Parikh Representation for Splicing System

S. JeyaBharathi and S. Sinthanai Selvi

Abstract--- *Here the graph splicing scheme of Rudolf Freund is correlated to semigraph which is introduced by E. Sampath Kumar[10]. It had been recognized that at the time of splicing, the graph takes the bipartite structure and it generates the strings which leads to the Parikh concepts. Graphical representation of generalized Parikh vector of splicing system also represented by Codifiable Language.*

Keywords--- *Splicing System, Semigraph, Codifiable Languages, Generalized Parikh Vector*

I. INTRODUCTION

TOM Head introduced the new concept of splicing system for describing the recombinant behaviors of double stranded DNA molecules. Since then the theory of splicing systems has become a new interacting area of formal Languages Theory. DNA exist not only as Linear Molecules but also as Circular molecules. Investigating the Complex structures of genes, Rudolf Freund [5] suggested graphs as more suitable objects for modelling such graphs, and exhibit the relation between regular graph splicing systems on graphs and splicing systems on strings. DNA sequences are three-dimensional objects in a three dimensional space. Hence graphs seems to be more suitable objects for describing complex three-dimensional objects independently from their actual position in the three dimensional space. The advantages of arranging the vertices of an edge in a semigraph given in the following order,

1. Each edge can be written as a line with its vertices arranged in order, and so semigraphs look like graphs when drawn in a plane.
2. ii) The concept of planarity in semi graphs is a straight forward generalization of the concept of planarity in graphs, which otherwise is not true in hyper graph.

This semigraph is more powerful than graph model. This presents communication volume accurately.

Semi graph model will eventually replace graph partitioning in scientific computing. A DNA single-strand consists of four different types of units called nucleotides or bases strung together by an oriented backbone like beads on a wire. The bases are Adenine (A), Guanine (G), Cytosine (C), Thymine (T) and A can chemically bind to an opposing T on another single-strand while C can similarly bind to G. Bases that can thus bind are called Watson-Crick (WK) complementary and two DNA single strands with opposite orientation and with WK complementary bases at each position can bind to each other to form a DNA double strand in an process called base-pairing. The other biochemical properties of DNA are all harnessed in bio-computing [1]. To encode information using DNA, one can choose an encoding scheme mapping the original alphabet onto strings over {A, C, G, T} and proceed to synthesize the obtained information-encoding strings as DNA single strands. The DNA strands representing the output of the computation can then be read out and decoded.The Parikh Vector of a word introduced by Siromoney. R, Dare, V.R [16] has been a significant contribution in the theory of formal languages. The concept of Generalised Parikh Vector was introduced by Huldah Sathyaseelan, V. Rajkumar Dare [17]. The Parikh Vector of a word counts the number of occurrences of each letter of the alphabet where as the Generalised Parikh Vector of a word indicates the positions of each letter of the alphabet in the word.

II. PRELIMINARIES

2.1 Splicing System

Splicing is a model of the recombinant behaviour of double stranded molecules of DNA under the action of restriction enzymes and ligases. A single stranded of DNA is an oriented sequence of nucleotides A, C, G & T but since A can bind to T & G to C, two strands of DNA bind together to form a double stranded DNA molecule, if they have matching pairs of nucleotides when reading the second one along the reverse orientation.

2.2 SemiGraph

A semigraph G is a pair (V, X) where V is a non - empty set whose elements are called vertices of G and X is a set of n - tuples called edges of G of distinct vertices for various $n \geq 2$, satisfying the following conditions:

S.G-1 Any two edges have at most one vertex in common.
S.G-2 Two edges $(u_1,u_2,\ldots,u_n)$ and $(v_1,v_2,\ldots,v_m)$ are considered to be equal if and only if, (i) m = n and (ii) either $u_i = v_i$ or $u_i = v_{n-i+1}$ for $1 \leq i \leq n$. Thus the edges $(u_1, u_2, u_3 \ldots u_n)$ is the same as the edge $(u_n, u_{n-1} \ldots u_1)$.

2.3 Semi Vertices

Let G be a graph, when splicing G, we obtain new vertices which are called as semi vertices denoted by V', where

$$|V'| = p'.$$

Dr.S. JeyaBharathi, Associate Professor, Department of Mathematics, Thiagarajar College of Engineering, Madurai, India.E-mail:sjbmat@tce.edu
S. Sinthanai Selvi, Assistant Professor, Department of Mathematics, Mannar Thirumalai Naicker College, Madurai, India.E-mail:sinthanai.vel@gmail.com

ISBN 978-93-82338-35-2 | © 2013 Bonfring

2.4 Semi Edges

Let G be a graph when splicing G, we obtain new edges by decomposition of edges which are called as semi edges denoted by E', where |E' | = q'.

2.5 n-cut Graph Splicing

u_{C_n} Represents a n-cut spliced semi graph where u is the string, c- the number of cut and n- the number of splicing.

2.6 2 - Cut Splicing

Consider L = 4, the number of set of combinations required for 2– Cut splicing. Here p = 16, q = 6,

q' = 8, r = 1, k = 2.1, k = 2. Then p-(q + q') + r = 16 - (6 + 8) + 1 = 2 + 1 = 3 = k + 1. Euler's polyhedral formula p - (q + q') + r = k + 1 is satisfied in 2-Cut splicing. [from the following semi graphs H].

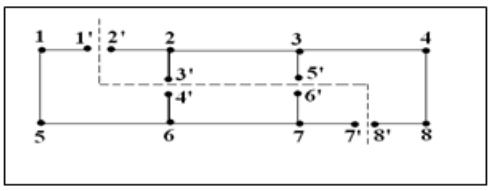


Figure 2.1

The graph H at 2-Cut splicing we get

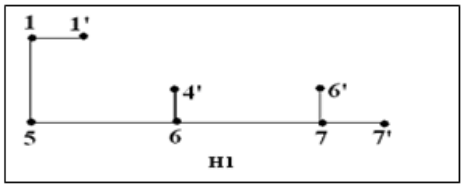


Figure 2.2

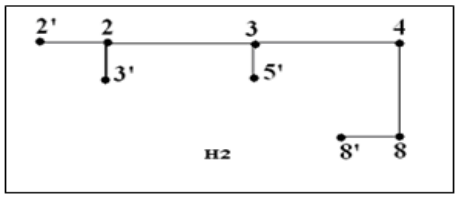


Figure 2.3

III. Language on Bipartite Spliced Semi Graph Strings (BSSG)

Let the weight of each edge of a bipartite spliced semi graph as 'a' then the weight of each semi edge becomes 'a/2' After 1–Cut splicing on graph G, we get spliced semi graphs which form two bipartite semi graphs. Each one has three semi edges and two edges. Therefore let the language be $(a/2)^3 a^2$. After 2–Cut splicing on graph G, each one has four semi edges and three edges. Therefore let the language be $(a/2)^4 a^3$.

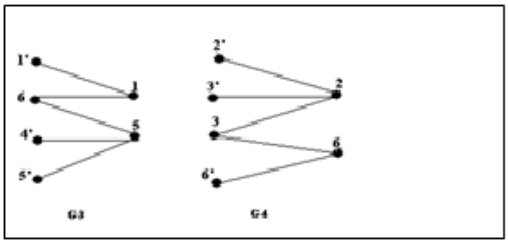


Figure 3.1

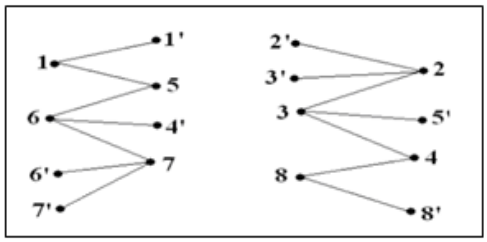


Figure 3.2

In general the language defined for all n cut SSG is $\{b^{n+2} a^{n+1}$ / for every n≥1} where $b = \frac{a}{2}$

IV. Codifiable And Partially Codifiable Languages

Definition [14]

Let ∑ be an alphabet with card (∑) =k.

a language $L \subseteq \sum^*$ is,

Codifiable if for each order on ∑ the Parikh matrix mapping $\Psi_m: L \rightarrow M_{k+1}$ is injective.

Partially codifiable if there is atleast one order on ∑ such that the corresponding Parikh mapping is injective in L.

Theorem 4.1:

If ∑ = {a,b} then a language L is contained ∑* is codifiable if and only if L is partially codifiable.

Proof: For a binary alphabet, only two orders are possible. Assume that ∑={a < b} and $\Psi_m : L \rightarrow M_3$ is not injective. $|\alpha|_a = |\beta|_a$ is not injective. Let α,β ЄL be two words such that $\Psi_m(\alpha) = \Psi_m(\beta)$. Note that $|\alpha|_a = |\beta|_a$,
$|\alpha|_b = |\beta|_b$ and $|\alpha|_{\text{scatt-ab}} = |\beta|_{\text{scatt-ab}}$.

Since for each binary word x,

$|x|_{\text{scatt-ba}} = |x|_a |x|_b - |x|_{\text{scatt-ab}}$.

Hence it follows that

$|\alpha|_{\text{scatt-ba}} = |\beta|_{\text{scatt-ba}}$

Hence $\Psi_{m\ o}(\alpha) = \Psi_{m\ o}(\beta)$ and therefore L is not codifiable on the ordered alphabet

$$\sum = \{a < b\}.$$

Hence proved

ISBN 978-93-82338-35-2 | © 2013 Bonfring

V. The Context Free Codifiable Language on N -Cut BSSG Strings

The Language on n cut BSSG is

$L=\{b^{n+2}a^{n+1}=b^{m+1}a^{m}$,where m=n+1, n≥1}

The language is generated by the following context free grammar in the Greiback normal form,

$$S' \to S, S \to aSB \mid abB \mid, B \to b \mid$$

$$S' \to S \;\; n_a := 0, n_b := 0, n_{ab} := 0$$

$$S \to aSB, n_a := n_a + 1$$

$$S \to aabBBn_b := n_a + 1, n_{ab} := n_{ab} + n_a$$

$$B \to b, n_b := n_b + 1$$

For example, let the word = aabbb.

$$S' \Rightarrow S_{(0,0,0)} \Rightarrow aSB_{(1,0,0)} \Rightarrow aabBB_{(2,1,2)} \Rightarrow aabbB_{(2,2,4)} \Rightarrow aabbb_{(2,3,6)}$$

Hence $\Psi_m(aaabb) = \begin{pmatrix} 1 & n_a & n_{ab} \\ 0 & 1 & n_b \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 6 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{pmatrix}$

Number of Strings on the language generated from n cut splicing.

Table 5.1

EVERY n-CUT	LANGUAGE	NUMBER of STRINGS
1 CUT	b^3a^2= bbbaa	$\frac{5!}{3!2!}=10$
2 CUT	b^4a^3 =bbbbaaa	$\frac{7!}{4!3!}=35$
n CUT	$b^{n+2}a^{n+1}$	$\frac{2n+3\ !}{n+2\ !(n+1)!}$

VI. The Generalised Parikh Vector

For each u$\in \sum^{\alpha}$, the generalized parikh vector denoted by P (u) = (p_1,p_2,…p_n)

Where $p_i = \sum_{j \in A_i} \frac{1}{2^j}$ where $A_i \subset N$ contains all the positions where a_i occurs in u. For 1 cut BSSG,

Let w_1 = bbbaa

P(w_1)=($1/2^4$+$1/2^5$,$1/2^1$+$1/2^2$+$1/2^3$)=(0.094,0.875)

w_2=babab, p(w_2) = $1/2^2$+$1/2^4$,$1/2^1$+$1/2^3$+$1/2^5$)

=(0.313,0.656)

w_3 = aabbb, p(w_3) = (0.75,0.219)

w_4 = bbaab, p(w_4) = (0.188,0.781)

w_5 = baabb, p(w_5) = (0.315,0.594)

w_6 = abbba, p(w_6) = (0.281,0.438)

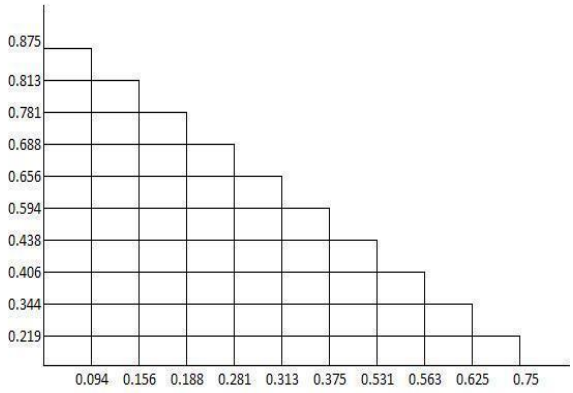
w_7=ababb, p(w_7) = (0.375,0.344)

w_8 = abbab, p(w_8) = (0.563,0.406)

w_9 = bbaba, p(w_9) = (0.156,0.813)

w_{10} = babba, p(w_{10}) = (0.281,0.688)

6.1 Graphical Representation of Generalized Parikh Vector on Bipartite Spliced Semi Graph



Note:

Similarly we can have the Generalized Parikh vector for every n cut BSSG strings and also we can have the Geometrical representation of each.

Theorem 6.2:

Every n cut BSSG strings of length (2n+3) lying on the corresponding language line is

$$x+y = (2^{(2n+3)} -1) / 2^{(2n+3)}$$

where x and y are the number of positions of a and b.

Proof:

Let the 1 cut BSSG, the length of each string is 5. The GPV of each string is of the form of 31/32. (ie) (2^5 -1)/2^5. Similarly for 2 cut BSSG strings of length 7. The GPV of each string is of the form 127/128. (ie) (2^7-1)/2^7. In general the GPV of every n cut BSSG strings lies on the language line of

$$x+y=(2^{(2n+3)}-1)/\ 2^{(2n+3)}.$$

VII. Conclusion

In this paper, we discussed the Graphical representation of generalized parikh vector of splicing system represented by codifiable language. In future, the techniques of generalized Parikh vector for n-cut splicing can be extended to the concept of Parikh matrix for splicing system.

Acknowledgement

The authors gratefully acknowledge the support from theGrant by UGC coming under Major Research Project- [41-800/2012].

References

[1] L. Adleman, Molecular computation of solutions to combinatorial problems, Science, 266, Pp.1021-24, 1994.

[2] J. Dassow, V. Mitrana, Self Cross-over Systems, Discrete Mathematics and Theoretical Computer Science', Springer Series,G.Paun (Ed.), ,Pp. 283-294, 1998.

[3] R. Freund, Splicing Systems on Graphs, Proc. Intelligence in Neural and Biological Systems, IEEE Press, Pp.189-194, 1995.
[4] K. Gnanamalar David , K.G. Subramanian and D. Gnanaraj Thomas, A Note on Graph Splicing Languages, Lecture Notes in Computer Science, Vol.2340, Springer-Verlag, Pp.381-390, 2001.
[5] F. Harary, Graph Theory, Addision-Wesley, Reading, Mass, 1969.
[6] T. Head, Formal Language Theory and DNA An analysis of the generative capacity of recombinant behaviors. Bulletin of Mathematical Biology, 49, Pp.735-759, 1987.
[7] N. Jonoska, 3D DNA Patterns and Computation, Proc. Molecular Computing, India, Pp.20-32, 1998.
[8] J. Padmashree, K. Thiagarajan, M. Kameshwari, S. Jeya Bharathi, DNA Splicing System through semigraph, Proc. International Conference on Emerging Trends in Mathematics and Computer Applications, India, , Pp.75 – 78, 2010.
[9] A. Salomaa, Formal Languages, Academic Press, New York, 1973.
[10] E. Sampathkumar, Semigraphs and their Applications, Report on the DST project, 2000.
[11] R. Siromoney, K.G. Subramanian and V.R. Dare, Circular DNA and Splicing systems, Lecture Notes in Computer Science, Vol.654, Springer-Verlag, Berlin, Pp. 260-273, 1992.
[12] K.G. Subramanian and Ang Miin Huey on Parikh Matrices of Words Int. J.Found 2010.
[13] A. Atanasiu, Binary amiable words, Intern. J.Found Comput.Sci.18 Pp.387-400, 2007.
[14] Adrian Atanasiu, Carlos Martin-Vide and Alexandru Mateescu on codifiable and the Parikh Matrix Mapping J. of Uni. Comp. Sci. Vol.7 No.9, 783-793, 2001.
[15] A. Mateescu, A. Salomaa, K.Salomaa, S. Yu, A sharpening of the Parikh mapping, Thoret Informatics Appl., 35, Pp.551-564, 2001.
[16] R. Siromoney, V.R. Dare , A Generalization of Parikh Vectors for finite and infinite words, Lecturer notes in Computer Science, 206, Spinge Verlag, 1985.
[17] Huldah Sathyaseelan, V. Rajkumar Dare on Generalized Parikh Vectors proc. National Conference on Discrete Mathematics and its Applications, NCDMA 2007.

ISBN 978-93-82338-35-2 | © 2013 Bonfring