数理解析研究所講究録
第731巻 1990年 97-107

決定的に構文解析ができる 2次元アレイ文法のクラスについて

Uniquely Parsable Array Grammar

山本 泰則
Yasunori YAMAMOTO

森田 憲一
Kenichi MORITA

国立民族学博物館
National Museum of Ethnology

山形大学工学部
Faculty of Engineering, Yamagata University

Abstract

We define a new subclass of two-dimensional isometric array grammar, called Uniquely Parsable Array Grammar (UPAG). UPAG is a grammar class such that when we parse (*i.e.* rewrite by applying rewriting rules reversely) a given two-dimensional word, the rewritable portions of the word do not overlap each other and the rule that is applicable to each portion is uniquely determined. Thus parsing can be done in a deterministic manner. We study basic properties of UPAG. Then we define Monotone Terminating UPAG (MTUPAG), a subclass of UPAG, and show that any two-dimensional word generated by MTUPAG can be parsed in linear time (*i.e.* time proportional to the area of a given word) on RAM model.

1 はじめに

アレイ文法は, 記号の書き換えによって2次元の記号配列の集合を生成するシステムである. Rosenfeld ら [2, 4] によって提案されたアイソメトリック・アレイ文法 (IAG) では, 書き換え規則の左辺と右辺が幾何学的に同じ形をしているという条件がつけられおり, それをみたすために空白を表す記号 # がもちいられる. この条件によって, 記号配列の局所的な書き換えが配列全体に歪をあたえないようにすることができる.

1次元の記号列についての生成文法と同様に, 書き換え規則にさまざまな制約をつけることにより, IAG の3つのサブクラスが定義されている. それらは, 単調アレイ文法, 文脈自由アレイ文法, 正規アレイ文法とよばれ, 1次元の文法に似た Chomsky 風の階層をなしている [1].

正規アレイ文法は, その階層の最下位に位置するクラスである. その書き換え規則は, 左辺はただ1つの非終端記号, 右辺はたかだか1つの非終端記号とただ1つの終端記号をもつのみである. 記号配列の生成過程で現れる非終端記号は, 各ステップでただ1つであり, それに書き換え規則を適用するときも, まわりの記号を文脈としてみることができない.

このような単純な形式にもかかわらず, 正規アレイ文法の生成能力は, 予想以上に高い. 山本ら [7] は, 正規アレイ文法によって長方形や正方形のような幾何学図形の集合を生成できることを証明した. このような図形集合の生成には, 文脈による生成のコントロールが必要である. それが正規アレイ文法において可能であるのは,「アイソメトリック」という要請のために書き換え規則の左辺にに含まれる # 記号を利用して, 空白という一種の文脈 (図形のかたち) を正規アレイ文法が検出できるからである.

一方, そのため正規アレイ文法の決定問題は, 複雑になる. 森田ら [3] は, 正規アレイ文法においては, 所属問題は NP 完全になり, また, 空問題や等価問題は決定不能になることを証明した. このことは, 正規アレイ文法においては, 構文解析が事実上不可能であることを意味する.

本稿では, IAG の新しいサブクラスとして, 一意解析可能アレイ文法 (UPAG) を提案する. UPAG の書き換え規則の集合は, 右辺を互いに重ね合わせたとき共通部分をもたないような規則だけからなっている. そのため, 構文解析を決定的に進めることができるという特徴をもつ. 本稿では, まず UPAG のいくつかの基本性質を示す. つぎに, UPAG のサブクラスである単調終端 UPAG (MTUPAG) を定義し, MTUPAG の構文解析は, RAM モデルで線形時間, つまり入力の大きさ (面積) に比例した時間できることを証明する.

2 諸定義

この節では, まず, アイソメトリック・アレイ文法 (IAG) に関する基本的な定義をおこない, つぎに本稿で提案する一意解析可能アレイ文法 (UPAG) について定義する.

Σ を記号の空でない有限集合とする. Σ 上の (2次元の) 語とは, Σ の記号の2次元有限連結配列である. Σ 上のすべての語の集合を Σ^{2+} で表す (ただし, 空語は Σ^{2+} に含まれない). アイソメトリック・アレイ文法 (*Isometric Array Grammar: IAG*) とは, つぎの5項組である.

$$G = (N, T, P, S, \#)$$

ここで, N は非終端記号の空でない有限集合; T は終端記号の空でない有限集合; $N \cap T = \emptyset$; P は $\alpha \to \beta$ という形式の書き換え規則の有限集合. ただし, α と β は $N \cap T \cap \{\#\}$ 上の語で, 次の条件をみたす:

1. α と β の形は, 幾何学的に同じ形である.
2. α は少なくとも 1 つの非終端記号を含む.
3. α の終端記号は, 書き換え規則 $\alpha \to \beta$ の適用によって書き換えられない.
4. 書き換え規則 $\alpha \to \beta$ を適用しても, 適用を受けた配列の連結性は保存される.
   (詳しくは, 文献 [5] を参照のこと.)

$S\,(\in N)$ は開始記号; $\#\,(\notin N \cup T)$ は空白記号を表す.

語 $\xi \in (N \cup T)^{2+}$ を $\#$ の 2 次元無限配列に埋め込んだものを, $\xi_{\#}$ と記す. IAG G において, 語 η が ξ から直接導出されるとは, G の書き換え規則 $\alpha \to \beta$ があって, $\xi_{\#}$ 中の部分配列 α の 1 つを β に書き換えて, $\eta_{\#}$ が得られることをいい, $\xi \underset{G}{\Rightarrow} \eta$ と表す. G において, 語 ξ から語 η への $n\,(n \geq 0)$ ステップの直接導出,

$$\xi = \zeta_0 \underset{G}{\Rightarrow} \zeta_1 \cdots \underset{G}{\Rightarrow} \zeta_n = \eta$$

があるとき, これを ξ から η への長さ n の導出といい, $\xi \overset{n}{\underset{G}{\Rightarrow}} \eta$ と書く. また, 関係 $\underset{G}{\Rightarrow}$ の反射的かつ推移的閉包を $\overset{*}{\underset{G}{\Rightarrow}}$ で表す. $\xi \overset{*}{\underset{G}{\Rightarrow}} \eta$ のとき, η は ξ から導出されるという. 文法 G が明らかなときは, G を省略して, $\underset{G}{\Rightarrow}$, $\overset{n}{\underset{G}{\Rightarrow}}$, $\overset{*}{\underset{G}{\Rightarrow}}$ をそれぞれ, $\Rightarrow$, $\overset{n}{\Rightarrow}$, $\overset{*}{\Rightarrow}$ と書く.

$S \overset{*}{\Rightarrow} \sigma\,(\in (N \cap T)^{2+})$ であるとき, σ を G における文形式という. G によって生成される (2 次元)言語を $L(G)$ で表し, つぎように定義する.

$$L(G) = \{\, \sigma \mid S \overset{*}{\Rightarrow} \sigma, \text{ かつ } \sigma \in T^{2+} \,\}.$$

定義 2.1 $G = (N, T, P, S, \#)$ を IAG とする. P の規則 $\alpha \to \beta$ が $\xi \in (N \cap T)^{2+}$ に位置 (i, j) で適用可能とは, $\xi_{\#}$ 中に α が部分配列として存在し, かつその位置が (i, j) であることをいう. ただし, $\xi_{\#}$ 中の有限部分配列の位置とは, その部分配列の最上行の最左の記号の, $\xi_{\#}$ における座標をさす. $\xi_{\#}$ 中の部分配列 α のうちで, 位置 (i, j) にあるものを β で置き換えて得られる配列を $\eta_{\#}$ とするとき, ξ から η が, ラベル $L = [\alpha \to \beta, (i, j)]$ をもつ書き換えによって導出されるといい, $\xi \overset{L}{\Rightarrow} \eta$ と書く.

定義 2.2 $G = (N, T, P, S, \#)$ を IAG とする. P の規則 $\alpha \to \beta$ が $\eta \in (N \cap T)^{2+}$ に位置 (i, j) で逆適用可能とは, $\eta_{\#}$ 中に β が部分配列として存在し, かつその位置が (i, j) であることをいう. $\eta_{\#}$ 中の部分配列 β のうちで, 位置 (i, j) にあるものを α で置き換えて得られる配列を $\xi_{\#}$ とするとき, η から ξ が, ラベル $L = [\alpha \to \beta, (i, j)]$ をもつ書き換えによって還元されるといい, $\eta \overset{L}{\models} \xi$ と書く.

明らかに

$$\eta \overset{L}{\models} \xi, \text{ iff } \xi \overset{L}{\Rightarrow} \eta$$

が成り立つ. なお, ラベルに言及しないときは, $\overset{L}{\Rightarrow}$, $\overset{L}{\models}$ を, それぞれ, $\Rightarrow$, $\models$ と書く.

書き換え規則 $\alpha \to \beta$ を逆適用したとき, それによって書き換えられない (同じ記号に書き換えられる) β の部分配列を β の文脈部分, 真に書き換えられる β の部分配列を β の書き換え部分と呼ぶ. 同様に, $\alpha \to \beta$ を適用したとき, それによって書き換えられない α の部分配列を α の文脈部分, 真に書き換えられる α の部分配列を α の書き換え部分と呼ぶ.

つぎに, 一意解析可能アレイ文法 (UPAG) を定義する. UPAG は, IAG のサブクラスで, 構文解析が一意的にできるような文法である.

定義 2.3 IAG $G = (N, T, P, S, \#)$ において, P の規則がつぎの条件をみたすとき, G を一意解析可能アレイ文法 (*Uniquely Parsable Array Grammar: UPAG*) と呼ぶ.

1. 右辺は, # でも S でもない記号を少なくとも 1 つ含む.
2. P の任意の規則 r_1, r_2 について, それぞれの右辺をどのような位置で重ね合わせても, 重なっている部分の記号がすべて一致するならば,
   (a) それらはすべて, それぞれ r_1 と r_2 の文脈部分に属しているか, または,
   (b) 右辺全体が完全に重なっており, かつ $r_1 = r_2$ である.

条件 2は, $r_1 = r_2$ の場合にも課せられることに注意.

定義 2.4 UIAG $G = (N, T, P, S, \#)$ において, P のどの規則も右辺の終端記号の数が左辺の終端記号の数よりも真に多いならば, G を単調終端 **UPAG** (*Monotone Terminating UPAG: MTUPAG*) と呼ぶ.

UPAG の例を以下に示す.

例 T 字形 (図 1) の集合を生成する UPAG G.

$$G = (\{S, R, D\}, \{a\}, P, S, \#)$$

P の要素はつぎのとおり.

$$
\begin{array}{c}
a\ a\ a\ a\ a\ a\ a\ a\ a\ a\ a\ a \\
a \\
a \\
a \\
a \\
a
\end{array}
$$

図 1: T 字形

$$
S\ \#\ \to\ a\ S\ ,\qquad
\begin{array}{ccc} S & \# & \to \\ \# & & \end{array}\ \begin{array}{c} a\ R \\ D \end{array}\ ,
$$

$$
\begin{array}{ccc} R & \# & \\ \# & & \end{array}\ \to\ \begin{array}{c} a\ R \\ \# \end{array}\ ,\qquad
\begin{array}{ccc} D & \# & \\ \# & & \end{array}\ \to\ \begin{array}{c} a\ \# \\ D \end{array}\ ,
$$

$$
\begin{array}{ccc} \# & \# & \\ R & \# & \# \\ \# & \# & \end{array}\ \to\ \begin{array}{ccc} \# & \# & \\ a & a & \# \\ \# & \# & \end{array}\ ,\qquad
\begin{array}{ccc} \# & D & \# \\ \# & \# & \# \\ & \# & \end{array}\ \to\ \begin{array}{ccc} \# & a & \# \\ \# & a & \# \\ & \# & \end{array}\ .
$$

G は MTUPAG の条件もみたしていることに注意.

3 UPAG の基本性質

この節では, UPAG における語の還元に関するいくつかの補題を示す.

補題 3.1 $G=(N, T, P, S, \#)$ を UPAG とする. P 中の規則 $\alpha \to \beta$ が $\eta \in (N \cup T)^{2+}$ に位置 (i,j) で逆適用可能とする. このとき, もし,

$$
\begin{aligned}
&\xi \overset{L_1}{\Rightarrow} \zeta_1 \overset{L_2}{\Rightarrow} \cdots \overset{L_{n-1}}{\Rightarrow} \zeta_{n-1} \overset{L_n}{\Rightarrow} \eta, \\
&L_1 = [\alpha \to \beta, (i,j)], \\
&L_k \neq [\alpha \to \beta, (i,j)] \quad (1 < k \leq n)
\end{aligned}
$$

が成り立つならば,

$$
\xi \overset{L_2}{\Rightarrow} \zeta_1' \overset{L_3}{\Rightarrow} \cdots \overset{L_{n-1}}{\Rightarrow} \zeta_{n-2}' \overset{L_n}{\Rightarrow} \zeta_{n-1}' \overset{L_1}{\Rightarrow} \eta
$$

となるような $\zeta_1', \ldots, \zeta_{n-1}'$ が存在する.

証明　n についての帰納法で証明する. $n=1$ のときは, 自明である. $n>1$ のとき, 規則 $\alpha \to \beta$ が η に位置 (i,j) で逆適用可能で,

$$\xi \overset{L_1}{\Rightarrow} \zeta_1 \overset{L_2}{\Rightarrow} \cdots \overset{L_n}{\Rightarrow} \zeta_n \overset{L_{n+1}}{\Rightarrow} \eta,$$
$$L_1 = [\alpha \to \beta, (i,j)],$$
$$L_k \neq [\alpha \to \beta, (i,j)] \quad (1 < k \leq n+1)$$

であったとしよう. $L_1 \neq L_{n+1}$ であるので, η に L_{n+1} を逆適用したとき, UPAG の定義より, η 中の位置 (i,j) にある部分配列 β の書き換え部分の記号が, 書き換えられることはない. したがって, 規則 $\alpha \to \beta$ は ζ_n においても, 位置 (i,j) で逆適用可能である. いま, $\xi, \zeta_1, \ldots, \zeta_n$ に対して帰納法の仮定を適用すると,

$$\xi \overset{L_2}{\Rightarrow} \zeta'_1 \overset{L_3}{\Rightarrow} \cdots \overset{L_{n-1}}{\Rightarrow} \zeta'_{n-1} \overset{L_1}{\Rightarrow} \zeta_n$$

となる $\zeta'_1, \ldots, \zeta'_{n-1}$ が存在する. さらに, $L_1 \neq L_{n+1}$ であるので,

$$\zeta'_{n-1} \overset{L_{n+1}}{\Rightarrow} \zeta'_n \overset{L_1}{\Rightarrow} \eta$$

となる ζ'_n が存在する. したがって,

$$\xi \overset{L_2}{\Rightarrow} \zeta'_1 \overset{L_3}{\Rightarrow} \cdots \overset{L_{n-1}}{\Rightarrow} \zeta'_{n-1} \overset{L_{n+1}}{\Rightarrow} \zeta'_n \overset{L_1}{\Rightarrow} \eta$$

となる $\zeta'_1, \ldots, \zeta'_n$ が存在する. □

補題 3.2　$G=(N, T, P, S, \#)$ を UPAG とする. P 中の規則 $\alpha \to \beta$ が $\eta \in (N \cup T)^{2+}$ に位置 (i,j) で逆適用可能とする. このとき, もし,

$$S \overset{L_1}{\Rightarrow} \zeta_1 \overset{L_2}{\Rightarrow} \cdots \overset{L_{n-1}}{\Rightarrow} \zeta_{n-1} \overset{L_n}{\Rightarrow} \eta \tag{1}$$

であるならば,

$$L_k = [\alpha \to \beta, (i,j)]$$

となるような $k (1 \leq k \leq n)$ が存在する.

証明　導出 (1) において, ラベル $[\alpha \to \beta, (i,j)]$ をもつ書き換えが適用されなかったと仮定しよう. UPAG の定義により, η 中の位置 (i,j) にある部分配列 β には, $\#$ と S 以外の記号が少なくとも 1 つは含まれている. したがって, S から η を導出する導出過程で, そのような記号を (書き換え部分の記号として) 生みだすような書き換えは必ず行なわれなければならない. ゆえに, ある $p (1 \leq p \leq n)$ が存在して, η 中の位置 (i,j) にある β 中のある記号 $X (\in N \cup T \cup \{\#\})$ が, ラベル $L_p = [\gamma_p \to \delta_p, (i_p, j_p)] \neq [\alpha \to \beta, (i,j)]$ をもつ書き換えの結果, δ_p の書き換え部分の記号として生じることになる. このような p のうち最大のものを考えると, β の一部分が δ_p の書き換え部分と共通部分をもつので, 2 つの規則 $\alpha \to \beta$ と $\gamma_p \to \delta_p$ は UPAG の定義に反することになる. 以上により, この補題が成立する. □

補題 3.3　$G=(N, T, P, S, \#)$ をUPAGとする．P 中の規則 $\alpha \to \beta$ が $\eta \in (N \cup T)^{2+}$ に位置 (i,j) で逆適用可能とする．このとき，もし，

$$S \overset{n}{\Rightarrow} \eta$$

であるならば，

$$S \overset{n-1}{\Rightarrow} \zeta \overset{L}{\Rightarrow} \eta \quad (\text{ただし}, L=[\alpha \to \beta, (i,j)])$$

となるような ζ が存在する．

証明　S から η を導く n ステップの導出を任意に選び，

$$S \overset{L_1}{\Rightarrow} \zeta_1 \overset{L_2}{\Rightarrow} \cdots \overset{L_{n-1}}{\Rightarrow} \zeta_{n-1} \overset{L_n}{\Rightarrow} \eta$$

とする．このとき，補題 3.2より，

$$S \overset{L_1}{\Rightarrow} \zeta_1 \overset{L_2}{\Rightarrow} \cdots \overset{L_{k-1}}{\Rightarrow} \zeta_{k-1} \overset{L_k}{\Rightarrow} \zeta_k \overset{L_{k+1}}{\Rightarrow} \cdots \overset{L_{n-1}}{\Rightarrow} \zeta_{n-1} \overset{L_n}{\Rightarrow} \eta, \qquad (2)$$
$$L_k=[\alpha \to \beta, (i,j)]$$

となる k $(1 \le k \le n)$ が存在する．そのような k のうち，最大のものをとる．$L_k \neq L_m$ $(k < m \le n)$ であるので，

$$\zeta_{k-1} \overset{L_k}{\Rightarrow} \zeta_k \overset{L_{k+1}}{\Rightarrow} \cdots \overset{L_{n-1}}{\Rightarrow} \zeta_{n-1} \overset{L_n}{\Rightarrow} \eta$$

に，補題 3.1 を適用すると，

$$\zeta_{k-1} \overset{L_{k+1}}{\Rightarrow} \zeta'_k \overset{L_{k+2}}{\Rightarrow} \cdots \overset{L_n}{\Rightarrow} \zeta'_{n-1} \overset{L_k}{\Rightarrow} \eta \qquad (3)$$

となる $\zeta'_k, \ldots, \zeta'_{n-1}$ が存在する．導出 (2) の後半を導出 (3) で置き換えることにより，導出

$$S \overset{L_1}{\Rightarrow} \zeta_1 \overset{L_2}{\Rightarrow} \cdots \overset{L_{k-1}}{\Rightarrow} \zeta_{k-1} \overset{L_{k+1}}{\Rightarrow} \zeta'_k \overset{L_{k+2}}{\Rightarrow} \cdots \overset{L_n}{\Rightarrow} \zeta'_{n-1} \overset{L_k}{\Rightarrow} \eta$$

すなわち，

$$S \overset{n-1}{\Rightarrow} \zeta'_{n-1} \overset{L_k}{\Rightarrow} \eta$$

が導ける．　□

この補題は，文形式 η に逆適用可能な書き換えがいくつかあったとき，どの書き換えを最初に実行しても同じステップ数で S まで到達できることを示している．その意味で，UPAG は構文解析を一意的に進めることができる文法である．

```
    begin
1.      x 全体を走査し, x に逆適用可能な書き換え規則とその位置の
            対 L = [α → β, (i, j)] (つまり, 書き換えのラベル) を要
            素とするリスト iapplicable を作成する;
2.      ζ := x;
3.      while iapplicable が空でない do begin
4.          iapplicable から先頭の要素 L = [α → β, (i, j)] をと
                りだし, ζ に逆適用し, その結果を新しい ζ の値と
                する;
5.          ζ の位置 (i, j) のまわりを走査し, 逆適用可能な P の書
                き換え規則で, その右辺が, ζ の位置 (i, j) における
                部分配列 α の書き換え部分と共通部分をもつもの
                について, その書き換えのラベルを iapplicable
                の先頭に追加する
        end
6.      if ζ は # とただ1つの S からなる then
7.          x を受理する (x ∈ L(G) と判定)
        else
8.          x を受理しない (x ∉ L(G) と判定)
    end
```

図 2: アルゴリズム A

4 MTUPAG の構文解析時間

この節では, UPAG のサブクラスである MTUPAG の構文解析が, 線形時間つまり入力語の面積に比例した時間で可能であることを示す.

定理　MTUPAG $G = (N, T, S, P, \#)$ と語 $x \in T^{2+}$ が任意に与えられたとき, $x \in L(G)$ かどうかは, RAM モデルで $O(|x|)$ 時間[1] で判定可能である.

証明　判定アルゴリズム A を図 2に与える.

まず, このアルゴリズムが $x \in L(G)$ かどうかを正しく判定できることを証明する.

いま, アルゴリズム A が実行され, $x \in L(G)$ と判定したとしよう. このとき, アルゴリズム A の 4 行目で実行される還元を追跡すると,

$$x \overset{L_1}{\models} \zeta_1 \overset{L_2}{\models} \zeta_2 \overset{L_3}{\models} \cdots \overset{L_n}{\models} S$$

[1] $|x|$ は, 語 x の面積を表す.

という列ができる. $\eta \overset{L}{\models} \xi$ と $\xi \overset{L}{\Rightarrow} \eta$ は同値であるので, これは,

$$S \overset{L_n}{\Rightarrow} \cdots \overset{L_3}{\Rightarrow} \zeta_2 \overset{L_2}{\Rightarrow} \zeta_1 \overset{L_1}{\Rightarrow} x$$

という導出の列が存在することを意味する. すなわち, $x \in L(G)$ が成り立つ.

逆に, 任意の $x \in L(G)$ についてアルゴリズム A を実行したとき, A は x を受理することを示そう.

$x \in L(G)$ であるので, ある n に対して, 導出 $S \overset{n}{\Rightarrow} x$ が存在する. このとき x を入力としてアルゴリズム A を実行すると, つぎの命題が成り立つ.

命題 任意の k $(1 \leq k \leq n)$ について, k 回目にアルゴリズムの 3 行目が実行されるとき, iapplicable は ζ に逆適用可能な書き換えのラベルのみをもれなく含み, かつ $S \overset{n-k+1}{\Rightarrow} \zeta$ という導出が存在する.

k についての数学的帰納法で証明する.

$k = 1$ のとき. アルゴリズムの 1 行目と 2 行目により, iapplicable は, $\zeta\ (= x)$ に逆適用可能な書き換えのラベルのみをもれなく含む. また, $S \overset{n}{\Rightarrow} \zeta\ (= x)$ である.

k $(1 \leq k < n)$ のときこの命題が成り立つとして, $k+1$ 回目の **while** ループの実行を考えよう. 3 行目が実行されるとき, $\zeta = \zeta_1$ とすれば, 帰納法の仮定より,

$$S \overset{n-k}{\Rightarrow} \zeta_2 \overset{L_1}{\Rightarrow} \zeta_1$$

という導出があり, L_1 は iapplicable に含まれている. よって, iapplicable は空でないので, **while** ループの内側が実行される. 4 行目の実行の結果, iapplicable から選ばれた $L = L_1'\ (= [\alpha \to \beta, (i,j)]$, $L_1' = L_1$ とは限らない) が逆適用されて, $\zeta = \zeta_1$ は ζ_2' に還元されて, $\zeta = \zeta_2'$ となったとする. UPAG の定義により, iapplicable の要素はすべて ζ_2' に逆適用可能な書き換えのラベルでもある. さらにこの還元の結果, あらたに ζ_2' に逆適用可能になる書き換え規則と位置は, その右辺が, ζ_2' 中の位置 (i,j) における部分配列 α の書き換え部分と共通部分をもつものに限られるので, それらは, 5 行目の実行の結果すべて iapplicable に加えられる. このとき, iapplicable の中には, 重複するラベルはない. したがって, iapplicable は $\zeta = \zeta_2'$ に逆適用可能な書き換えのラベルのみをもれなく含んでいる. 一方補題 3.3 により,

$$S \overset{n-k}{\Rightarrow} \zeta_2' \overset{L_1'}{\Rightarrow} \zeta_1$$

という導出ができるので, $S \overset{n-k}{\Rightarrow} \zeta\ (= \zeta_2')$ である. 以上により, この命題は成立する. □

上の命題により, **while** ループは n 回まで実行される. n 回目に **while** ループの 3 行目が実行されるときには, $S \Rightarrow \zeta$ という導出があり, **iapplicable** は ζ に逆適用可能な書き換えのラベル, すなわち, 書き換え規則 $S \rightarrow \zeta$ (あるいは, この両辺にいくつかの # がついていてもよい) とその位置の対のみを含んでいる. したがって, 4 行目, 5 行目が実行され, $\zeta = S$ となり, **iapplicable** は空になる. つぎに 3 行目が実行されたとき, **while** ループの実行は終了し, 6 行目の **if** 文が実行され, 7 行目で x は受理される.

最後に, アルゴリズム A の RAM モデルによる時間計算量をもとめる. 1 行目の実行は $O(|x|)$ 時間でできる. 2 行目, 4 行目, 7 行目, 8 行目の実行は, それぞれ, 定数時間ですむ. 5 行目では, たかだか 2 点 $(i-h, j-2w)$, $(i+2h, j+2w)$ (h, w は, それぞれ, P の書き換え規則の各辺の高さと幅の最大値とする) で定まる ζ の長方形領域を走査すればよいので, その実行は定数時間で終る. よって, **while** ループ 3 〜 5 行目の実行は, ループの回数 (つまり, 書き換え規則の逆適用の回数) に比例した時間でできる. MTUPAG の定義により, 書き換え規則を 1 回逆適用すると文形式から少なくとも 1 つの終端記号がなくなるので, ($x \in L(G)$ の場合も $x \notin L(G)$ の場合も) x への逆適用は高々 $|x|$ 回可能である. したがって, **while** ループの実行に要する時間は全体で $O(|x|)$ となる. また, 6 行目の実行は, たかだか $O(|x|)$ 時間でよい. 以上により, アルゴリズム A の時間計算量は, $O(|x|)$ である. □

5 むすび

本稿では, アイソメトリック・アレイ文法 (IAG) のサブクラスとして, 一意解析可能アレイ文法 (UPAG) を提案し, その基本的性質を調べた. UPAG は, 記号配列に逆適用できる書き換え規則とその位置が, 一意的に決るようなアレイ文法である. さらに, そのサブクラスである単調終端 UPAG (MTUPAG) を定義し, その構文解析が, RAM モデルで面積時間でできることを証明した.

未解決の問題としては, つぎのようなものがあげられる. Chomsky 風階層を構成する IAG のサブクラスとの包含関係はどうか. 意味のある図形を生成する UPAG や MTUPAG が存在するか. これらの 2 次元文法の生成する言語のクラスにちょうど対応する機械のモデルがあるか.

参考文献

[1] Cook, C.R. and Wang, P.S.P., "A Chomsky hierarchy of isotonic array grammars and languages," *Computer Graphics and Image Processing*, **8**: 1, 144–152 (1978).

[2] Milgram, D.L. and Rosenfeld, A., "Array automata and array grammars," *Information Processing* **71**, 69–74 (1972).

[3] Morita, K., Yamamoto, Y., and Sugata, K., "The complexity of some decision problems about two-dimensional array grammars," *Information Sciences*, **30**, 241–262 (1983).

[4] Rosenfeld, A., "Isotonic grammars, parallel grammars, and picture grammars," in *Machine Intelligence VI*, Eds. D. Michie and B. Meltzer, pp.281–294, University of Edinburgh Press, Scotland (1971).

[5] Rosenfeld, A., *Picture Languages*, Academic Press, New York (1979).

[6] Yamamoto, Y., Morita, K., and Sugata, K., "An isometric context-free array grammar that generates rectangles," *Trans. IECE Japan*, **E65**, 754–755 (1982).

[7] Yamamoto, Y., Morita, K., and Sugata, K., "Context-sensitivity of two-dimensional regular array grammars," *International Journal of Pattern Recognition and Artificial Intelligence*, **3**: 3 & 4, 295–319 (1989) and in *Array Grammars, Patterns and Recognizers*, Ed. P.S.P. Wang, World Scientific Publ., Singapore, pp.17–41 (1989).

[8] Wang, P.S.P., "Hierarchical structures and complexities of parallel isometric languages," *IEEE Trans.* **PAMI-5**, 92–99 (1983).