International Journal of Advances in Computer and Electronics Engineering
Volume: 01 Issue: 03, July 2016, pp. 42 – 46

Superimposition of Infinite Words

[1]C. Annal Deva Priya Darshini, [2]Dr. V. Rajkumar Dare
[1]Assistant Professor, Department of Mathematics, Madras Christian College, Chennai, India
[2]Associate Professor, Department of Mathematics, Madras Christian College, Chennai, India
[1]annaldevapriyadarshini@gmail.com, [2]rajkumardare@yahoo.com

Abstract: *Motivated by the process of picture composition in Astronomy and Printing a new operation on words called the superimposition operation is introduced. In this paper, the operation is extended to infinite words. A variant of this operation where the superimposition of two words is restricted using a control language is also introduced in this paper. The study focuses on the algebraic aspects of the operation. This operation induces a group structure on the set of all infinite words over a finite alphabet. The commutativity and associativity properties of the control languages with respect to the operation are also examined.*

Keyword: *Please Infinite words, Control languages, Superimposition; Superimposition under control;*

1. INTRODUCTION

Combinatorics on words focuses on properties of words and patterns on words. The applications in Computer Science are many as any algorithm is a stringover a finite alphabet. The theory of formal languages was developed taking inspiration from problems in Computer science. The notion of a word is particularly useful in Mathematics as the representation of a natural number with any base is a word over a finite alphabet. The infinite word is a representation of a real number. Problems on words appear in different contexts in pure algebra. The set Σ^* of all words over the alphabet Σ is called the free monoid over the set Σ. Several generalizations of the catenation operation like shuffle operation, insertion and deletion operations, shuffle on trajectories were introduced and studied by many authors in [8, 10]. The shuffle operation is useful in modelling parallel composition of words and languages. Theoretical generalization to the case of infinite words was studied in detail by Kadrie et al. in [1]. Another operation is the Collage operation on words [6].In this paper, a new operation on words and languages is introduced motivated by different real life problems [2].

A colour image on a television screen or computer monitor is generated using two main sets of colours called colour spaces - the RGB colour space consisting the primary stimuli for human colour perception and CMYK colour space – Cyan, Magenta, Yellow and Black. A printed colour picture is create with the aid of the CMYK colour space. All other colours are combinations of the basic colours in varying proportions.

The superimposition operation on finite words is introduced to model this phenomenon as an operation on words and languages. This paper is an extension of the operation to the infinite words.

The theory of infinite words is more complex than finite words. The cardinality of the set of all infinite words is non – denumerable while that of all finite words is only denumerable.

For any alphabet Σ, an ω – word over Σ is a function f: $N \rightarrow \Sigma$ denoted by the infinite sequence $f(0)f(1)f(2)$.... An ω - word w is ultimately periodic if $w = \alpha v^{\omega}$ where $\alpha$is a finite word which may be empty. An ω - word is periodic if $w = v^{\omega}$for some non-empty word $v$inΣ^*. The set of all ω - words over Σ is Σ^{ω}. An ω - language is a subset L of Σ^{ω}. The set of all words and ω -words over the alphabet Σ is denoted by $\Sigma^{\infty} = \Sigma^* \cup \Sigma^{\omega}$. The basic notations and relevant background material is from the papers [3, 4, 5, 7, 9].

2. SUPERIMPOSITION OF INFINITE WORDS

The superimposition operation was introduced in [2].The superimposition operation is a rule wherein two letters of the alphabet coalesce to give another letter of the same alphabet. For $a, b \in \Sigma$, $a \otimes b = c$ where c is also in Σ is called the superimposition operation.
This operation can be extended to Σ^*.

Definition 2.1

Let $u, v \in \Sigma^*$ and $u = u_1u_2\ldots u_n$, $v = v_1v_2\ldots v_m$ where $u_i, v_j \in \Sigma$, for $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$.

(i) If $|u| = |v|$, then $u \otimes v = (u_1 \otimes v_1)(u_2 \otimes v_2) \ldots (u_n \otimes v_n)$

(ii) If $|u| > |v|$, and if $u = u'u''$ where $|u'| = |v| = m$, then $u \otimes v = (u_1 \otimes v_1)(u_2 \otimes v_2) \ldots (u_m \otimes v_m).u''$.

(iii) If $|u| < |v|$, and if $v = v'v''$ where $|v'| = |u| = n$, then $u \otimes v = (u_1 \otimes v_1)(u_2 \otimes v_2) \ldots (u_n \otimes v_n).v''$

(iv) For $u \in \Sigma^*$, $u \otimes \lambda = \lambda \otimes u = u$.

International Journal of Advances in Computer and Electronics Engineering
Volume: 01 Issue: 03, July 2016, pp. 42 – 46

Example 2.1

Let Σ be the alphabet consisting of the three colours black, white and gray denoted by *{B, W, G}* and the operation $\otimes$ given by the following *table 1*.

Let $x = B^5W^3$, $y = W^3G^3B^3$ be words in Σ^* then $x \otimes y = B^5GB^3$.

$\otimes$	b	w	G
b	b	b	B
w	b	w	G
g	b	g	g

Table 1

This notion of superimposition is extended to the set of all ω - words.

Definition 2.2

Let Σ be a finite alphabet. For any *a, b, c* in Σ, a rule of superimposition is given by $a \otimes b = c$.

Let $u = u_1u_2u_3\ldots, v = v_1v_2v_3\ldots \in \Sigma^\omega$ where $u_i, v_i \in \Sigma$ for all $i = 1, 2, \ldots$. Then the superimposition of *u* and *v* with respect to the rule $\otimes$ is

$$u \otimes v = (u_1 \otimes v_1)(u_2 \otimes v_2)\ldots = \prod_{k=1}^{\infty}(u_k \otimes v_k)$$.

The result of superimposition is a unique word in Σ^ω.

The following example illustrates the definition. The length restriction present in the finite case is no more necessary.

Example 2.2

Let Σ be the alphabet *{a, b, c, d, e, f, g}* and the rule of superimposition given by the *table 2*. Let $x = (abcdefg)^\omega, y = (gfedcba)^\omega$ be words in Σ^ω. Then $x \otimes y = g^\omega$.

$\otimes$	a	b	c	d	e	f	g
a	a	b	c	d	e	f	g
b	b	c	d	e	f	g	a
c	c	d	e	f	g	a	b
d	d	e	f	g	a	b	c
e	e	f	g	a	b	c	d
f	f	g	a	b	c	d	e
g	g	a	b	c	d	e	f

Table 2

The operation can be extended to languages in a natural way.

Definition 2.3

For two languages $L_1, L_2 \subseteq \Sigma^\omega, L_1 \otimes L_2 = \bigcup_{k=1}^{\infty}(u_k \otimes v_k), u_k \in L_1, v_k \in L_2$ or $L_1 \otimes L_2 = \{z / z = x \otimes y; x \in L_1, y \in L_2\}$.

Note that the superimposition of two infinite words is always non – empty. The Superimposition of a language on itself infinite number of times gives the ω – language.

For a given language L, $L_\otimes^n = L \otimes L \otimes L..ntimes$. Then the star closure of a language is $L_\otimes^* = L_\otimes^0 \cup L_\otimes^1 \cup L_\otimes^2.....$ and the positive closure is $L_\otimes^+ = L_\otimes^1 \cup L_\otimes^2.....$ for a given superimposition operation.

Definition 2.4

The superimposition of a language on itself infinite nume=ber of times is the limit of a finite language $L_\otimes^\omega = \lim_{n\to\infty} L_\otimes^n$.

The properties of finite languages can be extended to the infinite case.

Proposition 2.1

For any language $L \subseteq \Sigma^\omega$ the following hold:

(i) $L \otimes \{\lambda\} = \{\lambda\} \otimes L = L$

(ii) $L \otimes \phi = \phi \otimes L = L$

(iii) $L \otimes L_\otimes^\omega = L_\otimes^\omega \otimes L$

(iv) $(L_\otimes^\omega)^\omega = L_\otimes^\omega$

Proposition 2.2

For languages $L_1, L_2, M_1, M_2 \subseteq \Sigma^\omega$ the superimposition operation preserves the subset relation on sets. For

$L_1 \subseteq L_2, M_1 \subseteq M_2$

implies $L_1 \otimes M_1 \subseteq L_2 \otimes M_2$

Proof:

Let $x \in L_1 \otimes M_1$ then $x = y \otimes z$ for $y \in L_1$ and $z \in M_1$. Now, $y \in L_1 \subseteq L_2$ implies *y* is in L_2. Also, $z \in M_1 \subseteq M_2$ implies *z* is in M_2. Therefore, $y \otimes z$ is in $L_2 \otimes M_2$. Thus, $L_1 \otimes M_1 \subseteq L_2 \otimes M_2$

Proposition 2.3

Let $L_1, L_2, L_3 \subseteq \Sigma^\omega$ be any three languages and $\otimes$ a superimposition operation defined on the same alphabet. Then the following relations hold:

i. $(L_1 \cup L_2) \otimes L_3 = (L_1 \otimes L_3) \cup (L_2 \otimes L_3)$

ii. $L_1 \otimes (L_2 \cup L_3) = (L_1 \otimes L_2) \cup (L_1 \otimes L_3)$

iii. $(L_1 \cap L_2) \otimes L_3 = (L_1 \otimes L_3) \cap (L_2 \otimes L_3)$

iv. $L_1 \otimes (L_2 \cap L_3) = (L_1 \otimes L_2) \cap (L_1 \otimes L_3)$

Proof:

International Journal of Advances in Computer and Electronics Engineering
Volume: 01 Issue: 03, July 2016, pp. 42 – 46

(i) Let $x \in (L_1 \otimes L_3) \cup (L_2 \otimes L_3)$. Then $x = p \otimes q$ or $x = p' \otimes q'$ where $p \in L_1, q, q' \in L_3, p' \in L_2$. This implies that $x = p \otimes q$ where $p \in L_1 \cup L_2, q \in L_3$ or $x = p' \otimes q'$ where $p' \in L_1 \cup L_2, q' \in L_3$ which implies that $x \in (L_1 \cup L_2) \otimes L_3$. Since x is arbitrary, we conclude that $(L_1 \otimes L_3) \cup (L_2 \otimes L_3) \subseteq (L_1 \cup L_2) \otimes L_3$. Similarly, let $x \in (L_1 \cup L_2) \otimes L_3$. Then $x = p \otimes q$ where $p \in L_1 \cup L_2, q \in L_3$. We have $x = p \otimes q$ where $p \in L_1, q \in L_3$ or $x = p \otimes q$ where $p \in L_2, q \in L_3$. Hence $x \in (L_1 \otimes L_3) \cup (L_2 \otimes L_3)$. Since x is arbitrary, we conclude that $(L_1 \cup L_2) \otimes L_3 \subseteq (L_1 \otimes L_3) \cup (L_2 \otimes L_3)$.

(ii) If $x \in L_1 \otimes (L_2 \cup L_3)$ then $x = p \otimes q$ where $p \in L_1$ and $q \in L_2$ or $q \in L_3$. Hence, $x \in L_1 \otimes L_2$ or $x \in L_1 \otimes L_3$. Therefore, we have $x \in (L_1 \otimes L_2) \cup (L_1 \otimes L_3)$. Since x is arbitrary, we conclude that $L_1 \otimes (L_2 \cup L_3) \subseteq (L_1 \otimes L_2) \cup (L_1 \otimes L_3)$. The other inclusion can be proved in a similar way.

In a similar manner, (iii) and (iv) can be proved.

Thus, the superimposition operation distributes over union and intersection both on the right and the left.

An infinite word can be generated as the limit of an increasing sequence of finite words.

Let $x \in \Sigma^\omega$. Let $x[i] = x_1 x_2 ... x_i$ be the prefix of x of length i. Then x is the limit of the finite sequence of prefixes $x[i]$. This is denoted by $x = \lim_{i \to \infty} x[i]$.

The prefix ordering on Σ^* is $<_p : \Sigma^* \to \Sigma^*$ where $x <_p y$ if x is a prefix of y. It is obvious that $x[1] <_p x[2] <_p x[3] ...$ for $x \in \Sigma^\omega$.

Let $x, y, z \in \Sigma^\omega$. Let $z = x \otimes y$.

Then $z[i] = z_1 z_2 ... z_i$ where $z_i = x_i \otimes y_i$ for all i.

We have

$$z[i] = (x_1 x_2 ... x_i) \otimes (y_1 y_2 ... y_i) = x[i] \otimes y[i]$$

So that $z = \lim_{i \to \infty} z[i] = \lim_{i \to \infty} \{x[i] \otimes y[i]\}$.

The limit of a word can be extended to languages.

The limit of a language $L \subset \Sigma^*$ denoted by $\lim L \in \Sigma^\omega$ is $\lim L = \{z / z = \lim x : x \in L\}$.

Proposition 2.4

Let $L_1, L_2 \subset \Sigma^*$. Then, $\lim(L_1 \otimes L_2) = \lim L_1 \otimes \lim L_2$.

Proof:

Let $z \in \lim(L_1 \otimes L_2)$. Then $z = \lim_{i \to \infty} z[i]$ where $z[i] \in L_1 \otimes L_2$.

This implies that $\lim_{i \to \infty} z[i] = \lim_{i \to \infty} \{x[i] \otimes y[i]\}$.

But we have $\lim_{i \to \infty} \{x[i] \otimes y[i]\} = \lim_{i \to \infty} x[i] \otimes \lim_{i \to \infty} y[i]$
$= x \otimes y \in \lim L_1 \otimes \lim L_2$ ……………………..(1)

Conversely, let $z \in \lim L_1 \otimes \lim L_2$.

Then $z = x \otimes y$; where $x \in \lim L_1, y \in \lim L_2$.

This implies $x = \lim_{i \to \infty} x[i]$ and $y = \lim_{i \to \infty} y[i]$.

Then $x \otimes y = \lim_{i \to \infty} x[i] \otimes \lim_{i \to \infty} y[i]$
$= \lim_{i \to \infty} \{x[i] \otimes y[i]\} \in \lim_{i \to \infty} (L_1 \otimes L_2)$ ……………..(2)

From equations (1) and (2) equality follows.

Theorem 2.3.1

If in the superimposition table for the operation $\otimes$, every row and column is a permutation of the letters of Σ, then (Σ^ω, $\otimes$) is a group.

The superimposition operation is a binary operation on words like the catenation operation. However, the catenation operation induces a monoid structure on the set of all words. The power of the superimposition lies in the group structure that it gives to Σ^n and Σ^ω. This structure is characterised and studied in this section.

A relation on Σ^ω based on the superimposition operation can be introduced.

Definition 2.5

Let $x, y \in \Sigma^\omega$. Then $x \leq_\otimes y$ if $y = x \otimes z$ for $z \in \Sigma^\omega$.

Proposition 2.5

The relation $\leq_\otimes$ is transitive if the binary operation $\otimes$ is associative,

Proof:

Let $a \leq_\otimes b$ and $b \leq_\otimes c$ for $a, b, c \in \Sigma^\omega$. We have $b \leq_\otimes c$ if $c = b \otimes t = (a \otimes z) \otimes t$ since $a \leq_\otimes b$. Therefore, $c = a \otimes (z \otimes t)$ if $\otimes$ is associative. Let $z \otimes t = t'$ then $c = a \otimes t'$ which implies $a \leq_\otimes c$.

International Journal of Advances in Computer and Electronics Engineering
Volume: 01 Issue: 03, July 2016, pp. 42 – 46

Therefore, $\leq_{\otimes}$ is transitive if $\otimes$ is associative.

Definition 2.6

For $L \subseteq \Sigma^{\omega}$, $UC_{\otimes}(L) = \{y \in \Sigma^{\omega} / x \leq_i y, x \in L\}$ and $LC_{\otimes}(L) = \{x \in \Sigma^{\omega} / x \leq_i y, y \in L\}$.

Proposition 2.6

$UC_{\otimes}(L)$ and $LC_{\otimes}(L)$ are ω - regular languages if *L* is ω - regular.

Definition 2.7

A language *L* is upper – closed with respect to a superimposition operation $\otimes$ if $UC_{\otimes}(L) = L$ and a language is lower – closed if $LC_{\otimes}(L) = L$.

Proposition 2.7

Let $\otimes$ be a commutative and associative relation on *Σ**. Let $L_1, L_2 \subseteq \Sigma^{\omega}$ Then,

(i) $UC_{\otimes}(L_1 \otimes L_2) = UC_{\otimes}(L_1) \otimes UC_{\otimes}(L_2)$

(ii) $LC_{\otimes}(L_1 \otimes L_2) = LC_{\otimes}(L_1) \otimes LC_{\otimes}(L_2)$.

Proof: Let $y \in UC_{\otimes}(L_1 \otimes L_2)$. Then $x \leq_{\otimes} y$ for $x \in L_1 \otimes L_2$ so that $x = \alpha_1 \otimes \alpha_2$ for some $\alpha_1 \in L_1, \alpha_2 \in L_2$ and $y = x \otimes z$ for some $z \in \Sigma^{\omega}$. Now $y = (\alpha_1 \otimes \alpha_2) \otimes z = \alpha_1 \otimes (\alpha_2 \otimes z)$ since $\otimes$ is associative. But $\alpha_1 \in L_1 \subseteq UC_{\otimes}(L_1)$ and $\alpha_2 \in L_2 \subseteq UC_{\otimes}(L_2)$ so that $y \in UC_{\otimes}(L_1) \otimes UC_{\otimes}(L_2)$ implies $UC_{\otimes}(L_1 \otimes L_2) \subseteq UC_{\otimes}(L_1) \otimes UC_{\otimes}(L_2)$. To prove the reverse inclusion, we have $y \in UC_{\otimes}(L_1) \otimes UC_{\otimes}(L_2)$ then $y = \alpha_1 \otimes \alpha_2$ where $\alpha_1 \in UC_{\otimes}(L_1)$ and $\alpha_2 \in UC_{\otimes}(L_2)$ then $\alpha_1 = \beta_1 \otimes \gamma_1$ and $\alpha_2 = \beta_2 \otimes \gamma_2$ for $\beta_1 \in L_1, \beta_2 \in L_2, \gamma_1, \gamma_2 \in \Sigma^{\omega}$.

Then $y = (\beta_1 \otimes \gamma_1) \otimes (\beta_2 \otimes \gamma_2) = (\beta_1 \otimes \beta_2) \otimes (\gamma_1 \otimes \gamma_2)$ as $\otimes$ is commutative and associative. This implies $UC_{\otimes}(L_1 \otimes L_2) = UC_{\otimes}(L_1) \otimes UC_{\otimes}(L_2)$. Hence the result.

Corollary

For any family of languages F, $y \in UC_{\otimes}(\bigcup_{L \in F} L) = \bigcup_{L \in F} UC_{\otimes}(L) = \bigcup_{L \in F} L$.

3. SUPERIMPOSITION UNDER CONTROL ON INFINITE WORDS

Let *V* be the alphabet *V = {f, s}*. Let $c \in V^{\omega}$ be the ω - control word and $C \subseteq V^{\omega}$ be the set of ω - controls.

Definition 3.1

Let $x = x_1 x_2 \ldots \in \Sigma^{\omega}$ and $y = y_1 y_2 \ldots \in \Sigma^{\omega}$ for $x_i, y_i \in \Sigma, i = 1,2,\ldots, j = 1,2,\ldots$.

Let $c = z_1^{n_1} z_2^{n_2} \ldots \in V^{\omega}, z_i \in V, n_i \in \mathrm{N}$.

The superimposition of *x* and *y* controlled by $c \in V^{\omega}$ is $x \otimes_c y = \begin{cases} (x_1 x_2 .. x_{n_1}) \otimes (y_1 y_2 .. y_{n_2}) \ldots .., z_1 = f \\ (y_1 y_2 .. y_{n_1}) \otimes (x_1 x_2 .. x_{n2_1}) \ldots .., z_1 = s \end{cases}$

Example 3.1

Let *α = a₁a₂a₃..., β = b₁b₂b₃..., c = f²s³f⁵sfs...* Then $\alpha \otimes_c \beta = (a_1 a_2 \otimes b_1 b_2 b_3)(a_3 a_4 a_5 a_6 a_7 \otimes b_4) \ldots$

Example 3.2

Let $x = (abcdefg)^{\omega}$ and $y = (gfedcba)^{\omega}$ be ω - words over the finite alphabet *Σ = {a, b, c, d, e, f, g}*. Let $c = sf^2(s^2f^2)^{\omega}$. Then $x \otimes_c y = gba^{\omega}$.

Definition 3.2

If C is a control language over V^{ω} then $x \otimes_C y = \bigcup_{c \in C} x \otimes_c y$.

If *C* is the empty set then $x \otimes_c y = \phi$.

Also if $L_1, L_2 \subseteq \Sigma^{\omega}$ then $L_1 \otimes_c L_2 = \{x \otimes_c y / x \in L_1, y \in L_2\}$.

Definition 3.3

Let *C* be a control set. The set *C* is commutative if and only if the operation $\otimes_c$ is commutative, that is, $x \otimes_c y = y \otimes_c x$ for all $x, y \in \Sigma^{\omega}$.

Let $\aleph^{\omega}$ be the family of all commutative sets of control words.

Proposition 3.1

If $(C_i)_{i \in I}$ is a family of control languages such that (C_i) is a commutative control language for all $i \in I$ then $C' = \bigcap_{i \in I} C_i$ is also a commutative control language.

Proof: Let $u, v \in \Sigma^*$ and let $w \in u \otimes_{C'} v$. Then it follows that $w \in u \otimes_{C_i} v$ for all $i \in I$. But, each (C_i) is commutative and hence $w \in v \otimes_{C_i} u$ for all $i \in I$. Therefore, $w \in v \otimes_{C'} u$. Thus, we have $u \otimes_{C'} v = v \otimes_{C'} u$ which implies that $C' = \bigcap_{i \in I} C_i$ is also a commutative control language.

Definition 3.4

International Journal of Advances in Computer and Electronics Engineering
Volume: 01 Issue: 03, July 2016, pp. 42 – 46

Let C be a control language. The commutative closure of C denoted by $\overline{C}$ is given by $\overline{C} = \bigcup C'$.

Corollary: For all $C \subseteq \{f, s\}^{\omega}$, $\overline{C}$ is a commutative control language.

Remark: $\overline{C}$ is the smallest control language that contains C.

Definition 3.5

A control language C is associative if and only if $\otimes_c$ is associative. We have $x \otimes_c (y \otimes_c z) = (x \otimes_c y) \otimes_c z$ for all $x, y, z \in \Sigma^{\omega}$.

Proposition 3.2

If $(C_i)_{i \in I}$ is a family of control languages such that (C_i) is an associative control language for all $i \in I$ then $C'' = \bigcup_{i \in I} C_i$ is also an associative control language.

Proof: Let $x, y, z \in \Sigma^{\omega}$ and let $w \in x \otimes_{C''} (y \otimes_{C''} z)$. Then it follows that $w \in x \otimes_{C_i} (y \otimes_{C_i} z)$ for all $i \in I$. But, each (C_i) is associative and hence $w \in (x \otimes_{C_i} y) \otimes_{C_i} z$ for all $i \in I$. Therefore, $w \in (x \otimes_{C''} y) \otimes_{C''} z$. Thus, we have $x \otimes_{C''} (y \otimes_{C''} z) = (x \otimes_{C''} y) \otimes_{C''} z$ which implies that $C'' = \bigcup_{i \in I} C_i$ is also an associative control language.

Definition 3.6

Let C be a control language. The associative closure of C denoted by $\overline{\overline{C}}$ is $\overline{\overline{C}} = \bigcup_{c \subseteq C'', C' \in A} C''$ where A is the family of all associative control languages.

Proposition 3.3

The associative closure $\overline{\overline{C}}$ is also an associative control language.

4. CONCLUSION

The present study is new and not found in the literature. Hence it is a significant work and a step forward in the research on combinatorics on words. Since the study involves Picture composition in Astronomy,the theoretical studies can also be applied to the field of Astronomy. This study can be a foundation for connecting diverse fields – Formal language theory, Combinatorics on words and Astronomy.

The superimposition operation on infinite words is defined and the properties of the operation are characterised. Future developments on this topic are many as it is basically a binary operation.This study can be extended to two dimensions by defining the superimposition on arrays and array languages. Visualisation of data in Astronomy is done by the superimposition of arrays. This study has applications in such studies.

REFERENCES

[1] Ahmad Kadrie, V.R.Dare, D.G.Thomas, K.G.Subramanian(2001), Algebraic properties of the shuffle over trajectories,Information Processing LettersVol. 80, pp. 139-144

[2] C. Annal Deva Priya Darshini, V. Rajkumar Dare,2015,Superimposition Operations on Words, Proceedings ofNational Conference on Mathematics and Computer Applications,pp. 48 – 52.

[3] J.E. Hopcroft, J.D. Ullman, Introduction to Automata Theory, Languages and Computation,1979,Narosa Publishing House.

[4] J.P. Tremblay and R. Manohar, (1997)Discrete Mathematical Structures with Applications to Computer Science, Tata McGraw Hill Edition.

[5] Choffrut C. and Karhumaki J.,(1997),Combinatorics of words in Handbook of Formal Languages, Vol. 1, Springer Verlag, pp.329 – 438.

[6] Christian Choffrut, Berke Durak,(2005) ,Collage of two – dimensional words,Theoretical Computer Science 340 pp. 364 – 380.

[7] Jean Berstel, Dominique Perrin, (2007) The origins of combinatorics of words,European Journal of Combinatorics, 28,pp. 996 – 1022.

[8] Lila Santean, Six Arithmetic – like operations on Languages, R.R.L – C.A.T.A, XXV, 1, pp.65 – 73, Bucarest, 1988.

[9] M. Lothaire, Combinatorics on Words,(1983) Encyclopedia of Mathematics and its Applications vol. 17, Addison-Wesley, Reading, Mass.

[10] A. Mateescu, G. Rozenberg and A. Salomaa, (1998), Shuffle on Trajectories: Syntactic Constraints,Theoretical Computer ScienceVol. 197, pp.1 – 56.

Authors Biography

Ms. C. Annal Deva Priya Darshini is an Assistant Professor teaching in the Department of Mathematics, Madras Christian College, Tambaram since 2006. She completed her PG in Mathematics in 1997 from the University of Madras. Her research interests are Formal languages, Combinatorics on Words and DNA Computing.

Dr. V. Rajkumar Dare taught in the Department of Mathematics, Madras Christian College, Tambaram, Chennai for over 30 years. He headed the Department during the years 2008 – 2012. His research interests are Formal Languages and Automata Theory, Infinite Words, Topological studies on Words and Arrays.