Characteristics of Self-Assembled Semigraphs in DNA Splicing System

Dr. S. JeyaBharathi\textsuperscript{a}, P. R. Indhu\textsuperscript{b}
Department of Mathematics,
Thiagarajar College of Engineering,
Madurai, India
\textsuperscript{a}sjbmat@tce.edu, \textsuperscript{b}indhupr92@gmail.com

S. Sinthanaiselvi\textsuperscript{c}
Department of Mathematics,
SRM University,
Chennai, India
\textsuperscript{c}sinthanai.vel@gmail.com

Abstract

In this paper, the notion of self-assembly to overlap two semigraphs in DNA splicing system is proposed. Besides studying the characteristics of these self assembled semigraphs, it had been identified that \( n+1 \) number of graphs are produced in self-assembling of any two \( n \)-cut spliced semigraph. To assemble more than two graphs at the same time, sequential self-assembling is introduced. The necessary and sufficient condition for a sequence of semigraphs not to self-assemble into a single semigraph in DNA splicing system is answered. The question of the existence of the self-assembled graph that connects all graphs in a given set of semigraphs is also discussed.

1. Introduction

Tom Head proposed [4] an operation called splicing for describing the recombinant behaviours of double stranded DNA sequences under restriction enzymes and ligases. Investigating the Complex structures of genes, Rudolf Freund [2] suggested graphs as more suitable objects for modelling such processes and exhibit the relation between regular graph splicing systems on graphs and splicing systems on strings. In [5], the graph splicing scheme of Rudolf Freund and the semigraphs introduced by E. Sampathkumar [8] is correlated. As a result, the notion of \( n \)-cut Splicing Semigraph was introduced. The semigraphs that is gotten spliced is known as spliced semigraphs. This paper illustrates all type of semigraphs in DNA splicing system and its characteristics in the third section. Self-assembly is a process in which smaller objects selectively aggregate with each other into complex structures, which in turn self-assemble into larger aggregates. Under self-assembly, objects spontaneously arrange themselves into patterns. DNA recombination is one such DNA self-assembly by which Adleman [1] solved an instance of Hamiltonian path problem. DNA self-assembly is the core of the most of DNA computing. In [9], R. Rama propose a graph-based self-assembly model. Two simple graphs \( G_1 \) and \( G_2 \) with a vertex of common degree overlap and a new self-assembled graph is formed and this definition is applicable for simple graphs that is, the graph without multiple edges or loops. In this paper, we define the self-assembly on semigraphs in DNA splicing system. In particular, the self-assembly of \( n \)-cut spliced semigraph [5] is investigated. The sequential self-assembly approach is also proposed in the fourth section.

2. Preliminaries

We begin with basic definitions. Much of this section appeared first in [5], we include it for completeness. Basic definitions of splicing system and graph-theory are not numbered, but only recalled. Splicing System 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 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.

A graph \( G \) over \( V \) is a triple (\( N, E, L \)) where \( N \) is the set of nodes, \( E \) is the set of edges of the form (\( n, m \)) with \( n, m \in N \), \( n \neq m \) and \( L \) is the function from \( N \) to \( V \) assigning a label from \( V \) to each node of \( N \). The degree of a vertex \( v \) in \( G \) (say \( \text{deg} (v) \)) is the number of edges incident on \( v \). By taking \( V \) as the set of nucleotides we can draw a graph for DNA molecule. The splicing operation on graphs was described by
Freund in [2] and the graph of DNA over the set of nucleotides holds the characterization of semigraph during splicing. 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 ≥ 2, satisfying the following conditions:

S.G-1 Any two edges have at most one vertex in common. S.G-2 Two edges (u₁, u₂, u₃, ..., uₙ) and (v₁, v₂, v₃, ..., vₙ) are considered to be equal if and only if (i) n = m and (ii) either uᵢ = vᵢ or uᵢ = vₙ₋₁ᵢ for 1 ≤ i ≤ n. Thus the edge (u₁, u₂, u₃, ..., uₙ) are the same as the edge (uₙ, uₙ₋₁, ..., u₁).

During the splicing of a semigraph of DNA molecule, we will get two isomorphic spliced semigraphs which is generalized as n-cut Spliced Semigraph (n-cut SSG) and the formal definitions are as follows.

**Definition 2.1:** \(u_{c,p,c} \) - Represents an n-cut splicing semigraph where u is the string, c- the number of cut and n- the number of splicing. [5]

**Definition 2.2:** Let G be a graph, when splicing G, we obtain new vertices which are called as semi vertices denoted by \(V'\), where |\(V'\)| = p'.

**Definition 2.3:** Let G be a graph when splicing G, we obtain new edges by decomposition of edges which are called as semiedges denoted by \(E'\), where |\(E'\)| = q'.

Let us call the vertices other than semivertices in the n-cut SSG as non-semivertices and the non-semi vertex set is denoted by V. Similarly the edges other than semiedges are known by non-semiedges and the set is denoted by E.

**Example 2.1:**

![Fig 2.1. 1-cut splicing](image1)

![Fig 2.2. 1-cut SSG](image2)

The graph in fig 2.1 represents 1-Cut splicing \((u_{c,1})\).

During splicing G is spliced into two isomorphic spliced semigraphs (fig 2.2 and 2.3). First graph in fig 2.2 is a 1-cut spliced semigraph with semivertices \{1',4',5'\}, non-semivertices \{1,6,5\} and the semiedges \{(1',1'), (6,4'), (5,5')\}, non-semiedges \{(1,6), (6,5)\}. Before going into the next section, we must understand that every spliced semigraph has semivertices under any kind of splicing (not only n-cut splicing)

### 3. Characteristics of Semigraphs in DNA Splicing System.

In this section, we depict the semigraph in DNA Splicing System and investigate its characteristics. Let us consider D as the set of all semigraphs in DNA splicing system. As the semigraphs in DNA splicing system is a graph representation of double-stranded DNA molecules structure, each semigraph in D should preserve the linearity of dsDNA structure. In general, a semigraph in D can be defined as follows.

**Definition 3.1:** A semigraph in DNA splicing system is a pair \(<V', V''\), \(<E', E''\)> in which the vertex set is partitioned into semivertices and non-semivertices. Similarly the edge set is partitioned into semiedges and non-semiedges. Each graph in D will be a spliced semigraph where splicing can be of any type.

**Example 3.1:** Any n-cut SSG is in D.

![Fig 3.1. 2-cut SSG and 4-cut SSG](image3)

**Example 3.2:** The following example shows various type of splicing and spliced semigraphs that are in D.

![Fig 3.2. Spliced in semigraphs in D](image4)

In (a), a complete rDNA semigraph is undergone three kinds of splicing. (a-1-2-3-4) are the spliced semigraphs. Note that any spliced semigraphs will have at least two semivertices and semiedges. It is the necessary condition only but not a sufficient condition. It can be proved from the following counter example.
Example 3.3:

![Semigraph that is not in D](image)

Fig 3.3. Semigraph that is not in D

The above semigraph has two semivertices and semiedges but not in D, since this kind of semigraph cannot be made by any kind of splicing.

Example 3.4:

![Rotation of semigraph in D](image)

Fig 3.4. Rotation of semigraph in D

We may wonder the graph (d) cannot be in D. But the rotation of upper portion about 90 degree downwards makes the graph (d) into (e) which is of course belongs to D. So we can conclude that a semigraph which can be made into a semigraph that satisfies the conditions of semigraphs in DNA splicing system by doing some desired and possible rotations is also in the set D. This undefined operation on graphs is known as maintaining/preserving the linearity of DNA structure.

**Definition 3.2:** Let v be any vertex in a semigraph from D.

- If deg(v) = 0, then v is denoted by Sdeg(v).
- If deg(v) = 1, then v is denoted by NSdeg(v).

**Result 3.1:** Any SG ∈ D has the following properties:

1. Always degree of a semivertex is one.
2. deg(v) = Sdeg(v) + NSdeg(v), if v ∈ <V, V'>
3. Sdeg(v) = 0 and NSdeg(v) = 1, if v ∈ V'
4. 0 ≤ Sdeg(v) ≤ 3 and 0 ≤ NSdeg(v) ≤ 3, if v ∈ V
5. \[ \sum_{v \in <V, V'>} Sdeg(v) = \sum_{v \in V'} Sdeg(v) = \sum_{v \in V} NSdeg(v) = |V'|
6. \[ \sum_{v \in <V, V'>} NSdeg(v) = \sum_{v \in V'} NSdeg(v) + \sum_{v \in V} NSdeg(v) \]

**Definition 3.3:** A pair of semivertices u', v' in a semigraph are said to be fused if the two vertices are replaced by a single new vertex such that every edge (both semi and non-semiedges), that was incident on either u' or v' or on both, is incident on the new vertex.

The new vertex is known as fusing vertex and it is neither semivertex nor non-semivertex. It pretends as an imaginary vertex.

**Result 3.2:** By fusing operation, the number of non-semivertices will not be changed and semi-vertices are decreased by two. The number of non-semiedges is increased by one and semiedges are decreased by two.

**Definition 3.4:** Complete rDNA semigraph in D is a semigraph without any semivertices.

Example 3.5:

![Complete rDNA semigraph](image)

Fig 3.5. Complete rDNA semigraph

**Definition 3.5:** A star semigraph (SG_{1,n}) is a semigraph with one non-semivertex v as internal node and all other vertices which are evidently semivertices, are only adjacent to v. For example SG_{2} in Fig 4.1 is SG_{1,3}.

4. Self-Assembly of Semigraph in DNA Splicing System

In this section, we try to relay the self-assembly with the semigraph on DNA splicing system. At this time we are taking into consideration only the assembling of the semivertices of a graph instead of taking the end vertices with same degree as in [9].

**Definition 4.1:** Let SG_{1}, SG_{2} be two semigraphs in D. Then the self-assembly of semigraphs SG_{1}, SG_{2} involves two steps,

(i). SG_{1} > v < SG_{2}, v_{1}' = v = v_{2}', v_{1}' \in V'(SG_{1}), v_{2}' \in V'(SG_{2}), v is the assembling vertex.

(ii). Fusing the pair of semivertices in SG that are at the attractable position. Two semivertices are said to be at the attractable position, if the fusion of these vertices maintains the linearity of DNA semigraph structure.

By SG = SG_{1} > v < SG_{2}, we mean that the graph SG is obtained by drawing SG_{1} and SG_{2} such that the vertex v_{1}' \in V'(SG_{1}) and v_{2}' \in V'(SG_{2}) are superimposed. The vertex v (= v_{1}' = v_{2}') is called the assembling vertex and it is just an imaginary vertex like fusing vertex. Hence the new semigraph SG' is known as self-assembled semigraph. That is SG' is a semigraph got by superimposing a semivertex v' of SG_{1} with a semivertex u' of SG_{2} and fusing the semivertices of SG_{1} and SG_{2} that are at the attractable position.
Example 4.1: Let SG₁ be the 1-cut SSG (Fig. 2.2) and the star semigraph SG₁₃ as SG₂. We only take the semivertices. \{u₁, u₂, u₃\} and \{v₁, v₂, v₃\} are such semivertices in SG₁ and SG₂ respectively.

\[ S₁(SG₁, SG₂): \]

\[ (I) \quad v(= u₁'= v₁') \]
\[ (II) \quad v(= u₁'= v₁') \]
\[ (III) \quad v(= u₂'= v₂') \]
\[ (IV) \quad v(= u₂'= v₂') \]
\[ (V) \quad v(= u₁'= v₁') \]
\[ (VI) \quad v(= u₁'= v₁') \]
\[ (VII) \quad v(= u₁'= v₁') \]
\[ (VIII) \quad v(= u₁'= v₁') \]

**Fig 4.1. Self-assembly of SG₁ and SG₂**

From the above nine graphs, v is the assembling vertex (enclosed by a circle) which may be taken as a non-semivertex and the fusion of semivertices at the attractive position is marked by the rectangles. Note that the semigraphs (I) \approx (II) \approx (III) \approx (IV) \approx (V) \approx (VI) and (VII) \approx (VIII) \approx (IX). As an outcome, \( S₁(SG₁, SG₂) = \{ (I), (VII) \} \). Notice that we maintain the linearity of self-assembled graphs. Hence we can consider semigraphs (I) and (II) are same semigraph.

The set of all self-assembled semigraph got by self-assembling SG₁ and SG₂ is denoted by \( S(SG₁, SG₂) \). If \( SG \in S(SG₁, SG₂) \), then \( V(SG) = V(SG₁) \cup V(SG₂) \). \( E(SG) = E(SG₁) \cup E(SG₂) \cup A \), where

\[ A = \{(x,y,z)/ \text{either y is assembly vertex or fusing vertex and x, z } \in \text{E(SG₁)} \cup \text{E(SG₂)}\}. \] Similarly, \( V'(SG) = (V'(SG₁) - \{v₁\} - \{f₁, f₂, ...\}) \cup (V'(SG₂) - \{u₁\} - \{g₁, g₂, ...\}) \) where \( f₁, f₂, ... \) and \( g₁, g₂, ... \) are the semivertices under fusion. \( E'(SG) = (E'(SG₁) - C) \cup (E'(SG₂) - D) \), where \( C = \{(x,y)/ y' \text{ is under either assembling or fusion in } SG₁\} \) and \( D = \{(x,y)/ y' \text{ is under either assembling or fusion in } SG₁\} \). The order of \( SG = \text{order of } SG₁ + \text{order of } SG₂ - 2 - 2x \) (number of fused vertices).

**Result 4.1:** If the number of semivertices in SG₁ and SG₂ is p and q respectively, then \( |S(SG₁, SG₂)| \) is at most \( pxq \).

**Definition 4.2:** For the sets of semigraphs \( L₁, L₂ \subset D \),

\[ S(L₁, L₂) = \bigcup_{SG₁ \in L₁, SG₂ \in L₂} S(SG₁, SG₂) \]

We write \( S(L) \) instead of \( S(L, L) \). For example, Let \( L₁ \) is the finite set of n-cut SSGs for different value of n. Then \( S₁(L₁) \) is the set of all semigraphs formed by self-assembling each pair of n-cut SSG from \( L₁ \).

**Result 4.2:** All the semigraphs in \( S(v', SG₁,n') \) are isomorphic. That is \( |S(v', SG₁,n')| = 1 \).

**Result 4.3:** Consider the semigraph SG with p semivertices. Then \( |S(SG, SG₁, v)| = p \).

**Theorem 4.1:** For any semigraphs \( SG₁, SG₂ \) from D,

1. \( S(SG₁, SG₂) = S(SG₂, SG₁) \).
2. \( S(SG₁, SG₂) \neq \phi \) iff \( SG₁ \) and \( SG₂ \) has at least one semivertex.
3. Always self-assembly is not possible with a complete rDNA semigraph.

**Proof:** The proof is immediate.

**Proposition 4.1:** Let \( L₁ \) be a set of n-cut SSG such that \( |L₁| = m \) - any positive integer, then \( S(L₁) \) contains ‘m’ number of complete rDNA Semigraph.

**Proof:** From [7], every n-cut Splicing forms two isomorphic semigraphs (n-cut SSG) and when \( i \neq j \), i-cut SSG is not isomorphic to j-cut SSG. Hence in \( S(L₁) \), complete rDNA is built during the self-assembly \( S(i \text{-cut SSG}, i \text{-cut SSG}) \), for \( i = 1, 2, ..., m \).

**Theorem 4.2:** The necessary and sufficient condition for \( S(i \text{-cut SSG}, j \text{-cut SSG}) \) to contain a complete rDNA semigraph is \( i = j \).

**Proof:** The proof is obvious from proposition 3.1.
Theorem 4.3: Consider any n-cut SSG and m-cut SSG and \( n \leq m \). Then \(|S(n\text{-cut SSG, m-cut SSG})| = n + 2m\).

Proof: We know that every n-cut SSG structure ends with a portion like a non-semivertex with two semivertices. During assembly if this structure within each SSG tends to impose obviously we will get two different structures. By superimposing the end portion of m-cut SSG with the semivertices of n-cut SSG, we will get \( n-1 \) number of graphs. Finally, by assembling the other semivertices of m-cut SSG with n-cut SSG, we will get \( 2(m-1) + 1 \) graphs and no other assembling is possible. So we get \( 2 + (n-1) + 2(m-1) + 1 = n + 2m \).

Corollary 4.3.1: \(|S(n\text{-cut SSG})| = 3n\).

Result 4.4: For any n-cut SSG, \( \sum_{v \in V \times \Delta \deg(v)=0} 1 = n+1 \)

The following theorem is an alternate to theorem 4.5.

Theorem 4.4: Let \( n \leq m \), then \(|S(n\text{-cut SSG, m-cut SSG})| = c_1 + 2c_2 -3 \) where \( c_1 = \sum_{v \in V_1 \times \Delta \deg(v)=0} 1 \) and \( c_2 = \sum_{v \in V_2 \times \Delta \deg(v)=0} 1 \).

Proof: From the result 4.3, \( c_1 = (n+1) \) and \( c_2 = (m+1) \). Hence \( c_1 + 2c_2 -3 = (n+1) + 2(m+1) - 3 = n + 2m \).

Corollary 4.4.1: \(|S(n\text{-cut SSG})| = 3(c_1 - 1)\).

Example 4.2: Let \( SG_1 \) be 1-cut SSG and \( SG_2 \) be 2-cut SSG. Then \( c_1 = 2 \) and \( c_2 = 3 \), \(|S(SG_1, SG_2)| = 2 + (2*3) - 3 = 5\).

And, \(|S(SG_1, SG_2)| = 1 + (2*2) = 5\).

Example 4.3: \(|S(5\text{-cut SSG})| = 3(6-1) = 15\).

By theorem 4.3, \(|S(5\text{-cut SSG})| = 3*5 = 15\).

Although we get lot of results on the self-assembly of two semigraphs in \( D \), the following example shows that all such self-assembled graphs need not to be in \( D \).

Example 4.4: The self-assembled graph brought by assembling the following graphs does not belong to \( D \).

Example 4.5: Let \( SG \) be the following graph.

**Fig 4.3. Iterative self-assembly of SG**

but, \( S\_D^2(SG) \) is empty. Hence the process is stop at second iteration.

Similarly, we write for the set of graphs \( L \) as

\[ S\_D^0(L) = S(L, S\_D^{n-1}(L)) \ n \geq 1, \] where \( S\_D^0(L) = L \) and \( S\_D^2(L) = \bigcup_{n \geq 1} S\_D^n(L) \).

Result 4.6: \( S\_D^+(S_{1,3}) \) is infinite.

Proposition 4.2: \(|S\_D(n\text{-cut SSG, m-cut-SSG})| = n+1\) where \( n \leq m \).

Proof: From the proof of theorem 4.3, every n-cut SSG structure end with a claw and during assembling of this claw from each graph makes 2 graphs which will be in \( D \). The successive assembly of the claw in m-cut SSG with other vertices of n-cut SSG produces n-1 number of graphs in \( D \). After, the end claw from m-cut SSG meet with the other end of n-cut SSG, by assembling the other semivertices of m-cut SSG with n-cut SSG, we will get \( 2(m-1) + 1 \) graphs. Obviously these graphs cannot be semigraphs in DNA splicing system. As a total, we will get \( 2 + (n-1) = n+1 \) number of graphs that is in \( D \).
Theorem 4.5: \( S'(m\text{-cut SSG}) \) is empty for \( n>2 \).

**Proof:** From proposition 4.2, \( |S_0(m\text{-cut SSG})| = m+1 \) when \( n=1 \). By theorem 4.2, whenever two \( m\text{-cut SSG} \) self-assembled, one complete \( rDNA \) is produced and the two ends of other self-assembled graphs will have a non-semivertex that adjacent to a semivertex and a non-semivertex. Although the self-assembly of \( m\text{-cut SSG} \) with these graphs is possible, these assembled graphs cannot be in \( D \). Hence, when \( n=2 \), \( S'_D(m\text{-cut SSG}) = S_D(m\text{-cut SSG}) \) and so further iteration is not possible. Hence the theorem is proved.

**Corollary 4.5.1:** \( S'_D(m\text{-cut SSG}) = S_D(m\text{-cut SSG}) \).

**Corollary 4.5.2:** \( S'_D(m\text{-cut SSG}) \) is finite and contains \( m+1 \) self-assembled graphs.

**Result 4.7:** In a SG, if there exists at least one non-semivertex \( v \) such that \( S\deg(v) = NS\deg(v) = 1 \) and the NSdegree of the adjacent non-semivertex is 2, then \( S'_D(SG) = S_D(SG) \).

**Definition 4.4:** Self-assembly sequence is a sequence of semigraphs from \( D \) and the self-assembly should be taken in the given sequential order. This is written as \( (SG_1, SG_2,..., SG_k) \) which is equivalent to the expression: \( S_D(...S_D(S_D(SG_1, SG_2), SG_3), SG_4)..., SG_k) \).

Although we define a particular self-assembly sequence of graphs instead of taking self-assembly of set graphs, we will not end with a unique graph structure. Because the self-assembly of any two semigraph in \( D \), may end with more number of self-assembled graphs. Fortunately, this sequential self-assembly approach will help us to construct the desired semigraph structure or to find which graph structures can be formed with the particular sequence of graphs. So this comes close to the idea of Adelman to form a DNA computer using self-assembly of DNA molecules. To reduce the complexity and to get a desired graph structure from the general self-assembly sequence, we may restrict the assembling process by mentioning the vertices from each graph that is supposed to undergo the self-assembly operation, like \( (SG_1 > [u_1, v_1], SG_2 > [u_2, v_2], SG_3 > [u_3, v_3],...[u_k, v_k]) \) where each \( v \) and \( u \) are the semivertices from \( SG_1, SG_2, SG_3 \) respectively.

**Example 4.6:** Let us refer 1-cut SSG and \( SG_{1,3} \) from fig 4.1 and take SG:

\[
\begin{align*}
&\text{w'}_1 \\
&\text{w'}_2
\end{align*}
\]

Then ( 1-cut SSG > \([u'_1, v'_1], SG_{1,3} > [v'_3, w'_1] < SG\) where this assembly sequence will make a complete \( rDNA \) semigraph structure.

**Result 4.8:** Always \((< SG_{1,3} > < SG_{1,3} >...< SG_{1,3} > < SG > < SG)\) will form a complete \( rDNA \) semigraph. Here we refer SG from example 4.5.

**Result 4.9:** Always ( n-cut SSG > < SG\( ^{(1)} \) > < SG\( ^{(2)} \) >... < SG\( ^{(n)} \) > < SG ) will form a complete \( rDNA \) semigraph, we refer SG from example 4.5.

**Result 4.10:** \((SG_1 > < SG_2 > < SG_3 >... < SG_k)\) need not be equal to \((SG_k > < SG_{k-1} > < SG_{k-2} >... < SG_i)\). But they are equal in case of restricted self-assembly.

**Result 4.11:** Consider any subsequence \( X \) from the sequence \{(n-cut SSG / n= 1, 2,...)\} where the first two terms are n-cut SSG and m-cut SSG \((n \leq m)\), then the number of sequential self-assembly is at most \( n+1 \).

**Theorem 4.6:** A sequential self-assembly \((SG_1 > < SG_2 > < SG_3 >... < SG_k)\) is empty iff there exists a subsequential self-assembly \( M = (SG_1 > < SG_2 > < SG_3 >... < SG_{i-1}) \) such that \( S_D(M, SG_i) \) is empty. Then by definition 4.4,

\[
S_D (S_D(M, SG_i), SG_{i-1}) = \phi
\]

\[
S_D (S_D(S_D(M, SG_i), SG_{i-1}), SG_{i-2}) = \phi
\]

proceeding like this, we get \( S_D(...(S_D(S_D(M, SG_i), SG_{i-1}), SG_{i-2}),..., SG_k) = \phi \). Hence \( (SG_1 > < SG_2 > < SG_3 >... < SG_k) \) is empty.

Conversely, Let \( N = (SG_1 > < SG_2 > < SG_3 >... < SG_k) \) be empty. Then \( N \) itself can be taken as the subsequential self-assembly such that \( S_D(N, SG_i) \) is empty.

**Definition 4.5:** Let \( L \) be a subset of \( D \) with \( k \) number of graphs and \( \{L_n / 1 \leq n \leq k!\} \) be the sequence of graph sequences of \( L \). Then \( S(L_0) \) is the set of self-
assembled semigraphs that connect all graphs in L and \( \tilde{S}_D(L_n) \subseteq \tilde{S}(L_n) \) and contains the graphs that only belong to D.

**Result 4.12:** Let L be a subset of D and contains k number of semigraphs then \( |\tilde{S}(L_n)| \leq k! \).

**Lemma 4.1:** \( \tilde{S}_D(L_n) \) is empty if there exist two semigraphs in L such that each graph has exactly two non-semivertices whose NSdegree and Sdegree is one.

**Proof:** Suppose SG\(_1\) and SG\(_2\) are the semigraphs in L such that each graph has exactly two non-semivertices whose NSdegree and Sdegree is one. For every sequence of L\(_n\), there exists a situation to overlap the semivertices that are adjacent to these four non-semivertices and which is impossible.

**Theorem 4.7:** If L contains more than two n-cut SSG structure, then \( \tilde{S}_D(L_n) \) is empty.

**Proof:** By theorem 4.2, whenever two m-cut SSG is self-assembled, one complete rDNA is produced and the other self-assembled graphs will have exactly two non-semivertices whose NSdegree and Sdegree is one. In case of complete rDNA semigraph by theorem 4.1 we know, self-assembly is not possible with a complete rDNA semigraph. The second case is proved by the lemma 4.1.

**5. Conclusion**

The semigraph in DNA splicing system is studied in this paper. A self-assembly model is defined on these semigraphs with which we can plan the recombination of dsDNA molecule in splicing system. In particular, our main concern is on self-assembly of n-cut Spliced Semigraph, as this allows more generality than other semigraphs. A sequential self-assembling approach on semigraphs is proposed to support, to construct the desired graph structure from a given set. The possibilities of combining a set of graphs into a single assembled graph are analyzed.

**6. Nomenclature**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SG</td>
<td>Semigraph</td>
</tr>
<tr>
<td>SSG</td>
<td>Spliced Semigraph</td>
</tr>
<tr>
<td>rDNA</td>
<td>Recombinant DNA</td>
</tr>
<tr>
<td>dsDNA</td>
<td>Double Strand DNA</td>
</tr>
</tbody>
</table>

**7. Acknowledgement**

This work was supported by the Grant-in-Aid major research project coming under University Grants Commission- [41-800/2012].

**8. References**