A STUDY ON INSERTION AND DELETION OF WORDS

Sindhu J Kumaar[1,] P. J. Abisha[2]

[1]*Department of Mathematics, B. S. A. University, Chennai – 600 048, Tamil Nadu, India.*
[2]*Department of Mathematics, Madras Christian College, Chennai – 600 059, Tamil Nadu, India.*

Abstract: In addition to being fundamental in formal language theory, the operations of insertion and deletion have recently become of interest in connection with the topic of molecular computing. In this paper we have reviewed four papers of Lila Kari

I. INTRODUCTION

The area of molecular computing was born in 1994 when Adleman succeeded in solving an instance of the directed Hamiltonian Path Problem solely by manipulating DNA strands. This marked the first instance where a mathematical problem could be solved by biological means. To answer this question, various models of molecular computation have been proposed, and for some of these models it has been shown that the bio operation involved can simulate the actions of a Turing machine.

II. REVIEW – PAPER 1

Contextual Insertions / Deletions and Computability

- *"Lila Kari and Gabriel Thierrin"*

In this paper the authors focus on the formal language operations of contextual insertion and deletion of words that have recently become of interest in the context of molecular computing. Given a pair of words (x, y), called a context, the (x, y) – contextual insertion of a word v into u is performed as follows:

$$u\ \overleftarrow{(x,y)}\,v = \{u_1\,xvy\,u_2 \,/\, u_1,u_2 \in X^*, u = u_1 x y u_2\}$$

If the word u does not contain xy as a sub word, the result of the (x, y) – contextual insertion is empty set. If $C \subseteq X^*xX^*$ is a set of contexts, the C – contextual insertion of u into v is defined as

$$u\ \overleftarrow{C}\,v = \{u_1\,xvy\,u_2 \,/\, (x,y) \in C, u = u_1 x y u_2, u_1,u_2 \in X^*\}$$

In a manner similar to contextual insertion, authors define contextual deletion: deletion of a word takes place only if certain contexts are present. More precisely, let (x, y) € X^* x X^* be a context. The (x, y) – contextual deletion of v € X^* from u € X^* is defined as

$$u\,\overrightarrow{(x,y)}\,v = \{u_1 xy u_2 \,/\, u_1,u_2 \in X^*, u = u_1 xvy u_2\}$$

If $C \subseteq X^*xX^*$ is a set of contexts, the C – contextual deletion of v from u is defined as

$$u\ \overrightarrow{C}\,v = \{u_1 xy u_2 \,/\, (x,y) \in C, u = u_1 xvy u_2, u_1,u_2 \in X^*\}$$

Also the authors have defined dipolar deletion which is defined as follows

Let u, v be words in X^* and (x, y) € X^* x X^* be a context. The (x, y) – contextual dipolar deletion of v from u is defined as

$$u\,\overleftrightarrow{(x,y)}\,v = \{w \in X^* \,/\, u = u_1 x w y \text{ and } v = u_1 x y u_2\}$$

The findings of this paper

The following propositions and corollaries were discussed in this paper

1. If the set of contexts C is finite, REG, CF, CS are closed under C – contextual insertion
2. If L_1, L_2 are languages over X^+, L_2 a regular language, and if (x, y) € X^* x X^* is a context, then there exists a gsm g (with erasing),

depending on (x, y) and L_2, such that $L_1 \overrightarrow{(x,y)} L_2 = g(L_1)$

3. If the set of contexts C is finite, then REG and CF are closed under C – contextual deletion with regular languages
4. There exists a finite set of contexts C such that the families of context free and context sensitive languages are not closed under C – contextual deletion.
5. If $C \subseteq X^*xX^*$ is a set of contexts, the C – ins – closure of a language If $L \subseteq X^*$ is $I_c(L) \subseteq L \overset{*}{\overleftarrow{C}} L$
6. If $C \subseteq X^*xX^*$ is a set of contexts, the C – del – closure of L is $D_c(L) = \bigcup_{n \geq 0} D_n(L)$
7. If C is a finite set of contexts, the family of regular languages is closed under C – contextual dipolar deletion.
8. There exists a finite set of contexts such that CF and CS are not closed under C – contextual dipolar deletion.
9. If L is a language over X and $C \subseteq X^*xX^*$ is a set of contexts, then ins_c (L) = $(L^c \underset{c}{\Leftrightarrow} L)^c$
10. If $C \subseteq X^*xX^*$ is a finite set of contexts and L is a regular language, th erothen ins_c (L) is regular.
11. Given a language $L \subseteq X^*$ and a set of contexts $C \subseteq X^*xX^*$, del_c (L) = $(L \underset{c}{\Leftrightarrow} L^c)^c \cap$ Sub_c (L)
12. Let L, R be languages over an alphabet X and let ◊, ∇ be two binary operations right inverses (left inverses) to each. If the equation L ◊ X = R has a solution, then also the language R' = $(L \nabla R^c)^c$ is a solution. Moreover, R' includes all the other solutions of the equation.
13. The left inverse of the C – contextual insertion is the C – contextual deletion, while its right inverse is the reversed C – contextual dipolar deletion
14. If a language is accepted bya Turing machine TM, then there exists an insdel system ID accepting the same language

III. REVIEW – PAPER 2

Word insertion and Primitivity

- *"Lila Kari and Gabriel Thierrin"*

This paper deals with relation between the operation of word insertion and Primitivity. A necessary and sufficient condition under which the insertion u ← u of a word u into itself has maximum cardinality is obtained. The notion of insertion sequence is introduced and sufficient condition under which an insertion sequence is a special type of language (regular, context free, biprefix code) is obtained. Based on the operations of insertion, shuffle and commutative shuffle (which generalize catenation), the notion of ins – primitive words, shuffle primitive words, and com – shuffle – primitive words defined and investigated.

The result of insertion u ← v is a set with cardinality greater than one, and less than or equal to $|u| + 1$, where $|u|$ denotes the length of u. The cardinality of u ← v is in direct connection with the structure of u and v, if they contain repetitive patterns, then chances are that there are two different places of insertion of u and v that yield the same word. To formalize this idea, authors introduce the notion of insertion of rank i, $u \leftarrow_i v$ means that v is inserted exactly after the i^{th} letter of u.

Then the authors talk about Primitivity. A necessary and sufficient condition for u ← u to have maximum cardinality is that the word is primitive. Primitivity is defined as a word u € X^+ is said to be primitive if $u = v^n$, $n \geq 1$, implies n = 1.

The findings of this paper

The following propositions and corollaries and lemmas were discussed in this paper

1. Let u, v € X^+,
   (i) If $u \leftarrow_i v = v \leftarrow_j u$ and o ≤ I, j ≤ min $\{|u|, |v|\}$, then either u is an infix of v or vy = xu for some x, y € X^*
   (ii) If $u \leftarrow_i v = v \leftarrow_j u$ and o ≤ I, j ≤ min $\{|u|, |v|\}$, then either u is a prefix of v or v is a prefix of u

(iii) If u $\leftarrow_i$ u = u $\leftarrow_j$ u and o < I < j < │u│, then either u = gk for some g € X^+ and k ≥ 2, the word u is not primitive.

2. Let u € X^+ with │u│= k. Then card (u ← u) = k iff u is a primitive word.
3. Let w € X^+. If there exists an integer N such that for n ≥ N the pilot sequence is a context free language
4. Let w € X^+ is a primitive word the pilot sequence is constant, not equal to 0 or │w│, then the corresponding inserting sequence is a non regular context free language.
5. If u is a word in X^* then $(u \leftarrow^n u) \leftarrow^m (u \leftarrow^p u) \subseteq (u \leftarrow^* u)$, for all m, n, p ≥ 0
6. For every word u € X^+ there exists an ins – primitive word v and a positive integer n such that u € $v \leftarrow^n v$
7. Let u € X^+. The word u is a dipolar word ⇔ for every interval pilot sequence σ, the corresponding inserting sequence σ(u) = {u_0, u_1, u_2, …, u_k, …} is a language that is biprefix code.
8. If w € $u \leftarrow^n u$ then $N_a(w) = (n + 1) N_a(u)$ for all a € X
9. If u € X^+, a € X, u ≠ a^n then either u is ins – primitive or all words w € (u ← a) are ins – primitive
10. If L ⊆ X^+ be an ins – closed language such that L^c is ins – closed. Let IP(L) be set of ins – primitive words and let IF(L) be set of minimal words of L relatively to the infix order $\leq_I$. Then
    (i) If u € L and if v is ins root of u, then v € L
    (ii) If L' is an ins – closed language containing IP(L) then L ⊆ L'
    (iii) Every word u € IF(L) is ins – primitive
11. For every word u € X^+ there exists a shf – primitive word v and a positive integer n such that u Π^n v.
12. If L ⊆ X^+ be a shuffle – closed language such that L^c is also shuffle closed . Let SH(L) be shf – primitive words in L and let EB(L) be the set of minimal words of L relatively to the embedding order $\leq_a$. Then
    (i) If u € L and if v is a shf – root of u then v € L
    (ii) If L' is a shuffle closed language containing SH(L), then L ⊆ L'
    (iii) Every word u € EB(L) is shf – primitive
13. For every word u € X^+ there exists a com – shf – primitive word v and a positive integer n such that u € v $©^n$ v.
14. Let u € X^* be a word and let aalph(u) = {a_1, …, a_n}, n ≥ 2. Then u is com – shf – primitive iff the numbers $N_{a_1}(u), N_{a_2}(u), \ldots, N_{a_n}(u)$ are relatively prime.

IV. REVIEW – PAPER 3

Power of controlled insertion and deletion

- *"Lila Kari"*

This paper investigates classes of languages obtained as the closure of certain atomic languages under some insertion and deletion operations. Each of the classes studied is closed under an insertion operation, a deletion operation and an iterative insertion one. The operations are controlled and have been chosen as stated in order to allow an increase as well as decrease of the length of the words in the operands.

The simplest and most natural generalization of catenation is the sequential insertion. Given two words u and v, instead of catenating v at the right extremity of u, the new operation inserts v in an arbitrary place in u. But in this paper the author defines a control mechanism for insertion of v into u (i.e) each letter determines what can be inserted after it. The catenation operation will then be obtained by using a particular case of controlled insertion. The Δ - controlled insertion is defined as:

$$L \leftarrow \Delta = \bigcup_{u \in L} (u \leftarrow \Delta), \text{where}$$

$$u \leftarrow \Delta = \{u_1 a v_a u_2 / u = u_1 a u_2 , v_a \in \Delta(a)\}$$

Note that the empty word does not occur in the result of insertion. Instead, the notion of control implies the presence of at least one letter in the word in which the insertion is performed.

Therefore we have

$$\left(L_1 \leftarrow \Delta\right) = \left(L_1 - \{\lambda\}\right) \leftarrow \Delta$$

The findings of this pap:

The following propositions and corollaries and lemmas were discussed in this paper

1. The family of regular languages is not closed under iterated insertion
2. The family of context – free and context sensitive languages are closed under iterated insertion
3. The family of regular and the family of context free languages are not closed under parallel insertion
4. The family of context – sensitive languages is closed under iterated parallel insertion
5. S is contained in the family of context free languages and properly contains the family of regular languages (Where S is the smallest class of languages which contains Φ)
6. For each language L in S there exists a natural number n_0 such that every word w € L with I_g (w) > n_0 possesses a decomposition w = $w_1w_2w_3$ satisfying
   (i). $w_2 = w_2' w_2''$
   (ii). There exists a nonempty word u such that $w_1w_2'''w_3$ € L for all k ≥ 0
7. The family S is strictly included in the family of context free languages
8. The family S is closed under insertion
9. P is contained in the family of context sensitive languages and properly contains the family of regular languages (where P is the smallest class of languages which contains the empty set)
10. P′ is a Boolean algebra properly containing the family of regular languages.

V. REVIEW – PAPER 4

Insertion and Deletion for Involution Codes

- *"Lila Kari and Kalpana Mahalingam"*

This Paper introduces a generalization of the operation of catenation: $u[k]_l v$, the left k – insertion, is the set of all words obtained by inserting v into u in positions that are at k letters away from the left extremity of the word u. The authors define k – suffix codes using the left – k insertion operation and extend the concept of k – prefix and k – suffix codes to involution k – prefix and involution k – suffix codes. An involution code refers to any of the generalizations of the classical notion of codes in which the identity function is replaced by an involution function. (An involution function θ is such that θ^2 equals identity). They also extend the notion of k – insertion closure and k – deletion closure of a language to incorporate the notion of an involution function. Thus to an involution map θ and a language L, we associate a set k – o – ins(L) with the property that their k – insertion into any word L yields words which belong to θ(L). The properties of these languages are also studied.

The findings of this paper:

The following propositions and corollaries and lemmas were discussed in this paper

1. If L is a nonempty language, $\delta^*_k(L)$ is the minimal left k – ideal containing L.
2. A nonempty language L $\subseteq \sum^+$ is a k – suffix code if and only if L is a δ_k antichain
3. Let S $\subseteq \sum^+$
   (i) If S is a k = suffix code, then $\delta_k(S)$ is a left k – ideal and $Suf_k(S) = S$
   (ii) If L is a left k – ideal, L ≠ $\sum^*$, then there exists a unique k – suffix code S namely S = $Suf_k(L)$, such that L = $\delta_k(S)$
4. The catenation of k – suffix codes is k – suffix codes
5. When θ is antimorphic involution, if L is a k – θ – bifix code, then L^n is a k – θ – bifix code for all n ≥ 1.
6. When θ is morphic involution the class of k – θ – prefix (suffix) codes is closed under concatenation.
7. Let L $\subseteq \sum^+$ be such that L ∩ θ(L) ≠ Φ

   (i) If L^m is a k – θ – prefix for m ≥ 1, then

   L is a k – θ – prefix.

(ii) If L^m is a k – θ – suffix for m ≥ 1, then L is a k – θ – suffix.

(iii) If L^m is a k – θ – bifix for m ≥ 1, then L is a k – θ – bifix.

8. If L is a commutative language, then * - k – θ- ins(L) is also a commutative language
9. * - k – θ- ins(L) = $((\theta(L)) \Leftrightarrow^{k}_{*} L)^c$
10. L is * - k – θ- ins – closed iff L $\leftarrow^{k}_{*}$ L ⊆ θ(L)
11. If L is a commutative language, then * - k – θ- del(L) is also commutative.
12. * - k – θ- del(L) = $(\theta(L) \Leftrightarrow^{k}_{*} L^c)^c \cap$ * - k – Sub (θ(L)).
13. Let L be such that L is * - k – θ- ins – closed. Then L is * - k – θ- ins – closed iff L = $(\theta(L) \Leftrightarrow^{k}_{*} L)$.

VI. GENERALIZED SHUFFLE ON TRAJECTORY

The binary word operations, whose simplest examples are catenation and left/right quotient, have been extensively studied in the formal language theory. They are important for composition / decomposition of languages and their descriptions (grammars automata). They are of key importance for forming algebraic structures of formal languages, as the abstract families of languages.

Here we give a new definition for catenation, left/right quotient etc…

Definition:

Generalized shuffle of α by β on trajectory t denoted by $\alpha \blacktriangle_t \beta$ is defined as follows.

$$\alpha \blacktriangle_t \beta = \left\{ \begin{array}{l} \alpha_1 \alpha_2 \alpha_3 \ldots \alpha_i \beta_1 \beta_2 \ldots \beta_j \alpha_{i+1} \ldots \alpha_k \;/\; \beta = \beta_1 \beta_2 \beta_3 \ldots \beta_n, \\ t = 0_i 1_1 1_2 \ldots 1_j 0_{i+1}, 1 \le i \le k, 1 \le j \le n \end{array} \right\}$$

Where in $\alpha \blacktriangle_t \beta$, the 0's in t corresponds to letters in α and the 1's in t corresponds to letters in β. The subscripts represent the position of these letters in α and β.

In the above definition if trajectory t = $0_1 0_2 \ldots 0_k 1_1 1_2 \ldots 1_n$ then it represents catenation. Similarly depending on the position of i and j in the trajectory the operation becomes contextual insertions / deletions, word insertion, controlled insertion and deletion etc…

VII. CONCLUSION

The authors team is working on Generalized Shuffle on Trajectory which is expected to give more results as shuffle on trajectory.

REFERENCES

[1]. Lila Kari and Gabriel Thierrin, "Contextual Insertions / Deletions and Computability", National Sciences and Engineering Research Council of Canada, Academic Press, 1996, 47–60.

[2]. Lila Kari and Gabriel Thierrin, "Word Insertions and Primitivity", Utilitas Mathematica, 53(1998), pp 49 – 61.

[3]. Lila Kari, "Power Controlled Insertion and Deletion", Part of the Project 11281 of the academy of Finland, 1998, pp 197 – 212.

[4]. Lila Kari and Kalpana Mahalingam, "Insertion and Deletion for Involution Codes" National Science and Engineering Research Council of Canada, Academic Press, 1996, 47 – 60.