの手順 2.1 を行う前に毎回判定すればよい．この条件が成立しないならば，手続きは失敗とする．先ほどの $g(f(x)) = x$ では $v \equiv x \in \mathcal{X}$ であるが，$u_1$ が規則の左辺 $f(x)$ と重なるため失敗となる．

また，本手続きは一般には停止性を持たないと考えられる．これは，単一化によって，$j$ 番目に追加された規則 $l' \to r' \in S_j$ が $i$ 番目（$i < j$）に追加された規則 $l \to r \in \bigcup_{k=0}^{i} S_k$ と一致した場合に，手順 2 の規則の追加をする操作が無限に行われてしまうためである．例えば $R = \{g(x, f(x)) \to r\}$（$r$ は任意の項），$E = \{f(f(x)) = x\}$ に対して手続きを行うと，$S_1 = \{g(f(x), x) \to r\sigma'\}$，$S_2 = \{g(x, f(x)) \to r\sigma''\}$ となり，手続きが無限回続くことになる．

## 4. アンビエント計算への適用

本節では，本手法をプロセス計算の一つであるアンビエント計算に適用した例を示す．

アンビエント計算 [4] は，アンビエントと呼ばれる名前付きのプロセス群のネットワーク上での移動を形式化する抽象計算モデルであり，モバイルエージェントや分散システムなどのモバイルソフトウェアをモデル化している．アンビエント $n[P]$ は，名前 $n$ で境界づけられた領域内でプロセス $P$ が動作していることを表す．アンビエントの主なアクションは，他のアンビエントの中に入る in，外に出る out，境界を取り払う open の 3 つである．アンビエント計算の操作意味論は，構造等価関係（図 4）とプロセスの遷移関係（図 5）で与えられる．$P, Q, R$ はプロセス変数を，$n, m$ は名前変数を表す．

$E$ の構造等価関係のうち，$P \mid 0 \equiv P$，$!0 \equiv 0$ の 2 つを本手法により削除し，$\to_{R/E}$ と等価な遷移をする $\to_{R''/E''}$ へと変換する．$E_0$ の左側にある 8 つの構造等価関係および $R_0$ の下側にある 4 つの遷移規則は，等式付き書換え系では自明な関係，規則であるため，考慮しない．

構造等価関係 $P' \mid 0 \equiv P'$ の除去する場合を示す．

遷移規則 $n[in\ m.P \mid Q] \mid m[R] \to m[n[P \mid Q] \mid R]$ に対して，本手続きを適用すると，$l|_{12} = in\ m.P \mid Q$ より

$$\sigma = uni(l|_{12}, u) = \{P' \mapsto in\ m.P, Q \mapsto 0\}$$

より

$$\begin{array}{lll} l[v]_\pi\sigma & \equiv (n[P'] \mid m[R])\sigma & \equiv (n[in\ m.P] \mid m[R]) \\ r\sigma & \equiv m[n[P \mid Q] \mid R] & \equiv m[n[P \mid 0] \mid R] \end{array}$$

となるので，規則：$(n[in\ m.P] \mid m[R]) \to m[n[P \mid 0] \mid R]$ が $S_1$ 追加される．

なお，この規則を追加したことで，

$$\begin{array}{lll} n[in\ m.P] \mid m[R] & \sim_{E_0} & n[in\ m.P|0] \mid m[R] \\ & \to_{R_0} & m[n[P \mid 0] \mid R] \end{array}$$

というプロセスの遷移を，$P \mid 0 \equiv P$ を用いること無く，$R'$ のみで遷移できる．

以下同様に，他の遷移規則についても手続きを実行すると，

$$\begin{array}{ll} S_1 = \{ & n[in\ m.P] \mid m[R] \to m[n[P] \mid R] \\ & m[n[out\ m.P] \mid R] \to n[P] \mid m[R] \\ & m[n[out\ m.P \mid Q]] \to n[P \mid Q] \mid m[0]\ \} \\ S_2 = \{ & m[n[out\ m.P]] \to n[P] \mid m[0]\ \} \\ S_3 = & \emptyset \end{array}$$

となり，図 6 の遷移規則が得られる．この $R'$ は，$P \mid 0 \equiv P$ とこれ以上単一化できないため，ここで終了する．

$!0 \equiv 0$ と $R'$ から得られる遷移規則は図 7 の $R''$ となる．

本手続きで除去できていない構造等価関係は以下の 7 つである．

$$\begin{array}{l} P \mid Q \equiv Q \mid P \\ (P \mid Q) \mid R \equiv P \mid (Q \mid R) \\ !P \equiv P \mid !P \\ (\nu n)(\nu m)P \equiv (\nu m)(\nu n)P \\ n \notin fn(P) \Rightarrow (\nu n)(P \mid Q) \equiv P \mid (\nu n)Q \\ n \neq m \Rightarrow (\nu n)m[P] \equiv m[(\nu n)P] \\ (\nu n)0 \equiv 0 \end{array}$$

$P \mid Q \equiv Q \mid P, (P \mid Q) \mid R \equiv P \mid (Q \mid R)$ は，それぞれ関数記号 $\mid$ に対する交換律，結合律である．そのためこの二つの構造等価関係は除去せずとも，AC 書換え計算についての既存の手法を用いて計算を決定的に進めることができる．

$!P \equiv P \mid !P$ は，等式の左辺が非線形であるため，定理 3.6 の条件を満たしていない．定理 3.6 の条件の緩和と，AC 単一化を用いた手続きを行うことで，規則化できると考えている．

一方，他の三つの構造等価関係については，$\alpha$ 合同性を考慮に入れて規則化しなければならないものである．$\alpha$ 合同性は $\pi$ 計算などでも用いられている性質であり，束縛変数の名前替えを適切に行って遷移を行うという性質である．この性質を吸収した遷移規則を追加する方法については今後の課題の一つである．

## 5. おわりに

本稿では，等式付き書換え系の等式数を削減する等価変換手続きを与えた．本手法は等式を規則化し，さらにそれらの等式を法として等価な左辺を持つ書換え規則を追加するものである．本手続きを用いて等式集合の等式数を削減し，$E$ 単一化が決定可能な等式集合をもつ等式付き書換え系へ等価変換することで，ある項からの 1 ステップの計算で到達可能な全ての項を決定可能とすることができた．本変換は，各等式の適用順序が可換であるシステムを対象した．また，プロセス計算の一つであるアンビエント計算に本手法を適用した例を示した．

本手続きは，$u \to v$ という等式を単純に方向付けた規則を出力の TRS に加えている．しかし，この方向の付いた等式も，他の書換え規則との重なりに基づいて，規則化できるできると考えられ，そのような手続きを考案することは今後の課題の一つである．また、変換の完全性を示す定理 3.6 が成立する条件を，条件 C とした．このため，今回の手法では $P \mid !P \equiv !P$ のような構造等価関係を規則化できなかった．変換手続きを改良し，これら条件を緩和することに努めたい．

また，アンビエント計算の例に関しては，$\alpha$ 合同性を利用する残りの構造等価関係を規則化する変換手続きを考案する必要がある．

**謝辞** 本研究は一部，科研費#16650005，#17700009，#18500011 ならびに名古屋大学 21 世紀 COE プログラム（社会情報基盤のための音声・映像の知的統合）の補助を受けている．

### 文　献

[1] F. Baader and T. Nipkow: "Term Rewriting and All That", Cambridge University Press (1998).

[2] E. Ohlebusch: "Advanced Topics in Term Rewriting", Springer (2001).

$$
\begin{array}{ll}
P \equiv P & P \mid Q \equiv Q \mid P \\
P \equiv Q \Rightarrow Q \equiv P & (P \mid Q) \mid R \equiv P \mid (Q \mid R) \\
P \equiv Q, Q \equiv R \Rightarrow P \equiv R & !P \equiv P \mid !P \\
 & (\nu n)(\nu m)P \equiv (\nu m)(\nu n)P \\
P \equiv Q \Rightarrow (\nu n)P \equiv (\nu n)Q & n \notin fn(P) \Rightarrow (\nu n)(P \mid Q) \equiv P \mid (\nu n)Q \\
P \equiv Q \Rightarrow P \mid R \equiv Q \mid R & n \neq m \Rightarrow (\nu n)m[P] \equiv m[(\nu n)P] \\
P \equiv Q \Rightarrow !P \equiv !Q & P \mid 0 \equiv P \\
P \equiv Q \Rightarrow n[P] \equiv n[Q] & (\nu n)0 \equiv 0 \\
P \equiv Q \Rightarrow M.P \equiv M.Q & !0 \equiv 0
\end{array}
$$

**図 4　アンビエント計算の構造等価関係** $E$

$$
\begin{array}{l}
n[in\ m.P \mid Q] \mid m[R] \rightarrow m[n[P \mid Q] \mid R] \\
m[n[out\ m.P \mid Q] \mid R] \rightarrow n[P \mid Q] \mid m[R] \\
open\ n.P \mid n[Q] \rightarrow P \mid Q \\
P \rightarrow Q \Rightarrow P \mid R \rightarrow Q \mid R \\
P \rightarrow Q \Rightarrow (\nu n)P \rightarrow (\nu n)Q \\
P \rightarrow Q \Rightarrow n[P] \rightarrow n[Q] \\
P' \equiv P, P \rightarrow Q, Q \equiv Q' \Rightarrow P' \rightarrow Q'
\end{array}
$$

**図 5　アンビエント計算の遷移関係** $R$

$$
\begin{array}{rl}
R' = R \cup \{ & n[in\ m.P] \mid m[R] \rightarrow m[n[P] \mid R] \\
 & m[n[out\ m.P] \mid R] \rightarrow n[P] \mid m[R] \\
 & m[n[out\ m.P \mid Q]] \rightarrow n[P \mid Q] \mid m[0] \\
 & m[n[out\ m.P]] \rightarrow n[P] \mid m[0]\ \} \\
\cup \{ & P \mid 0 \rightarrow P\ \}
\end{array}
$$

**図 6　$P \mid 0 \equiv P$ と等価な等式付き書換えを行う遷移関係** $R'$

$$
R'' = R' \cup \{ \quad !0 \rightarrow 0\ \}
$$

**図 7　$P \mid 0 \equiv P$，$!0 \equiv 0$ と等価な等式付き書換えを行う遷移関係** $R''$


[3] J. Giesl and D. Kapur: "Dependency pairs for equational rewriting", Proceedings of the 12th International Coference on Rewriting Techniques and Applications, Vol. 2051 of LNCS (2001).

[4] L. Cardelli and A. D. Gordon: "Mobile ambients", Foundations of Software Science and Computation Structures: First International Conference, FOSSACS '98, Springer-Verlag, Berlin Germany (1998).