アルゴリズム 66−8
(1999. 1. 27)

有界長文字列の word 問題について

渡邊敏宏[†] 梅村恭司[††]

有界長文字列の word 問題とは，長さに制限のある有限文字列空間において，任意の二つの文字列に対する等式が与えられた等式集合から導けるかどうかを決定する問題であり，長さに制限が無い場合とは異なり原理的に決定可能な問題である．word 問題を効率的に解くには，等式集合を完備化してそれに対して完全な書換え系を作り，二つの文字列の正規形を求めて比較すれば良い．しかし，等式の適用が長さで制限され得る有界長文字列では，通常のアルゴリズムで完備化された書換え系は元の等式集合に対して十分であるが一般には健全ではない．本論文では，健全性に注意して与えられた等式集合に対して完全な書換え系を生成する方法について述べる．

The Word Problem on Bounded Strings

TOSHIHIRO WATANABE[†] and KYOJI UMEMURA[††]

The word problem on bounded strings is theoretically decidable problem whether an equation for two strings can be derived from a given equation set on finite strings bounded in length. This problem can be efficiently solved by using completed rewriting system from the given equation set. However, in the case of bounded strings, the completed rewriting system in unbounded string is not generally sound for the given equation set. In this paper, we will show the way to generate a complete rewriting system for a given equation set on bounded strings.

1. はじめに

文字列の word 問題は，ある二つの文字列が与えられた等式集合による合同関係で等しいかどうかを決定する問題である．この問題の解法の一つに，与えられた等式集合を基にして一方の文字列の合同類を求めてその中にもう一方の文字列が含まれるかどうかを調べる方法がある．しかし，与えられる等式集合によってはある文字列の合同類が有限でないこともありえるため，このアルゴリズムは停止性が保証できない．そこで，アルゴリズムの停止性を保証するために，文字列を長さに上限がある有界長文字列に限定して探索空間である文字列の全体集合を有限にして問題を解くのが実際的である．我々はこの問題を有界長文字列の word 問題と呼ぶことにする．一般に word 問題は，等式集合をそのまま使って解くよりも，等式集合に対して完全な書換え系を使って解く方が効率的である．有界長文字列上で等式集合を書換え系に書き換える方法に，文字列書換え系の完備化アルゴリズムを使う方法があるが，長さの制限がないものとして完備化された書換え系は元の等式集合に対して十分であるが有界長の系では健全ではない．これは，完備化の際に等式集合の等式上で推論することで合同関係が長さの制限で変化することが原因であるそこで本論文では，書換え規則に膨らみと呼ばれるパラメータを追加することで健全な書換え系を生成する方法を提案し，実際に与えられた等式集合に対して完全な書換え系を生成する方法を示す．

2. 有界長文字列の word 問題に関する定義と記法

まず，文字列に関する定義から始める．Σ を有限アルファベットの集合とし，空文字列 e を含む Σ 上の全文字列の集合を Σ^* で表す．つまり，Σ^* は空文字列を単位元とし，文字列の連結を乗算とする Σ により生成される自由モノイドである．もし $s \in \Sigma^*$ なら，s の長さを $|s|$ で表し，次のように定義する．$|e| = 0$, $a \in \Sigma$ に対して $|a| = 1$, $s \in \Sigma^*$ と $a \in \Sigma$ に対して

† 豊橋技術科学大学情報工学専攻
Department of Information and Computer Sciences, Toyohasi University of Technology
†† 豊橋技術科学大学情報工学系
Department of Information and Computer Sciences, Toyohasi University of Technology

$|sa| = |s|+1$ である．また，0以上の整数は N で表す．

次に有界長文字列の word 問題に関する必要事項を定義する．

有界長文字列の定義　ある一定の長さ以下の文字列を有界長文字列と呼ぶ．長さの上限が $L \in N$ の全有界長文字列の集合を Σ_L^* で表し，次のように定義する．

$$\Sigma_L^* = \{x \in \Sigma^* \mid |x| \leq L\}$$

有界長文字列上の簡約関係の定義　Σ_L^* 上の反射的，反対称的，かつ推移的な関係を Σ_L^* 上の簡約関係と呼び，記号 $\rightarrow$ で表す．Σ_L^* 上の簡約関係は許容可能な関係である．すなわち，任意の $l, r \in \Sigma_L^*$ に対して $l \rightarrow r$ が成り立つならば，l, r の前後に共通の有界長文字列を連結してできる $s_l, s_r \in \Sigma_L^*$ の間にも $s_l \rightarrow s_r$ が成り立つとしても文字列の連結に矛盾しない．

有界長文字列上の同値関係と合同関係の定義　Σ_L^* 上の許容可能な同値関係を Σ_L^* 上の合同関係と呼び，記号 $\leftrightarrow$ で表す．

有界長文字列上の大小関係の定義　Σ_L^* 上の大小関係 $>$ を次のように定義する．$l, r \in \Sigma_L^*$ であるとすると，

(1)　$|l| > |r|$ なら $l > r$ である．

(2)　$|l| = |r|$ で，r が l より辞書式順序で後ろなら $r > l$ である．

この順序は任意の $l, r \in \Sigma_L^*$ に対して，$l > r, l = r, r > l$ のいずれかであるので全順序である．

等式集合の定義　任意の有界長文字列の非順序対の集合を等式集合と呼び，E_L で表す．すなわち，$E_L \subseteq \Sigma_L^* \times \Sigma_L^*$ である．$(l, r) \in E_L$ ならば (r, l) を E_L に加えても E_L の意味は不変である．等式集合は Thue システムとも呼ばれる．

等式集合による合同関係の定義　$(l, r) \in E_L$ ならば $l \leftrightarrow r$ であるとすれば，E_L を与えることで Σ_L^* 上の合同関係を決めることができる．このようにして決められる合同関係を等式集合 E_L による合同関係と呼び，$\leftrightarrow_{E_L}$ で表す．また，$\leftrightarrow_{E_L}$ の推移閉包は $\overset{*}{\leftrightarrow}_{E_L}$ で表す．$\overset{*}{\leftrightarrow}_{E_L}$ は合同関係となるのは明らかである．

等式集合による合同類の定義　任意の $s \in \Sigma_L^*$ に対する等式集合 E_L による合同類を $[s]_{E_L}$ のように表し，次のように定義する．

$$[s]_{E_L} = \{x \in \Sigma_L^* \mid x \overset{*}{\leftrightarrow}_{E_L} s\}$$

等式集合による変形の定義　任意の $l, r \in \Sigma_L^*$ に対して $l \overset{*}{\leftrightarrow}_{E_L} r$ が成り立つ時，l を r で (あるいは r を l で) 置き換えることを E_L による変形と呼ぶ．

有界長文字列の word 問題の定義　有界長文字列の word 問題とは，ある等式集合 E_L が与えられた時に，任意の $l, r \in \Sigma_L^*$ に対して $l \overset{*}{\leftrightarrow}_{E_L} r$ であるかどうかを決定する問題である．

3. 等式集合による解法

有界長文字列の word 問題は決定可能な問題である．ここでは，等式集合をそのまま使ってこの問題を解く方法を示す．

与えられる等式集合を E_L，合同判定をする二つの有界長文字列を l, r とする．このとき，E_L を使って有界長文字列の word 問題を解くアルゴリズムを以下に示す．

アルゴリズム　ある有界長文字列集合 $S \subseteq \Sigma_L^*$ を考える．

(1)　$S = \{l\}$ とする．

(2)　S の任意の要素 s に対して $s \leftrightarrow_{E_L} s'$ かつ $s' \notin S$ であるような $s' \in \Sigma_L^*$ を一つだけ見つけて (3) に進む．もし s' が一つも見つからなければ $l \overset{*}{\not\leftrightarrow}_{E_L} r$ である．

(3)　もし $s' = r$ ならば $l \overset{*}{\leftrightarrow}_{E_L} r$ である．そうでなければ S に s' を加えて (2) に戻る．

このアルゴリズムにおいて S は $[l]_{E_L}$ の部分集合である．したがって，$r \in S$ なら $l \overset{*}{\leftrightarrow}_{E_L} r$ であり，$S = [l]_{E_L}$ かつ $r \notin S$ なら $l \overset{*}{\not\leftrightarrow}_{E_L} r$ である．任意の $s \in \Sigma_L^*$ に対して $[s]_{E_L}$ は有限集合であり，S は単調に増加するのでこのアルゴリズムは必ず停止する．よって，有界長文字列の word 問題は決定可能な問題である．

ところで，一般的に word 問題を解くには等式集合をそのまま使って解くよりも，等式集合に対して完全な書換え系を使って解く方が効率的である．次の節以降でこの方法について述べる．

4. 書換え系に関する定義と記法

まず，書換え系に関する必要事項を定義する．なお，文字列書換え系に関する解説として文献 1) がある．

書換え系の定義　任意の有界長文字列の順序対 (書換え規則) の集合を書換え系と呼び，R_L で表す．すなわち，$R_L \subseteq \Sigma_L^* \times \Sigma_L^*$ である．順序対であるので $(l, r) \in R_L$ であっても $(r, l) \in R_L$ であるとは限らない．書換え系は準 Thue システムとも呼ばれる．

書換え系による簡約関係の定義　R_L をそのまま E_L

と同じに考えて $(l,r) \in R_L$ ならば $l \rightarrow r$ であるとすれば，R_L を与えることで Σ_L^* 上の簡約関係を決めることができる．このような簡約関係を書換え系 R_L による簡約関係と呼び，$\rightarrow_{R_L}$ で表す．$\rightarrow_{R_L}$ の推移閉包は $\overset{*}{\rightarrow}_{R_L}$ で表す．$\overset{*}{\rightarrow}_{R_L}$ が簡約関係となるのは明らかである．

書換え系による合同関係の定義　$(l,r) \in R_L$ ならば $l \leftrightarrow r$ であるとすれば，R_L を与えることで Σ_L^* 上の合同関係を決めることができる．このようにして決められる合同関係を書換え系 R_L による合同関係と呼び，$\leftrightarrow_{R_L}$ で表す．$\leftrightarrow_{R_L}$ の推移閉包は $\overset{*}{\leftrightarrow}_{R_L}$ で表す．$\overset{*}{\leftrightarrow}_{R_L}$ が合同関係になるのは明らかである．

書換え系による合同類の定義　任意の $s \in \Sigma_L^*$ に対する書換え系 R_L による合同類を $[s]_{R_L}$ のように表し，次のように定義する．

$$[s]_{R_L} = \{x \in \Sigma_L^* \mid x \overset{*}{\leftrightarrow}_{R_L} s\}$$

書換え系による簡約の定義　任意の $l,r \in \Sigma_L^*$ に対して $l \overset{*}{\rightarrow}_{R_L} r$ が成り立つ時，l を r で置き換えることを書換え系 R_L による簡約と呼ぶ．

書換え系の正規形の定義　書換え系が簡約できない有界長文字列をその書換え系の正規形と呼ぶ．R_L の正規形の集合は $nf(R_L)$ のように表す．

書換え系による正規形への簡約の定義　任意の $l,r \in \Sigma_L^*$ に対して，$l \rightarrow_{R_L} r$ かつ $r \in nf(R_L)$ ならば，l から r への簡約を書換え系 R_L による正規形 r への簡約と呼び，$\mapsto_{R_L}$ で表す．また，$l \overset{*}{\rightarrow}_{R_L} r$ かつ $r \in nf(R_L)$ ならば，$l \overset{*}{\mapsto}_{R_L} r$ で表す．

書換え系の停止性の定義　書換え系 R_L の停止性とは，R_L があらゆる $s \in \Sigma_L^*$ に対して無限の簡約の列 $s \rightarrow_{R_L} s_1 \rightarrow_{R_L} \ldots$ を与えない性質である．

書換え系の合流性の定義　書換え系の合流性とは，書換え系があらゆる有界長文字列に対して二つ以上の正規形を与えない性質である．

書換え系の完備性の定義　書換え系の完備性とは，書換え系があらゆる有界長文字列に対して正規形を一意的に定める性質である．すなわち，停止性と合流性のある書換え系には完備性がある．

書換え系が生成する危険対の定義　l_1 の任意の接頭文字列と l_2 の任意の接尾文字列が同じ文字列であるような二つの書換え規則 $(l_1,r_1),(l_2,r_2)$ があるとする．このとき，同じ文字列部分で l_1 と l_2 を重ねてできる文字列をこの二つの書換え規則で簡約してできる二つの異なる有界長文字列対を危険対という．書換え系 R_L が生成する危険対とは，R_L のあらゆる要素の組み合わせでできる危険対の集合のことであり，これを $cp(R_L)$ で表す．

5. 書換え系による解法

等式集合の代わりに書換え系を使って有界長文字列の word 問題を解くためには，その書換え系が与えられた等式集合に対して完全でなければならない．すなわち，書換え系は次の 4 つの条件を満足しなければならない．

条件 1.　書換え系が有限である．

条件 2.　書換え系に停止性がある．

条件 3.　書換え系に合流性がある．

条件 4.　書換え系が与えられた等式集合に対して健全かつ十分である．すなわち，$\overset{*}{\leftrightarrow}_{E_L} = \overset{*}{\leftrightarrow}_{R_L}$ である．

もし与えられた等式集合 E_L に対して完全な書換え系 R_L が得られるなら，以下のアルゴリズムで有界長文字列の word 問題を解くことができる．

アルゴリズム　l,r を合同判定したい二つの有界長文字列とする．

(1)　$l \overset{*}{\mapsto}_{R_L} l', r \overset{*}{\mapsto}_{R_L} r'$ であるような $l',r' \in nf(R_L)$ を求める．

(2)　$l' = r'$ ならば $l \overset{*}{\leftrightarrow}_{E_L} r$ であり，$l' \neq r'$ ならば $l \overset{*}{\not\leftrightarrow}_{E_L} r$ である．

書換え系に完備性 (すなわち，停止性と合流性) があるなら，任意の有界長文字列に対して正規形が一意に決まるので，この判定アルゴリズムは停止する．よって，与えられた等式集合に対して完全な書換え系が生成できれば，このアルゴリズムで有界長文字列の word 問題が解ける．

一般的に，等式集合を書換え系に変換する手段として，完備化と呼ばれる手続きがある．[2] 有界長文字列上の等式集合を書換え系に書き換えるには，文字列書換え系に対する完備化アルゴリズムをそのまま使うのが実際的である．しかし，一般にこの方法で完備化された書換え系は元の等式集合に対して健全ではない．この問題を次の節で説明する．

6. 一般的な文字列書換え系の完備化

完備化はある簡約順序に基づいて等式集合を書換え系に書き換える手続きである．通常，完備化と言えば項書換え系の完備化のことであり，**Knuth-Bendix** の完備化アルゴリズム [2] で等式集合を書換え系に書き換えることを意味する．文字列書換え系は規則の両辺の項がある同じ変数で終端された単項的な項である項書換え系なので，**Knuth-Bendix** の完備化アルゴリズム

表 1　有界長文字列上の書換え系の完備化
Table 1　Completion for R_L

削除	$l = r$ のとき $(E_L \cup \{(l,r)\}; R_L) \quad \vdash (E_L; R_L)$
方向付け	$l > r$ のとき $(E_L \cup \{(l,r)\}; R_L) \quad \vdash (E_L; R_L \cup \{(l,r)\})$ $l < r$ のとき $(E_L \cup \{(l,r)\}; R_L) \quad \vdash (E_L; R_L \cup \{(r,l)\})$ 但し，方向付ける等式は一つだけである．
畳込み	$l \notin nf(R_L)$ のとき $(E_L; R_L \cup \{(l,r)\}) \quad \vdash (E_L \cup \{(l,r)\}; R_L\})$
演繹	$(l,r) \in cp(R_L)$ のとき $(E_L; R_L) \quad \vdash (E_L \cup \{(l,r)\}; R_L)$
合成	$r \overset{*}{\mapsto}_{R_L} r'$ のとき $(E_L; R_L \cup \{(l,r)\}) \quad \vdash (E_L; R_L \cup \{(l,r')\})$
単純化	$l \overset{*}{\mapsto}_{R_L} l', r \overset{*}{\mapsto}_{R_L} r'$ のとき $(E_L \cup \{(l,r)\}; R_L) \quad \vdash (E_L \cup \{(l',r')\}; R_L)$

が応用できる．但し，文字列書換え系に対する完備化アルゴリズムは停止性が保証できない．一般には，完備化する際に文字列長に上限を設けて無限に規則が生成されるのを防ぐことで停止性を保証するのが実際的である．つまり，有界長文字列の書換え系も，文字列書換え系と同様な完備化を行なうことを考える．具体的には，与えられた等式集合 E_L と空の書換え系 R_L から始めて**表 1** に示す六つの推論規則に従って E_L が空になるまで推論系列を生成すれば良い．

続いて，こうして生成される書換え系が元の等式集合に対して完全であるかどうかを検討した．

1. **有限性の条件**　Σ_L^* は有限なので，Σ_L^* 上の書換え系 R_L も有限になる．よって，完備化された書換え系は有限性の条件を満足する．
2. **停止性の条件**　表 1 の方向付けの際，書換え規則は適用すると文字列が Σ_L^* 上の大小関係で小さくなるように方向付けられる．表 1 の推論規則の中で，書換え系に書換え規則を追加する推論規則は方向付けだけなので，完備化された書換え系の書換え規則は文字列を小さくする方向に統一されている．このような書換え系で有界長文字列を簡約すると徐々に小さな文字列に簡約されいずれ正規形になる．よって，完備化された書換え系は停止性の条件を満足する．
3. **合流性の条件**　ある書換え系 R_L の合流性を調べるには，R_L の規則が二つ以上適用できるあらゆる有界長文字列に対して正規形が一つしかないことを示せば良い．これは，R_L が生成する危険対の両辺の正規形が同じであるかどうかを調べることに相当する．R_L が生成する危険対の集合は，表 1 の演繹の際に求められて E_L に加えられる．こうしておけば，もし R_L に合流性があれば，表 1 の単純化と削除によりこれらの危険対が E_L から取り除かれる．したがって，完備化が終了するなら完備化された R_L は合流性がある．そこで，完備化アルゴリズムの停止性を検証する．
表 1 の推論規則は，E_L による合同関係と矛盾する等式や書換え規則を生成しない．したがって，$(l,r,) \in E_L$ または $(l,r) \in R_L$ ならば $[l]_{E_L} = [r]_{E_L}$ である．ここで，E_L と R_L の要素に対する E_L による合同類の和集合 S を考える．
$$S = \{x \in [l]_{E_L} | (l,r,g) \in E_L \cup R_L\}$$
S は完備化を進めると単調増加する性質がある．なぜなら，演繹以外は S の中でだけ行なわれる推論であり，演繹だけが唯一 S 以外の有界長文字列を含む危険対を生成する可能性があるからである．但し，$S \subseteq \Sigma_L^*$ なので S は有限である．したがって，演繹により新たに考慮される合同類には限界がある．もし演繹により S が大きくならなければ，完備化アルゴリズムは停止する．なぜなら，合成や単純化によって R_L や E_L の文字列が毎回正規形に簡約され，同じ正規形になった等式は削除の際に取り除かれるので，必要以上に等式が残ることはない．そして，毎回方向付けにより等式は確実に一つ減るので，S が大きくならないなら E_L は完備化が進むに連れて小さくなり，やがて空になる．よって，完備化アルゴリズムは停止する．したがって，完備化された書換え系は合流性の条件を満足する．
4. **合同関係の等価性の条件**　表 1 の各推論規則は，元の等式集合 E_L による合同関係で等しくない二つの有界長文字列からなる等式や書換え規則は一切生成しない．したがって，完備化された書換え系 R_L に対して $(l,r) \in R_L$ ならば $l \overset{*}{\leftrightarrow}_{E_L} r$ であり，$(l,r) \in E_L$ ならば $l \overset{*}{\leftrightarrow}_{R_L} r$ である．
ところで，$l \overset{*}{\leftrightarrow}_{R_L} r$ なら $l \overset{*}{\rightarrow}_{R_L} r$ または $r \overset{*}{\rightarrow}_{R_L} l$ であるが，完備化された書換え系 R_L はどちらの場合も文字列を単調に小さくする．すなわち，簡約の途中で l と r のどちらよりも大きな文字列になることはない．したがって，l, r の前後に共通の有界長文字列を連結してできる任意の $s_l, s_r \in \Sigma_L^*$ に対して $l \overset{*}{\leftrightarrow}_{R_L} r$ なら $s_l \overset{*}{\leftrightarrow}_{R_L} s_r$ である．故に，$(l,r) \in E_L$ ならば $s_l \overset{*}{\leftrightarrow}_{R_L} s_r$ なので R_L は E_L に対して十分である．
一方，与えられた等式集合 E_L の等式は適用すると文字列を大きくすることがあるので $l \overset{*}{\leftrightarrow}_{R_L} r$ であっても変形の途中で l と r のどちらよりも大きな有界長文字列がなければ変形できない可能性

表 2　一般的な完備化の実行例
Table 2　Example for general completion

	E_L	R_L	推論規則
0	("abc", "ab") ("abc", "bc")		
1	("abc", "bc")	("abc", "ab")	方向付け
	("ab", "bc")	("abc", "ab")	単純化
2		("bc", "ab") ("abc", "ab")	方向付け
	("abc", "ab")	("bc", "ab")	畳込み
	("aab", "ab")	("bc", "ab")	単純化
3		("bc", "ab") ("aab", "ab")	方向付け
	("$aaab$", "abc")	("bc", "ab") ("aab", "ab")	演繹
	("ab", "ab")	("bc", "ab") ("aab", "ab")	単純化
4		("bc", "ab") ("aab", "ab")	削除

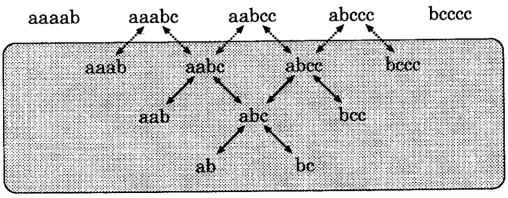

図 1　E_L による合同関係
Fig. 1　Congruence relations by E_L

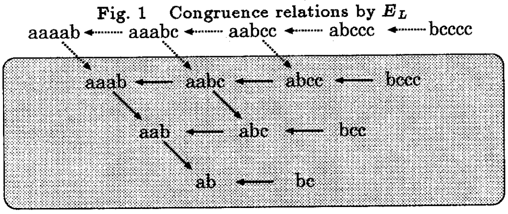

図 2　R_L による簡約関係
Fig. 2　Reduction relations by R_L

がある．この場合，l, r の前後に共通の有界長文字列を連結してできる任意の $s_l, s_r \Sigma_L^*$ に対して $s_l \overset{*}{\not\leftrightarrow}_{R_L} s_r$ となることがある．これを一つ例を上げて図を使って説明する．

$L \geq 3, E_L = \{("abc","ab"), ("abc","bc")\}$ の場合を考える．この場合，表 1 の推論規則で完備化すると**表 2** に示すようになり，書換え系 $R_L = \{("aab","ab"), ("bc","ab")\}$ が得られる．このとき，仮に $L = 4$ ならば E_L による合同関係と R_L による簡約関係はそれぞれ**図 1** と**図 2** のようになる．但し，図中の破線の矢印は文字列長に上限がなければ合同関係あるいは簡約関係であることを意味する．

図 1 では，有界長文字列の長さの制限により "$bccc$" に適用できる E_L の等式は一つもなく "$bccc$" $\overset{*}{\not\leftrightarrow}_{E_L}$ "ab" である．一方，図 2 では，"$bccc$" に ("bc","ab") $\in R_L$ が適用でき，最終的に "$bccc$" $\overset{*}{\leftrightarrow}_{R_L}$ "ab" となる．このように，書換え規則の両辺よりも大きな文字列が存在しなければ，元の等式集合による合同関係で等しくならないような書換え規則が存在すると完備化された書換え系は元の等式集合に対して健全ではない．よって，完備化された書換え系は合同関係の等価性の条件を満足しない．

我々は，等式集合と書換え系の合同関係の違いを無くして完全性の条件を満たすために，書換え規則に膨らみと呼ぶパラメータを持たせ，簡約される文字列の長さと適用する書換え規則の膨らみに応じて書換え規則の適用を制限する方法を本稿で提案する．

7. 膨らみ付き書換え系に関する定義と記法

まず，膨らみに関する事柄について定義する．なお，膨らみの概念が加わっても変更は書換え系で閉じており，合同類や正規形などの定義は膨らみのない等式集合や書換え系のものと同じ形式である．

等式集合による合同関係の上限の定義　任意の $l, r \in \Sigma_L^*$ に対して，$l \overset{*}{\leftrightarrow}_{E_L} r$ が成り立つような L の最小値を等式集合 E_N による合同関係の上限と呼び，$peak(l,r)_{E_L}$ のように表す．

$$peak(l,r)_{E_L} = \min_{l \overset{*}{\leftrightarrow}_{E_L} r} \{L\}$$

但し，$l \overset{*}{\not\leftrightarrow}_{E_L} r$ ならば $peak(l,r)_{E_L} = \infty$ とする．

等式集合による合同関係の膨らみの定義　任意の $l, r \in \Sigma_L^*$ に対する $peak(l,r)_{E_L}$ と l, r の大きい方の長さとの差を等式集合 E_N による合同関係の膨らみと呼び，$gap(l,r)_{E_L}$ のように表す．

$$gap(l,r)_{E_L} = peak(l,r)_{E_L} - max\{|l|, |r|\}$$

膨らみ付き書換え系の定義　有界長文字列の順序対にその膨らみを加えた三つ組を膨らみ付き書換え規則と呼ぶ．任意の膨らみ付き書換え規則の集合を膨らみ付き書換え系と呼び，$\hat{R_L}$ で表す．

膨らみ付き書換え系による簡約関係の定義　$(l, r, g) \in \hat{R_L}$ ならば l, r の前後に共通の有界長文字列を連結した任意の $s_l, s_r \in \Sigma_L^*$ に対して $max\{|s_l|, |s_r|\} + g \leq L$ の場合だけ $s_l \rightarrow s_r$ であるとすれば，$\hat{R_L}$ を与えることで Σ_L^* 上の簡約関係を決めることができる．このような簡約関係を膨らみ付き書換え系による簡約関係と呼び，$\rightarrow_{\hat{R_L}}$ で表す．$\rightarrow_{\hat{R_L}}$ の推移閉包は $\overset{*}{\rightarrow}_{\hat{R_L}}$ で表す．

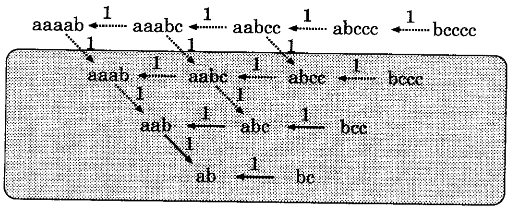

図 3　$\hat{R_L}$ による簡約関係
Fig. 3　Reduction relations by $\hat{R_L}$

膨らみ付き書換え系による合同関係の定義　$(l,r,g) \in \hat{R_L}$ ならば l,r の前後に共通の有界長文字列を連結した任意の $s_l, s_r \in \Sigma_L^*$ に対して $max\{|s_l|,|s_r|\}+g \leq L$ の場合だけ $s_l \leftrightarrow s_r$ であるとすれば，$\hat{R_L}$ を与えることで Σ_L^* 上の合同関係を決めることができる．このような簡約関係を膨らみ付き書換え系による合同関係と呼び，$\leftrightarrow_{\hat{R_L}}$ で表す．$\leftrightarrow_{\hat{R_L}}$ の推移閉包は $\overset{*}{\leftrightarrow}_{\hat{R_L}}$ で表す．

膨らみ付き等式集合の定義　有界長文字列の非順序対にその膨らみを加えた三つ組を膨らみ付き等式と呼ぶ．任意の膨らみ付き等式の集合を膨らみ付き等式集合と呼び，$\hat{E_L}$ で表す．

膨らみ付き等式集合による合同関係の定義　$(l,r,g) \in \hat{R_L}$ ならば l,r の前後に共通の有界長文字列を連結した任意の $s_l, s_r \in \Sigma_L^*$ に対して $max\{|s_l|,|s_r|\}+g \leq L$ の場合だけ $s_l \leftrightarrow s_r$ であるとすれば，$\hat{R_L}$ を与えることで Σ_L^* 上の合同関係を決めることができる．このような簡約関係を膨らみ付き書換え系による合同関係と呼び，$\leftrightarrow_{\hat{E_L}}$ で表す．$\leftrightarrow_{\hat{E_L}}$ の推移閉包は $\overset{*}{\leftrightarrow}_{\hat{E_L}}$ で表す．

膨らみの定義に基づいて図 2 の書換え系に膨らみを付けると，$\hat{R_L} = \{("aab","ab",1),("bc","ab",1)\}$ となる．この書換え系による簡約関係を**図 3** に示す．

図 3 において，"*aaab*" や "*bccc*" が $\hat{R_L}$ に対して正規形になり E_L による合同関係と一致することが分かる．しかし，その一方で "*aabc*" や "*abcc*" も正規形になり E_L による合同関係と矛盾してしまう．つまり，単に完備化された書換え系に膨らみを付けるだけでは等式集合に対して完全であるとは言えない．これは完備化して得られる書換え規則以外に別の膨らみ付き書換え規則を補って修正できるが，どのようにしてこのような規則を求めるかが新たな問題となる．我々は，このような規則は完備化の過程で一度生成される規則であることに着目し，このような規則が最後まで残るように完備化の手続きを修正した．この完備化は，膨らみを常に考慮しながら推論するので膨らみを考慮した完備化と呼ぶことにする．

8. 膨らみを考慮した完備化

膨らみ付き書換え系を生成するためには，等式集合にも膨らみがなければならない．そこで，膨らみの定義に基づいて最初に与えられる等式集合 E_L の全等式に膨らみ 0 を付け加えて膨らみ付き等式集合 $\hat{E_L}$ とする．膨らみを考慮した完備化は，この $\hat{E_L}$ と空の膨らみ付き書換え系 $\hat{R_L}$ から始めて**表 3** に示す五つの推論規則に従って $\hat{E_L}$ が空になるまで推論系列を生成する．

表 3 の各推論規則は次のような意味を持つ．

削除　同じ文字列からなる等式は推論上意味を持たないので取り除く．また，同じ文字列対からなる膨らみ付き等式が複数ある場合，膨らみの定義通り最も小さな膨らみの等式がだけを残す．

方向付け　等式を Σ_L^* 上の大小関係に従って順序付ける．他の推論規則は書換え規則を追加しないので $\hat{R_L}$ で文字列を簡約すると文字列は小さくなる．

演繹　$\hat{R_L}$ の合流性を検査するために $\hat{R_L}$ が生成する危険対を $\hat{E_L}$ に加える．各危険対の膨らみは，その危険対を生成する二つの膨らみ付き書換え規則から求まる．

合成　膨らみ付き書換え規則の第二項を正規形に簡約して単純化を容易にする．但し，膨らみをできるだけ小さく保ちたいので合成すると合同関係の上限が大きくなる場合は合成しない．

単純化　E_L の全等式を $\hat{R_L}$ で正規形に簡約める．もし正規形が同じになれば次の削除の際に取り除かれる．

次に，膨らみを考慮して完備化された膨らみ付き書換え系が完全性の条件を満足するかどうかを検証する．

1. **有限性の条件**　表 3 の削除の際に，同じ文字列対で膨らみの異なる膨らみ付き等式が複数あれば最も膨らみの小さい等式だけが残される．したがって，膨らみを考慮した完備化が終了した時点で同じ文字列対からなる複数の膨らみ付き書換え規則は存在せず，生成された膨らみ付き書換え系は有限性の条件を満足する．
2. **停止性の条件**　膨らみを考慮した完備化でも $\hat{R_L}$ は常に文字列を小さくするように簡約するので，膨らみを考慮して完備化された膨らみ付き書換え系は停止性の条件を満足する．
3. **合流性の条件**　膨らみを考慮しない完備化と同じ理由により，膨らみを考慮して完備化された膨らみ付き書換え系の合流性は膨らみを考慮した完備化アルゴリズムの停止性に依存する．したがって，膨らみを考慮した完備化アルゴリズムの停止性を

表 3　膨らみを考慮した完備化

Table 3　Completion consider a gap of congruence

削除	$l = r$ のとき	
	$(\hat{E_L} \cup \{(l, r, g)\}; \hat{R_L})$	$\vdash (\hat{E_L}; \hat{R_L})$
	$g' \geq g$ のとき	
	$(\hat{E_L} \cup \{(l, r, g), (l, r, g')\}; \hat{R_L})$	$\vdash (\hat{E_L} \cup \{(l, r, g)\}; \hat{R_L})$
方向付け	$l > r$ のとき	
	$(\hat{E_L} \cup \{(l, r, g)\}; \hat{R_L})$	$\vdash (\hat{E_L}; \hat{R_L} \cup \{(l, r, g)\})$
	$l < r$ のとき	
	$(\hat{E_L} \cup \{(l, r, g)\}; \hat{R_L})$	$\vdash (\hat{E_L}; \hat{R_L} \cup \{(r, l, g)\})$
	但し，方向付けるのは最も膨らみの小さな膨らみ付き等式一つだけである．	
演繹	$(l, r) \in cp(\hat{R_L})$ のとき	
	$(\hat{E_L}; \hat{R_L})$	$\vdash (\hat{E_L} \cup \{(l, r, gap(l, r)_{E_L})\}; \hat{R_L})$
合成	$r \overset{*}{\mapsto}_{\hat{R_L}} r'$ かつ $peak(l, r)_{E_L} \geq peak(r, r')_{E_L}$ のとき	
	$(\hat{E_L}; \hat{R_L} \cup \{(l, r, g)\})$	$\vdash (\hat{E_L}; \hat{R_L} \cup \{(l, r', g)\})$
単純化	$l \overset{*}{\mapsto}_{\hat{R_L}} l', r \overset{*}{\mapsto}_{\hat{R_L}} r'$ のとき	
	$(\hat{E_L} \cup \{(l, r, g)\}; \hat{R_L})$	$\vdash (\hat{E_L} \cup \{(l', r', gap(l', r')_{E_L})\}; \hat{R_L})$

検証する．

まず表 3 の各推論が必ず停止することを示す．

削除　Σ_L^* 上の $\hat{E}_L$ の大きさは有限なので停止する．

方向付け　方向付ける等式は一つだけなので停止する．

演繹　有限の $\hat{R}_L$ から生成される危険対の数は有限なので停止する．

合成　$\hat{R}_L$ が有限で停止性があるので停止する．

単純化　$\hat{E}_L$ が有限で $\hat{R}_L$ に停止性があるので停止する．

次に膨らみを考慮した完備化アルゴリズム全体の停止性を示す．表 3 には表 1 の畳込みがない．これは膨らみを考慮した場合に必要な規則を残すためである．畳込みを行なわないと演繹の際に毎回同じ危険対が生成されてしまうが，その時点で考慮している合同類の集合は変化しないのでアルゴリズムの停止性には影響しない．また，他の推論規則に関しては，膨らみを考慮してもその本質的な役割は変わっていないため，膨らみを考慮しない完備化アルゴリズムと同じ理由により膨らみを考慮した完備化アルゴリズムは停止する．したがって，膨らみを考慮して完備化された膨らみ付き書換え系は合流性の条件を満足する．

4. 合同関係の等価性の条件　まず，膨らみを考慮して完備化された膨らみ付き書換え系が健全であるかどうかを検証する．膨らみを考慮せずに完備化された書換え系が元の等式集合に対して健全でないのは，推論過程での合同関係の上限が有界長文字列の上界を超える場合に対処できないためであった．これは，明らかに書換え規則に膨らみを導入すれば解決可能である．また，膨らみを考慮せずに完備化された書換え系は元の等式集合に対して十分なので，明らかに膨らみを考慮した完備化が正しい膨らみを計算できれば，生成される膨らみ付き書換え系は元の等式集合に対して健全であると言える．そこで，表 3 の推論規則で正しい膨らみが求まるかどうかを検証する．

まず，完備化を始める時点で $\hat{E}_L$ の膨らみは正しい．削除は，不必要な等式と膨らみが不正に大きな等式しか削除しないので膨らみの計算に必要な等式は残されるので問題ない．方向付けは，明らかに合同関係の上限に影響しないので問題ない．演繹は，$\hat{R}_L$ が生成する危険対の膨らみが，危険対の元となる二つの膨らみ付き書換え規則から正しく求まるので問題ない．合成は，書換え規則の合同関係の上限を大きくしない範囲で簡約しているので問題ない．単純化は，等式の合同関係の上限を変えないので簡約後の等式の膨らみは正しい．なぜなら，方向付けの際に膨らみの小さな等式を優先しているため，書換え規則の膨らみが等式の膨らみ以下になるからである．よって，膨らみを考慮して完備化された膨らみ付き書換え系は元の等式集合に対して健全である．

次に，膨らみを考慮して完備化された膨らみ付き書換え系が十分であるかどうかを検証する．膨らみを考慮して完備化された膨らみ付き書換え系が十分であるためには，膨らみを考慮したために適用できる規則が無くなった有界長文字列に対して適用できる膨らみの小さな規則を残す必要がある．膨らみを考慮しない完備化で，このような規則を取り除いているのが畳込みである．膨らみを考慮

表 4　膨らみを考慮した完備化の実行例
Table 4　Example for completion consider a gap of congruence

	$\hat{E}_L$	$\hat{R}_L$	推論規則
0	("abc","ab",0) ("abc","bc",0)		
1	("abc","bc",0)	("abc","ab",0)	方向付け
	("bc","ab",1)	("abc","ab",0)	単純化
2		("abc","ab",0) ("bc","ab",1)	方向付け
	("aab","ab",1)	("abc","ab",0) ("bc","ab",1)	演繹
3		("abc","ab",0) ("bc","ab",1) ("aab","ab",1)	方向付け
	("aab","ab",1) ("aaab","abc",1) ("abc","aab",2)	("abc","ab",0) ("bc","ab",1) ("aab","ab",1)	演繹
	("ab","ab",0)	("abc","ab",0) ("bc","ab",1) ("aab","ab",1)	単純化
4		("abc","ab",0) ("bc","ab",1) ("aab","ab",1)	削除

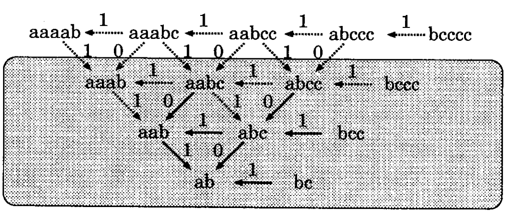

図 4　E_L に対して完全な $\hat{R}_L$ による簡約関係
Fig. 4　Reduction relations by complete $\hat{R}_L$ for E_L

した完備化では，畳込みを行なわないので膨らみを考慮して完備化された膨らみ付き書換え系は元の等式集合に対して十分である．よって，膨らみを考慮して完備化された膨らみ付き書換え系は合同関係の等価性の条件を満足する．

以上のことから膨らみを考慮した完備化は，元の等式集合に対して完全な膨らみ付き書換え系を生成する．よって，膨らみを考慮して完備化された膨らみ付き書換え系を使って有界長文字列の word 問題を正確に解くことができる．

9. 膨らみを考慮した完備化の実行例

図 1 に示した等式集合 E_L を与えた場合の膨らみを考慮した完備化の実行例を表 4 に示す．また，表 4 で生成された $\hat{R}_L$ による簡約関係を図 4 に示す．

図 1 と図 4 を比較すれば，E_L と $\hat{R}_L$ の合同関係が等しく $\hat{R}_L$ が E_L に対して完全であることが分かる．また，$\hat{R}_L$ による正規形が規則の適用順序に関わらず $\hat{R}_L$ による合同類の中で最も小さい有界長文字列になることが図 4 から確認できる．したがって，$\hat{R}_L$ による正規形を比較することで有界長文字列の word 問題を正確に解くことができる．

長さの制限が有効な例として $E_L = \{(\text{"}aba\text{"}, \text{"}ab\text{"})\}$ の場合を考える．もし文字列長に上限を設けずに完備化すると，$(\text{"}ab\ldots ba\text{"}, \text{"}ab\ldots b\text{"})$ のような形式の規則が無限に生成されるので完備化アルゴリズムは停止しない．これを例えば $L = 10$ として膨らみを考慮しないで完備化すると $R_L = \{(\text{"}aba\text{"}, \text{"}ab\text{"}), \ldots, (\text{"}abbbbbbbba\text{"}, \text{"}abbbbbbbb\text{"})\}$ となり，8 個の規則を生成して完備化が終了する．同様に，$L = 10$ として膨らみを考慮して完備化すると $R_L = \{(\text{"}aba\text{"}, \text{"}ab\text{"}, 0), (\text{"}abba\text{"}, \text{"}abb\text{"}, 1), (\text{"}abbba\text{"}, \text{"}abbb\text{"}, 2), (\text{"}abbbba\text{"}, \text{"}abbbb\text{"}, 2), (\text{"}abbbbba\text{"}, \text{"}abbbbb\text{"}, 3)\}$ となり，5 個の規則を生成して完備化が終了する．膨らみを考慮したことで生成されなくなった 3 つの規則は，推論の途中で有界長の上限を超えなければ合同であることを証明できない規則である．膨らみを考慮した場合，このような規則は与えられた等式集合による合同関係の上限が有界長の上限を超えるので生成されない．

10. おわりに

本稿では，有界長文字列空間において通常の文字列の完備化で生成される書換え系が元の等式集合に対して完全でないことを示し，書換え規則に膨らみを与えることでこの問題が解決することを述べた．また，等式集合に対して完全な膨らみ付き書換え系を生成する完備化アルゴリズムを定式化し，これにより有界長文字列の word 問題が正確に解けることを示した．

謝辞　本論文の内容について多くの有益な御意見を頂いた北陸先端科学技術大学院大学外山芳人教授に謹んで感謝の意を表する．

参考文献

1) R.V. BOOK, Thue systems as rewriting systems, J. Symbolic Comput.3(1987)39-68.
2) D.E.KNUTH and P.G.BENDIX, Simple word problems in universal algebras, in: J. Leech, ed., *Computational problems in abstract algebra* (Pergamon Press, Oxford, 1970) 263-297;reprinted in: *Automation of Reasoning 2* (Springer, Berlin, 1983) 342-376.