Theoretical Computer Science 24 (1983) 195–205
North-Holland

INFINITE ARRAYS AND INFINITE COMPUTATIONS

Rani SIROMONEY, V. Rajkumar DARE and K.G. SUBRAMANIAN
Department of Mathematics, Madras Christian College, Tambaram, Madras 600 059, India

Abstract. A complete metric topology is introduced on the set of all finite and infinite arrays and the topological properties of the space are studied. In this complete metric topology, infinite arrays are the limits of increasing sequences of finite arrays. The notion of successful infinite derivations in Generalized Context-free Kolam Array Grammars, yielding infinite arrays, is introduced. This concept is strengthened for parentheses-free context-free kolam array grammars, a subclass of Generalized context-free kolam array grammars. For this class, the finite array language generated by a reduced grammar in Greibach normal form and the set of infinite arrays generated by it are related through the notion of adherence.

1. Introduction

The notion of successful infinite computation has been used to define the semantics of recursive programs by Nivat [4]. While computing in a partially ordered domain, which contains infinite ascending chains, the computed value may be the lub of such a chain and may fail to be the result of a finite computation. For instance, in the domain of real numbers, starting with rational numbers, if we perform computations using the four arithmetic operations, we would have computed only a rational number, after any finite lapse of time, whereas the end result of such a computation may be an irrational number. Thus attempts have been made to give a proper meaning to successful infinite computation sequences and Nivat has found it convenient to replace the order structure on the computation domain by a complete metric topology. This helps to arrive at conditions for the equivalence of the computed function obtained by this method with that obtained by means of fixed points. In the course of his study, Nivat extends the computation domain, which is usually the free monoid over a finite alphabet by adding infinite words to the domain.

Infinite words or ω-words have been the subject of study in several other investigations arising from different motivations [2]. Extension to infinite arrays has been made in [3].

In this paper, we consider infinite arrays, which are two-dimensional analogs of infinite words, and define them as extensions of finite rectangular arrays. We introduce a complete metric topology on the set of all finite and infinite arrays and study the topological properties of the space. We note that the space is in fact totally bounded and compact. Also, we note that infinite arrays, are the limits of

0304-3975/83/$03.00

increasing sequences of finite arrays, in the metric topology. The notion of adherence of a language introduced in [4] is extended to an array language. The adherence of an array language L is a set of infinite arrays such that all the finite left initial segment arrays of these infinite arrays, are the left initial segments of arrays in L. We observe that the 'closure' topology, we define here on the set of infinite arrays, using the notion of adherence, coincides with the metric topology.

During the last decade, we have been interested in proposing several grammatical models for the generation of two-dimensional arrays. We examine here the question of generation of infinite arrays. We illustrate this by constructing a Generalized CF Kolam Array Grammar (GCFKAG) [7, 11] to generate the increasing sequence of finite rectangular arrays, which describe the digitized picture patterns of the successive stages of the curve of Peano [2]. The limit of this increasing sequence of arrays, is an infinite array, which describes the infinite space-filling curve of Peano. This curve is a typical example of a function from the unit interval [0, 1] onto the unit square [0, 1]×[0, 1] which is the result of an infinite computation of a sequence of functions which describe the various stages in the construction of the Peano curve. This provides the motivation for introducing the notion of successful infinite computation or derivation of infinite arrays by array grammars. It then becomes necessary to extend the domain of finite arrays by adding the infinite arrays, which are limits of increasing sequences of finite arrays, in a complete metric topology, thus providing a typical case where infinite arrays cannot be the result of finite derivations or computations.

Out of the various array grammars, a special class of CFKAG's, called Parentheses-free CFKAG [11] is a suitable model for the generation of infinite arrays, as a result of successful infinite computations. Finally, we note that the finite array language generated by a PFCFKAG G, which is reduced and which is in Greibach form, from a nonterminal ξ and the set of infinite arrays generated by G from the same nonterminal ξ, can be related through the notion of adherence.

2. Preliminaries

In this section, we define infinite arrays as extensions of finite (rectangular) arrays.

Definition 2.1. Let X be a finite alphabet and X^{**} denote the collection of all finite (rectangular) arrays (i.e. arrays with a finite number of rows and columns) over X. $X^{++} = X^{**} - \{\Lambda\}$ (Λ is the empty array). Let

$$\alpha = \{\alpha_{ij}\}_{\substack{i=1,\dots,m\\ j=1,\dots,n}} = \begin{bmatrix} \alpha_{11} & \cdots & \alpha_{1n} \\ \vdots & & \vdots \\ \alpha_{m1} & \cdots & \alpha_{mn} \end{bmatrix}, \quad \alpha_{ij} \in X \text{ for } i = 1, \dots, m,\ j = 1, \dots, n$$

be a finite array over X with m rows and n columns. By an initial segment or left factor of α, we mean an array

$$\beta = \{\beta_{ij}\}_{\substack{i=1,\dots,p \\ j=1,\dots,q}},$$

where $p \leqslant m$ and $q \leqslant n$ and $\alpha_{ij} = \beta_{ij}$, for all $i = 1, \dots, p$ and $j = 1, \dots, q$. We write β as $\alpha[p, q]$. When $p = q$, we write the square array $\alpha[p, q]$ as $\alpha[p]$. We note that $\alpha[p]$ exists only when $p \leqslant \min(m, n)$.

Definition 2.2. An infinite array α over X is of the form

$$\{\alpha_{ij}\}_{\substack{i=1,\dots,\infty \\ j=1,\dots,\infty}},$$

$\alpha_{ij} \in X$ for all $i = 1, \dots, \infty$ and $j = 1, \dots, \infty$.

The collection of all infinite arrays over X is denoted by $X^{\omega\omega}$ and the set $X^{**} \cup X^{\omega\omega}$, of finite and infinite arrays, is denoted by $X^{\infty\infty}$.

An initial segment $\alpha[m, n]$ of an infinite array α is defined in a manner similar to the finite case. Here again, we write the square array $\alpha[m, m]$ as $\alpha[m]$.

Thus every infinite array α is the limit of an increasing sequence of finite arrays, namely

$$\alpha[1] < \alpha[2] < \cdots < \alpha[m] < \cdots.$$

3. Topology on $X^{\infty\infty}$

In this section, we first define a metric topology on $X^{\infty\infty}$ and study the properties of the resulting space.

Definition 3.1. For $\alpha, \beta \in X^{\infty\infty}$, we define the distance $d(\alpha, \beta)$ by

$$d(\alpha, \beta) = \begin{cases} 1/2^n, & n = \min\{k \mid \alpha[k] \neq \beta[k]\}, \\ 0, & \text{otherwise.} \end{cases}$$

It is easily verified that d satisfies the axioms of an ultrametric distance, namely

(i) $d(\alpha, \beta) = 0$ iff $\alpha = \beta$,

(ii) $d(\alpha, \beta) = d(\beta, \alpha)$,

(iii) $d(\alpha, \beta) \leqslant \max\{d(\alpha, \gamma), d(\gamma, \beta)\}$ for $\alpha, \beta, \gamma \in X^{\infty\infty}$.

Theorem 3.1. *$(X^{\infty\infty}, d)$ is a complete metric space.*

Proof. Let $\{\alpha_n\}$ be a Cauchy sequence in $X^{\infty\infty}$ i.e. given $\varepsilon > 0$, we can find a positive integer N such that for all $m, n \geqslant N$, we have $d(\alpha_m, \alpha_n) < \varepsilon$. For $\varepsilon = 1/2^r$, where r

is any integer $\geqslant 1$, let N_r be the minimum N such that $d(\alpha_m, \alpha_n) < 1/2^r$ for all m, $n \geqslant N$. This implies that $\alpha_{N_r}[r] = \alpha_{N_{r+1}}[r] = \cdots$.

Now we consider the increasing sequence $\alpha_{N_1}[1], \alpha_{N_2}[2], \ldots, \alpha_{N_r}[r], \ldots$. Let

$$\alpha = \lim_{r \to \infty} \alpha_{N_r}[r] \in X^{\infty\infty}.$$

Then given $\varepsilon > 0$, we can find a M such that $1/2^M < \varepsilon$ and $d(\alpha_n, \alpha) < \varepsilon$ for all $n \geqslant M$. Thus, α is the required limit and so $(X^{\infty\infty}, d)$ is complete. □

Theorem 3.2. *$(X^{\infty\infty}, d)$ is totally bounded.*

Proof. It sufficies to show that for any $\varepsilon > 0$, there exists a finite set $\{f_1, \ldots, f_n\}$ such that $X^{\infty\infty} = \bigcup_{i=1}^{n} S_\varepsilon(f_i)$ where $S_\varepsilon(f_i) = \{f \in X^{\infty\infty} \mid d(f, f_i) < \varepsilon\}$. Choose k so that $1/2^k < \varepsilon$ and consider $\alpha \in X^{\infty\infty}$. Define $\mathrm{RF}_k(\alpha) = \{\beta \in X^{\infty\infty} \mid \beta_{ij} = \alpha_{ij} \text{ for } i, j > k\}$. Let A be the collection of all finite arrays $\{\alpha_{ij}\}_{i=1,\ldots,n, j=1,\ldots,m}$ such that n, $m \leqslant k$.

Consider the set $B = \mathrm{RF}_k(\alpha) \cup A$, which is finite. Let α' be an arbitrary element of $X^{\infty\infty}$. We can always find an element α'' in B such that $\alpha'[k] = \alpha''[k]$. Hence $d(\alpha', \alpha'') < 1/2^k$ and thus $\alpha' \in S_\varepsilon(\alpha'')$. This implies that $X^{\infty\infty} = \bigcup_{\alpha_i \in B} S_\varepsilon(\alpha_i)$. Hence we get this theorem. □

Theorem 3.3. *$(X^{\infty\infty}, d)$ is compact.*

This follows from Theorems 3.1 and 3.2.

Theorem 3.4. *The sup limit of an increasing sequence of elements of $X^{\infty\infty}$ coincides with the d-limit of the sequence.*

The proof is clear from the definitions.

Definition 3.2. For any language $L \subset X^{\infty\infty}$, define the adherence of L as $\mathrm{Adh}(L) = \{\alpha \in X^{\infty\infty} \mid \text{for all } n \geqslant 1, \text{ there exists a } \beta \in L \text{ such that } \alpha[n] = \beta[n]\}$.

For $\alpha \in X^{\infty\infty}$, let $\mathrm{FG}(\alpha) = \{\alpha[n] \mid n \in \mathbb{N}\}$ and $\mathrm{FG}(L) = \bigcup_{\alpha \in L} \mathrm{FG}(\alpha)$. Hence we have

$$\mathrm{Adh}(L) = \{\alpha \in X^{\infty\infty} \mid FG(\alpha) \subset FG(L)\}.$$

Definition 3.3. For $L \subset X^{\infty\infty}$, let $\mathrm{Cl}(L) = L \cup \mathrm{Adh}(L)$.

We note the following properties of Cl.

Theorem 3.5. *For $L, L' \subset X^{\infty\infty}$,*

(1) $L \subseteq \mathrm{Cl}(L)$,
(2) $\mathrm{Cl}(\emptyset) = \emptyset$,
(3) $\mathrm{Cl}(L \cup L') = \mathrm{Cl}(L) \cup \mathrm{Cl}(L')$,
(4) $\mathrm{Cl}(\mathrm{Cl}(L)) = \mathrm{Cl}(L)$.

Proof. (1), (2) and (3) are immediate from the definition.
We prove (4):

$$\begin{aligned} \mathrm{Cl}(\mathrm{Cl}(L)) &= \mathrm{Cl}(L \cup \mathrm{Adh}(L)) \\ &= L \cup \mathrm{Adh}(L) \cup \mathrm{Adh}(L \cup \mathrm{Adh}(L)) \\ &= L \cup \mathrm{Adh}(L) = \mathrm{Cl}(L) \end{aligned}$$

since

$$\begin{aligned} (\mathrm{Adh}(L \cup \mathrm{Adh}(L)) &= \mathrm{Adh}(L) \cup \mathrm{Adh}(\mathrm{Adh}(L)) \\ &= \mathrm{Adh}(L) \cup \mathrm{Adh}(L) = \mathrm{Adh}(L). \qquad \square \end{aligned}$$

The operator $\mathrm{Cl}(L)$ generates a topology on $X^{\infty\infty}$, for which $\mathrm{Cl}(L)$ is the closure of L. It is clear that this topology coincides with the topology induced by the metric, introduced above.

4. Generation of infinite arrays

The metric topology obtained in the previous section is a general study for infinite arrays independent of any array grammar generating finite or infinite arrays. We have proposed various array models generating finite rectangular arrays in our earlier studies [6, 7, 8, 11]. We first briefly review, in an informal way, some of the fundamental definitions of the array grammars relevant to our study.

Array rewriting grammars, whose rules allow replacement of a subarray of a picture by another subarray, have been extensively investigated in the literature [5].

Two-dimensional matrix grammars have been proposed in [6] to describe digitized rectangular arrays. These matrix grammars consist of two phases of derivations. The first phase generates (horizontal) strings of intermediate symbols using a Chomskian PS, CS, CF or Regular grammar. Each intermediate symbol is the start letter of a right-linear grammar. During the second phase of derivations the nonterminal rules of these right-linear grammars are applied in parallel in the vertical direction, rewriting a horizontal string of intermediates generated in the first phase. The application of these rules in the vertical direction can be continued or the derivation can be terminated by simultaneous application of terminal rules of these right-linear grammars, thus obtaining rectangular arrays.

Another class of array grammars, called Kolam Array Grammars (KAG), more powerful in generative power than the Matrix Grammars [6], have been introduced in [7, 11], generalizing the notion of rewriting rules in string grammars to array rewriting rules in which the catenation of strings is extended to row and column catenations of arrays. A KAG consists of two types of rules: The first type consists of a finite set of horizontal or vertical, CS, CF or Regular rules involving only the nonterminals and intermediates (treating the intermediates as terminals). The second type consists of rules which generate intermediate languages – one corresponding to each intermediate. These intermediate languages may be CS, CF or Regular over arrays consisting of a fixed number of rows or a fixed number of columns of elements from the set of terminals. Instead of enumerating the rules, the intermediate language corresponding to each intermediate is usually given. Derivations proceed in two phases. During the first phase, derivations proceed making use of the nonterminal and terminal rules, introducing parentheses at every stage to avoid ambiguity due to lack of associativity of the column and row operators, till all the nonterminals are replaced. The resultant of the first phase will consist of strings of intermediates catenated together with row and column catenation operators and with parentheses suitably introduced. During the second phase of derivations, starting from the innermost parentheses, each intermediate is replaced by the corresponding intermediate language subject to the conditions of row and column catenations. Once all the intermediates are replaced, we arrive at the rectangular array of terminals.

A more general family of array grammars, called Generalized CF Kolam Array Grammars, is proposed in [11]. This family is obtained by introducing parentheses in the productions of the first phase of a KAG and allowing productions either to have parentheses or not to have parentheses, instead of parenthesizing the sequence of applications of the rules in the derivations as is done in a KAG. This new class is interesting since we are able to generate new picture classes not generable by earlier models and the generative power of the grammar is increased. There are two special classes of these grammars – the class of Parentheses CF KAG (PCFKAG) in which the right side of every rule in the first phase is enclosed within a pair of parentheses and the class of Parentheses-free CF KAG (PFCFKAG) in which the right side of no rule in the first phase is enclosed within a pair of parentheses. These two classes are properly contained within the class of Generalized CF Kolam Array Languages.

We now examine the question of generation of infinite arrays, making use of these array grammars. Infinite arrays, which are limits of increasing sequences of finite rectangular arrays, can be obtained as the result of successful infinite derivations of Generalized CF Kolam Array Grammars or its subclasses. In fact, given a GCFKAG G, if the finite array language $L(G)$ consists of an increasing sequence of arrays

$$M_1 < M_2 < \cdots < M_n < \cdots$$

yielding an infinite array M as its limit, then the limit language of G can be defined to consist of such arrays M.

We illustrate this notion by providing a Generalized CF KAG generating the digitized geometric patterns of the curves of the successive stages in the construction of the space-filling curve of Peano [2].

Example 4.1. Consider the Generalized CF KAG $G = (N, I, T, P, S_1, \mathscr{S})$ where $N = \{S_i, X_i \mid 1 \leq i \leq 6\}$, $I = \{A_i \mid 1 \leq i \leq 6\}$, $T = \{b, c, \bar{c}, d, \bar{d}\}$. P consists of the rules

$$S_1 \to (X_1 \obar X_2 \obar X_3), \qquad S_2 \to (X_4 \obar X_5 \obar X_6),$$

$$S_3 \to (X_1 \obar X_2 \obar X_5), \qquad S_4 \to (X_2 \obar X_5 \obar X_6),$$

$$S_5 \to (X_4 \obar X_5 \obar X_2), \qquad S_6 \to (X_5 \obar X_2 \obar X_3),$$

$$X_1 \to (S_3 \ominus S_2 \ominus S_1), \qquad X_2 \to (S_4 \ominus S_1 \ominus S_5),$$

$$X_3 \to (S_1 \ominus S_2 \ominus S_6), \qquad X_4 \to (S_2 \ominus S_1 \ominus S_5),$$

$$X_5 \to (S_3 \ominus S_2 \ominus S_6), \qquad X_6 \to (S_4 \ominus S_1 \ominus S_2),$$

$$S_i \to (A_i), \quad i = 1, \ldots, 6,$$

$$\mathscr{L} = \{L_{A_i} \mid i = 1, \ldots, 6\},$$

$$L_{A_1} = \begin{Bmatrix} c & \bar{c} & b \\ b & b & b \\ b & d & \bar{d} \end{Bmatrix}, \qquad L_{A_2} = \begin{Bmatrix} b & c & \bar{c} \\ b & b & b \\ d & \bar{d} & b \end{Bmatrix}, \qquad L_{A_3} = \begin{Bmatrix} c & \bar{c} & c \\ b & b & b \\ b & d & \bar{d} \end{Bmatrix},$$

$$L_{A_4} = \begin{Bmatrix} \bar{c} & c & \bar{c} \\ b & b & b \\ d & \bar{d} & b \end{Bmatrix}, \qquad L_{A_5} = \begin{Bmatrix} b & c & \bar{c} \\ b & b & b \\ d & \bar{d} & d \end{Bmatrix}, \qquad L_{A_6} = \begin{Bmatrix} c & \bar{c} & b \\ b & b & b \\ \bar{d} & d & \bar{d} \end{Bmatrix}.$$

We note that G is, in fact, a Parentheses KAG, since the right side of every rule is enclosed in parentheses. It can be seen that G generates a sequence $\{M_n \mid n \geq 1\}$ of finite rectangular arrays, which describe the digitized patterns of the successive stages of the curve of Peano, so that $M_1 < M_2 < \cdots < M_n < \cdots$. We have given in Fig. 1, the first two arrays M_1 and M_2. By giving suitable instructions, as done in Narasimhan's method of kolam generation [8], we can obtain from M_1 and M_2, the corresponding curve patterns of the first two stages of the Peano curve (Fig. 2).

$$\begin{matrix} c & \bar{c} & b \\ b & b & b \\ b & d & \bar{d} \end{matrix} \qquad\qquad \begin{matrix} c & \bar{c} & c & \bar{c} & c & \bar{c} & c & \bar{c} & b \\ b & b & b & b & b & b & b & b & b \\ b & d & \bar{d} & d & \bar{d} & b & b & d & \bar{d} \\ b & c & \bar{c} & c & \bar{c} & b & b & c & \bar{c} \\ b & b & b & b & b & b & b & b & b \\ d & \bar{d} & b & b & d & \bar{d} & d & \bar{d} & b \\ c & \bar{c} & b & b & c & \bar{c} & c & \bar{c} & b \\ b & b & b & b & b & b & b & b & b \\ b & d & \bar{d} & d & \bar{d} & d & \bar{d} & d & \bar{d} \end{matrix}$$

Fig. 1. Arrays M_1 and M_2 of Example 4.1.

The limit of this increasing sequence M_n of arrays, is an infinite array and corresponds to the digitized geometric pattern of the space-filling curve of Peano.

The Peano curve is a continuous, nowhere differentiable function from the unit interval $[0, 1]$ onto the unit square $[0, 1] \times [0, 1]$, with its range, a subset of R_2. This curve is a typical example of a function which is the result of an infinite computation of a sequence of functions $\{f_n\}$, which converges to a mapping f from

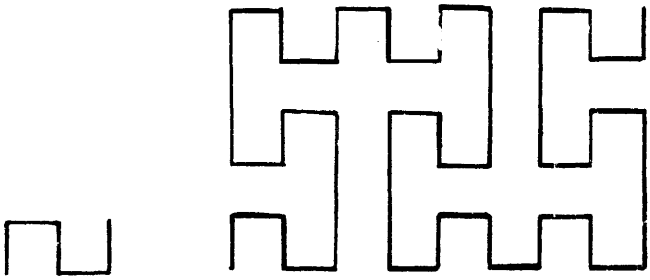

Fig. 2. The first two stages in the construction of the Peano curve.

the unit interval onto the unit square. The curves of the first two stages in the construction of the Peano curve are given in Fig. 2. The curve C_n of the nth stage is constructed from its predecessor C_{n-1}, by dividing the unit square into nine equal parts, drawing the curve C_{n-1}, possibly rotated or reflected in each and joining them up in the sense of C_1.

The above example is of interest, since it provides the motivation for extending finite arrays to infinite arrays and to obtain the function as a result of an infinite computation.

We now consider another example of a Generalized CFKAG generating a finite array language, the first three members of which are shown in Fig. 3.

$$\begin{array}{ccc} x & x & x \\ x & \bullet & x \\ x & x & x \end{array} \qquad \begin{array}{cccc} x & x & x & x \\ x & \bullet & \bullet & x \\ x & \bullet & \bullet & x \\ x & x & x & x \end{array} \qquad \begin{array}{ccccc} x & x & x & x & x \\ x & \bullet & \bullet & \bullet & x \\ x & \bullet & \bullet & \bullet & x \\ x & \bullet & \bullet & \bullet & x \\ x & x & x & x & x \end{array}$$

Fig. 3. Arrays of Example 4.2.

Example 4.2. Let $G=(N, I, T, P, S, \mathcal{I})$ be a GCFKAG, where $N=\{S, S_1\}$, $I=\{A, B, C, D, E\}$, $T=\{\cdot, X\}$, $P=\{S \rightarrow A \oplus S_1, S_1 \rightarrow B \ominus C \oplus S_1, S_1 \rightarrow D \ominus E\}$ and $\mathcal{I}=\{L_A, L_B, L_C, L_D, L_E\}$ with

$$L_A=\left\{\begin{matrix} XX \\ X\cdot \end{matrix}\right\}, \qquad L_B=\left\{\begin{matrix} X \\ (\cdot)_n \end{matrix} \,\middle|\, n \geq 1\right\}, \qquad L_C=\{X(\cdot)^n \mid n \geq 1\},$$

$$L_D=\{X_n \mid n \geq 1\} \quad \text{and} \quad L_E=\{X^n \mid n \geq 1\}.$$

We note that $L(G)$ does not contain any increasing sequence of arrays and so the limit language of $L(G)$ is empty.

On the other hand, if we modify the notion of successful infinite computation of arrays by a GCFKAG by requiring in the first phase of the grammar infinite derivations of the form

$$\alpha_1 \Rightarrow \alpha_2 \Rightarrow \cdots \Rightarrow \alpha_n \Rightarrow \cdots$$

where α_1 is the start symbol, such that

$$\mathrm{LF}(\alpha_1)<\mathrm{LF}(\alpha_2)<\cdots<\mathrm{LF}(\alpha_n)<\cdots$$

where $\mathrm{LF}(\alpha_n)$ is the largest left factor of α_n over intermediate symbols, and in the second phase, each $\mathrm{LF}(\alpha_n)$ to yield a finite array M_n so that

$$M_1 < M_2 < \cdots < M_n < \cdots,$$

then the limit language will consist of the limit of this increasing sequence of arrays. As an illustration, in Example 4.2, we note that the limit language is empty according to the earlier notion of successful infinite computation whereas the limit language consists of an infinite array in the modified notion, namely the limit of the sequence of arrays

$$M'_1 < M'_2 < \cdots < M'_n < \cdots$$

where M'_1, M'_2 and M'_3 are shown in Fig. 4.

$$\begin{array}{cc} x & x \\ x & \bullet \end{array} \qquad \begin{array}{ccc} x & x & x \\ x & \bullet & \bullet \\ x & \bullet & \bullet \end{array} \qquad \begin{array}{cccc} x & x & x & x \\ x & \bullet & \bullet & \bullet \\ x & \bullet & \bullet & \bullet \\ x & \bullet & \bullet & \bullet \end{array}$$

Fig. 4. Arrays M'_1, M'_2, M'_3 of Example 4.2.

We note that this modification in the definition of infinite successful computation, for this particular example is possible since the grammar is parentheses-free.

In the Generalized CF Kolam Array Grammars [11], replacements start from the innermost parentheses and the sequence of arrays is obtained in such a way that each array may be built up by catenating arrays on all four sides of the previous array. In such a case, it is difficult to establish the criterion for generating a sequence of ascending chain of arrays. Hence we have singled out the class of Parentheses-free CF Kolam Array Grammars, which is a special class of the Generalized CF Kolam Array Grammars. By its very nature of 'left-most replacement' in the second phase and the fact that the rules are parentheses-free in the first phase, the class is well-suited for defining the notion of successful infinite computation. Furthermore, for this class, the array language generated by a reduced grammar G in Greibach normal form and the set of infinite arrays resulting from successful infinite computations can be related through the notion of adherence.

We review the definition of a Parentheses-free CFKAG in order to define the notion of infinite derivations.

Definition 4.1. A Parentheses-free, Context-free Kolam Array Grammar (PFCFKAG) is $G = (\mathcal{N}, I, X, P, \xi_1, \mathcal{L})$ where $\mathcal{N}$, I, X are finite nonempty sets of symbols, called nonterminals, intermediates and terminals, respectively. $\xi_1 \in \mathcal{N}$ is the start symbol. P is a finite nonempty set of productions of the form

$$\xi_i \to \eta_1 \oplus \cdots \oplus \eta_k, \quad k \geq 1, \ \xi_i \in \mathcal{N}, \ \eta_i \in (\mathcal{N} \cup I), \ \oplus \in \{\text{⦶}, \ominus\}.$$

For each intermediate A in I, L_A is an intermediate array language, which is regular, CF or CS and whose terminals are arrays with a fixed number of rows or columns of symbols of X. $\mathcal{L} = \{L_A \mid A \in I\}$.

Infinite derivations are defined as follows: Starting with ξ_1, productions of P are applied as in string grammars so that an infinite sequence $f_1 = \xi_1, f_2, \ldots, f_n, \ldots$ such that $f_n \Rightarrow f_{n+1}$ for $n = 1, 2, \ldots$ is obtained. Furthermore, if $\mathrm{LF}(f_n)$ is the longest left factor of f_n over $I \cup \{\textcircled{\scriptsize |}, \textcircled{=}\}$, then $f_n \Rightarrow f_{n+1}$ implies $\mathrm{LF}(f_n) \leqslant \mathrm{LF}(f_{n+1})$.

Hence, if $f_1 = \xi_1, f_2, \ldots, f_n, \ldots$ is an infinite derivation in G, then the sequence

$$\mathrm{LF}(f_1) \leqslant \mathrm{LF}(f_2) \leqslant \cdots \leqslant \mathrm{LF}(f_n) \leqslant \cdots$$

is an infinite increasing sequence of words over $I \cup \{\textcircled{\scriptsize |}, \textcircled{=}\}$.

Let $\mathrm{LF}(f_1) = A_1 \oplus \cdots \oplus A_{k_1}$; $A_i \in I$ for $i = 1, 2, \ldots, k_1$; $\oplus \in \{\textcircled{\scriptsize |}, \textcircled{=}\}$ and for $n > 1$,

$$\mathrm{LF}(f_n) = \mathrm{LF}(f_{n-1}) \oplus A_{k_{n-1}+1} \oplus \cdots \oplus A_{k_n}.$$

We now replace intermediates in $\mathrm{LF}(f_n)$ $(n \geqslant 1)$ by terminal arrays as follows: First the intermediate A_1 in $\mathrm{LF}(f_1)$ is replaced by an array M_1, chosen from the intermediate array language L_{A_1}, A_2 is replaced by an array M_2 chosen from the intermediate array language L_{A_2} and is column or row catenated to M_1 according as the catentation operator $\oplus$ between A_1 and A_2 is $\textcircled{\scriptsize |}$ or $\textcircled{=}$ symbol; A_3 is then replaced by an array M_3 chosen from the intermediate array language L_{A_3} and is column or row catenated to $M_1 \oplus M_2$ according as the catenation operator $\oplus$ between A_2 and A_3 is $\textcircled{\scriptsize |}$ or $\textcircled{=}$ symbol and so on. Finally, A_{k_1} is replaced by a terminal array M_{k_1}, thus yielding a finite rectangular array α_1 over X. (The replacements are subject to the conditions for row and column catenations.) For $n > 1$, if $\mathrm{LF}(f_{n-1})$ yields the finite rectangular array α_{n-1}, over X, $\mathrm{LF}(f_n)$ yields the finite rectangular array α_n over X, as follows: $\mathrm{LF}(f_{n-1})$ is replaced by α_{n-1}; $A_{k_{n-1}+1}$ is replaced by an array $M_{k_{n-1}+1}$ chosen from $L_{A_{k_{n-1}+1}}$ and is catenated to α_{n-1}; $A_{k_{n-1}+2}$ is replaced by an array $M_{k_{n-1}+2}$ chosen from $L_{A_{k_{n-1}+2}}$ and catenated to $\alpha_{n-1} \oplus M_{k_{n-1}+1}$ and so on. Finally, A_{k_n} is replaced to yield the finite array α_n. We thus obtain an infinite, increasing sequence of arrays, $\alpha_1 \leqslant \alpha_2 \leqslant \cdots \leqslant \alpha_n \leqslant \cdots$. We say that the derivation $f_1, f_2, \ldots, f_n, \ldots$ is successful iff the sequence $\alpha_1, \alpha_2, \ldots, \alpha_n, \ldots$ has a lub α in $X^{\omega\omega}$. We write $\xi_1 \Rightarrow_G^{\omega} \alpha$.

The set of all infinite arrays generated by G from ξ_1, is the Parentheses-free Context-free ω-Kolam Array Language

$$L_{\omega\omega}(G, \xi_1) = \{\alpha \in X^{\omega\omega} \mid \xi_1 \Rightarrow_G^{\omega} \alpha\}.$$

Definition 4.2. A PFCFKAG G is in Greibach Normal form if and only if every rule in P is of the form $\xi_{i_1} \to A \oplus \xi_{i_2} \oplus \cdots \oplus \xi_{i_k}$; $A \in I$, $\xi_{i_j} \in \mathcal{N}$, $j = 1, \ldots, k$ $(\geqslant 1)$ and reduced if and only if $L(G, \xi_i) \neq \emptyset$ for $\xi_{i_j} \in \mathcal{N}$.

Theorem 4.1. *If a reduced* PFCFKAG G *is in Greibach Normal form, then for all* i,

$$L_{\omega\omega}(G, \xi_i) = \mathrm{Adh}[L(G, \xi_i)].$$

The proof of this theorem is on lines similar to the proof of Theorem [illegible] in [4] and is therefore omitted.

5. Conclusion

The study in this paper centers around kolam array grammars extended to generate infinite arrays. We note that the concept of infinite successful computation defined for Generalized KAG reflects the idea of limit language as introduced in [1], whereas the modified notion for Parentheses-free KAG is closer to the concept of infinite successful derivation of ω-words and ω-languages considered in Nivat [4].

Extension to L-systems is a natural generalization which will involve parallel rewriting. One of the models which is useful for such a study of infinite arrays, is the deterministic L-array grammars [9]. Here rewriting is done in parallel along the edges and if we impose the restriction that growth takes place only along two adjacent sides, then we get an increasing sequence of arrays, the resulting limiting array being an infinite array [10].

Acknowledgment

The authors thank the referees for their useful comments.

References

[1] K. Culik II and A. Salomaa, On infinite words obtained by iterating morphisms, *Theoret. Comput. Sci.* **19** (1982) 29–38.
[2] S. Eilenberg, *Automata, Languages and Machines: Vol. A* (Academic Press, New York, 1974).
[3] A. Nakamura and H. Ono, Pictures of functions and their acceptability by automata, *Theoret. Comput. Sci.* **23** (1983) 37–48.
[4] M. Nivat, Infinite words, infinite trees and infinite computations, Mathematical Centre Tracts **109** (1979) 1–52.
[5] A. Rosenfeld, *Picture Languages* (Academic Press, New York, 1979).
[6] G. Siromoney, R. Siromoney and K. Krithivasan, Abstract families of matrices and picture languages, *Comput. Graph. Image Processing* **1** (1972) 284–307.
[7] G. Siromoney, R. Siromoney and K. Krithivasan, Picture languages with array rewriting rules, *Information and Control* **22** (1973) 447–470.
[8] G. Siromoney, R. Siromoney and K. Krithivasan, Array grammars and Kolam, *Comput. Graph. Image Processing* **3** (1974) 63–82.
[9] R. Siromoney and G. Siromoney, Extended controlled table L-arrays, *Information and Control* **35** (1977) 119–138.
[10] R. Siromoney, V. R. Dare and K.G. Subramanian, On infinite words obtained by deterministic controlled table L-array systems. Presented at the Second Conference on FST and TCS, India (1982).
[11] K.G. Subramanian, Studies on array languages, Ph.D. Dissertation, Madras University (1979).