ON SOME OPERATIONS SUGGESTED BY GENOME EVOLUTION

J. DASSOW
Faculty of Computer Science, University of Magdeburg
P.O.Box 4120, D-39016, Magdeburg, Germany

V. MITRANA[a]
Faculty of Mathematics, University of Bucharest
Str. Academiei 14, 70109 Bucharest, Romania

Three operations involved in the genome evolution namely, inversion, transposition and duplication, are considered as operations on strings and languages. We show that, for any pair of these operations, there is a language family which is closed under one of the operations and not closed under the second one, however, under some mild conditions the closure of a language family under one of the operations implies that it also closed with respect to another one.

1 Introduction and definitions

The genomes of complex organisms are organized into chromosomes which contain genes arranged in linear order. Much of the current data for genomes is in the form of maps which are now becoming available and permits the study of the evolution of such organisms at the scale of genome for the first time [3].

It is rather commonly asserted that DNA is a language for specifying the structures and processes of life. Despite this opinion biological sequences has not been investigated very vivid so far by methodes developed in the field of formal language theory. A pioneer's work has been reported in [1] where very simple genes were described by means of regular grammars. Since then most investigations on the topic have used grammar formalism [4, 2, 15].

In the course of its evolution, the genome of an organism mutates by different processes. At the level of individual genes the evolution proceeds by local operations (point mutations) which substitute, insert and delete nucleotides of the DNA sequence. Evolutionary and functional relationships between genes can be captured by taking into considerations only local mutations [14]. These operations viewed as operations on strings and languages have been considered from different points of view [15] and the references thereof.

However, the analysis of the genomes of some viruses (Epstein-Barr and Herpes simplex viruses, see for instance [6, 9]) have revealed that the evolution

[a] Research supported by the Alexander von Humboldt Foundation

of these viruses involved a number of large-scale rearrangements in one evolutionary event. This non-local rearrangements include: *inversion, transposition, duplication* and *splicing.*

- Inversion replaces a segment of a chromosome with its reverse DNA sequence.
- Transposition moves a segment to a new location in the genome.
- Duplication copies a segment to a new location.
- Splicing results in recombination of genes in a pair of homologous chromosomes by exchanging segments between parental chromatides. Splicing can be modelled as a process that exchanges segments at the end of two chromosomes.

The splicing operation was introduced as a language theoretic operation in [7] and then vividly investigated in a series of papers e.g. [11, 12, 5, 10]. A survey can be found in [8].

The other aforementioned operations appear very attractive to us, too. Consequently, in the present paper we shall investigate the other operations from the formal language theory point of view.

It is worth mentioning here that these operations on languages have been considerated in [15] as well. The operations investigated in the present paper are generalizations of the operations studied in [15]. Furthermore, the iterated versions of operations in debate are also considered.

We now recall some notation from formal language theory and formalize the operations mentioned above.

We denote by V^* the free monoid generated by V under catenation, by λ the empty string and by V^+ the free semigroup generated by V, i.e. $V^+ = V^* \setminus \{\lambda\}$. The length of the string x is denoted by $|x|$ whilst the number of occurrences of the letter $a \in V$ in x is denoted by $|x|_a$.

Further we define the *mirror image* $mi(x)$ of a word $x = a_1a_2\ldots a_n$, $a_i \in V$ for $1 \leq i \leq n$, by

$$mi(a_1a_2\ldots a_n) = a_na_{n-1}\ldots a_1$$

and, for a word $x \in V^*$, let

$$Perm(x) = \{y \mid |y|_a = |x|_a \text{ for } a \in V\},$$

be the set of all words over V which are permutations of x.

Moreover, we recall that a family $\mathcal{F}$ of languages is called a *trio*, if $\mathcal{F}$ is closed under (λ-free) homomorphisms, inverse homomorphisms and intersections with regular sets. It is well-known that any trio is closed under restricted homomorphisms, too[13].

For further details in formal language theory we refer to[13].

Definition 1 *An <u>inversion scheme</u> is a pair $I = (V, I')$, where V is an alphabet and I' is a finite subset of V^*.*

For a given inversion scheme $I = (V, I')$ and a string $x \in V^*$, we define

$$Inv_I(x) = \{x_1 mi(x_2) x_3 \mid x = x_1 x_2 x_3,\ x_2 \in I'\}$$

Definition 2 *A <u>transposition scheme</u> is a pair $T = (V, T')$, where V is an alphabet and T' is a finite subset of $(V^*)^3$.*

For a given transposition scheme $T = (V, T')$ and a string $x \in V^*$ we define

$$Tr_T(x) = \begin{cases} x_1x_3x_4x_2x_5x_6 & \textit{for } x = x_1x_2x_3x_4x_5x_6,\ (x_2, x_4, x_5) \in T' \\ x_1x_2x_5x_3x_4x_6 & \textit{for } x = x_1x_2x_3x_4x_5x_6,\ (x_5, x_2, x_3) \in T' \end{cases} .$$

Obviously, if $(u, \lambda, \lambda) \in T'$, then u can be shifted to every place in the given word x.

Definition 3 *A <u>duplication scheme</u> is a pair $D = (V, D')$, where V is an alphabet and D' is a finite subset of $(V^*)^3$.*

For a given duplication scheme $D = (V, D')$ and a string $x \in V^*$ we define

$$Dupl_D(x) = \begin{cases} x_1x_2x_3x_4x_2x_5x_6 & \textit{for } x = x_1x_2x_3x_4x_5x_6,\ (x_2, x_4, x_5) \in D' \\ x_1x_2x_5x_3x_4x_5x_6 & \textit{for } x = x_1x_2x_3x_4x_5x_6,\ (x_5, x_2, x_3) \in D' \end{cases} .$$

If the applied scheme is obvious from the context, we write Inv, Tr and $Dupl$ instead of Inv_I, Tr_T and $Dupl_D$, respectively.

For all $S \in \{mi, Perm, Inv, Transp, Dupl\}$, the operation can naturally be extended to languages by

$$S(L) = \bigcup_{x \in L} S(x).$$

The iterated versions of the above operations are naturally defined as follows. For $S \in \{Inv, Transp, Dupl\}$ we set

$$\begin{aligned} S^0(L) &= L, \\ S^{i+1}(L) &= S(S^i(L)), \\ S^*(L) &= \bigcup_{i \geq 0} S^i(L). \end{aligned}$$

2 Relations between the operations

The inversion operation looks similar to the mirror image operation mi. It consists in the application of mi to a subword. However, the two operations are quite different as shown in the following proposition.

Proposition 1 *There are families of languages closed under mi but not closed under inversions and vice versa.*

Proof. It is known that the family of D0L languages is closed under mi. Consider the D0L language $L = \{a^{2^n} b^{2^n} \mid n \geq 0\}$ and the inversion scheme $I = (\{a, b\}, \{ab\})$. The language

$$Inv_I(L) = \{a^{2^n - 1} bab^{2^n - 1} \mid n \geq 0\}$$

cannot be generated by a D0L system. Indeed, let us suppose that there exists a D0L system $G = (\{a, b\}, w, h)$ such that $L(G) = Inv_I(L)$. Since $h(a^{2^n - 1} bab^{2^n - 1}) \in Inv_I(L)$, for some $n \geq 2$, it follows that $|h(a)|_b = |h(b)|_a = 0$. Therefore, $h(a) = a^k$ and $h(b) = b^p$ for some $k, p \geq 1$.

If $k = p = 1$, then $L(G)$ is finite, which contradicts the infinity of $Inv_I(L) = L(G)$.

If $k > 1$ or $p > 1$, then $h(a^{2^n - 1} bab^{2^n - 1})$ contains a substring of the form $b^p a^k$, which contradicts the form of the words in $Inv_I(L) = L(G)$.

Now, we shall provide a family of languages closed under inversions but not closed under the mirror image. To this end, take the language $L_0 = \{a^n b^n \mid n \geq 1\}$ and construct recursively the following sequence of language classes:

$$\begin{aligned} \mathcal{F}_0 &= \{L_0\}, \\ \mathcal{F}_{k+1} &= \{Inv_I(L) \mid L \in F_k, I \textit{ is an inversion scheme}\}. \end{aligned}$$

The family

$$\mathcal{F} = \bigcup_{k \geq 0} \mathcal{F}_k$$

is obviously closed under inversions.

The following fact is essential in our proof.

Fact. *For every language $L \in \mathcal{F}$ and any $n \geq 1$ there exists a finite set $A(L, n) \subseteq L$ such that every string x in $L \setminus A(L, n)$ can be expressed as $x = a^p y b^q$ with $p, q \geq n$ and $y \in \{a, b\}^*$.*

If $L = L_0 \in F_0$, then the assertion is trivially true.

Assume that the assertion is true for any language $L' \in F_k$ and take $L \in F_{k+1}$. Then there exists an inversion scheme $Inv = (\{a, b\}, I)$ such that $L = Inv_I(L')$. Let $n \geq 1$ be a given integer and $m = max\{|x| \mid x \in I\}$. By the induction hypothesis it follows that $L' = A(L', n+m) \cup \bar{L}$, where $A(L', n+m)$ is a finite set and every string x in $\bar{L}$ can be written as $x = a^p y b^q,\ p, q \geq n+m$. Consequently,

$$L = Inv_I(L') = Inv_I(A(L', n+m)) \cup Inv_I(\bar{L}).$$

Note that $Inv_I(A(L', n+m))$ is a finite set and any string w in $Inv_I(\bar{L})$ can be decomposed as $w = a^r z b^s$ with $r, s \geq n$ and $z \in \{a, b\}^*$, which completes the proof of the fact.

Now it is clear that the mirror image of any language in $\mathcal{F}$ cannot be in $\mathcal{F}$ because it does not satisfy the requirements of the aforementioned fact. □

We now prove that the three operations introduced above also differ in that sense that the closure under one operation do not imply the closure with respect to another one.

Theorem 2 *For any pair (X, Y) with $x, y \in \{Inv, Tr, Dupl\}$, $x \neq Y$, there is a language family $\mathcal{L}$ such that $\mathcal{L}$ is closed under X and is not closed under Y.*

Proof. First we consider the family $\mathcal{F}$ defined in the second part of the proof of Proposition 1. By construction $\mathcal{F}$ is closed under inversion. On the other hand, if we apply the transposition scheme

$$T = (\{a, b\}, \{(aa, b, b)\})$$

to the language $L_0 \in F$ we obtain a language, which contains the set of all words $a^{n-2} b^{n-1} aab$ with $n \leq 2$. This contradicts the fact shown in the proof of Proposition 1. Therefore $\mathcal{F}$ is not closed under transposition.

Moreover, if we consider T as a duplication scheme we can prove by analogous arguments that F is not closed under duplication.

Let V be an alphabet. Then we consider the family $\mathcal{L}$ consisting of all languages L such that there is an integer $n \geq 1$ with $L \subseteq V^n$. Obviously, $\mathcal{L}$ is closed under inversion and transposition since these operations do not change the length of a word.

On the other hand, applying the duplication scheme

$$(V, \{(a, \lambda, \lambda), (aa, \lambda, \lambda)\}),$$

where $a \in V$, to the language $\{a^2\} \in \mathcal{L}$ yields the language $\{a^3, a^4\}$ which is not in $\mathcal{L}$.

Let $V = \{a, b\}$. Then let $\mathcal{L}'$ be the family of all languages L over V such that each word in L can be expressed as $x_1 a x_2 b x_3$, i.e. any word of L contains ab as a scattered subword. Obviously, $\mathcal{L}'$ is closed under duplication, since duplication adds additional subwords and does not destroy scattered subwords.

On the other hand, the application of the inversion scheme (V, ab) and the transposition scheme $(V, \{a, b, \lambda)\})$ to the language $\{ab\} \in \mathcal{L}'$ yields $\{ba\} \notin \mathcal{L}'$, which proves the nonclosure of $\mathcal{L}'$ under inversion and transposition. □

However, the situation changes if we restrict the families of languages under consideration.

Theorem 3 *Let $\mathcal{L}$ be a family of languages which is closed under homomorphisms and inverse homomorphisms. Then the following statements hold.*

i) $\mathcal{L}$ is closed under transpositions iff $\mathcal{L}$ is closed under duplications.

ii) The closure of $\mathcal{L}$ under transpositions (or duplications, respectively) implies the closure of $\mathcal{L}$ under inversions.

iii) If $\mathcal{L}$ is closed under union and inversions, then $\mathcal{L}$ is closed under transpositions and duplications.

Proof. i) First, we shall prove that the closure under transposition implies the closure under duplication. Let $D = (V, \{(x_i, y_i, z_i) \mid 1 \leq i \leq n\})$ be a duplication scheme. We consider the homomorphisms

$$\begin{aligned}
h_1 &: (V \cup \bigcup_{i=1}^{n} \{c_i, d_i\})^* \longrightarrow V^*, \\
&h_1(a) = a \text{ for } a \in V, \quad h_1(c_i) = x_i,\ h_1(d_i) = y_i z_i \text{ for } 1 \leq i \leq n \\
h_2 &: (V \cup \bigcup_{i=1}^{n} \{c_i, d_i\})^* \longrightarrow (V \cup \bigcup_{i=1}^{n} \{c_i, d_i, c_i', d_i'\})^*, \\
&h_2(a) = a \text{ for } a \in V, \quad h_2(c_i) = c_i c_i',\ h_2(d_i) = d_i d_i' \text{ for } 1 \leq i \leq n, \\
h_3 &: (V \cup \bigcup_{i=1}^{n} \{c_i, q_i, q_i', p_i\})^* \longrightarrow (V \cup \bigcup_{i=1}^{n} \{c_i, d_i, c_i', d_i'\})^*, \\
&h_3(a) = a \text{ for } a \in V, \\
&h_3(q_i') = d_i c_i' d_i',\ h_3(q_i) = c_i c_i',\ h_3(c_i) = c_i,\ h_3(p_i) = d_i d_i' \text{ for } 1 \leq i \leq n, \\
g &: (V \cup \bigcup_{i=1}^{n} \{p_i, q_i', q_i, c_i \mid 1 \leq i \leq n\})^* \longrightarrow V^*, \\
&g(a) = a \text{ for } a \in V, \\
&g(q_i') = y_i x_i z_i,\ g(q_i) = g(c_i) = x_i,\ g(p_i) = y_i z_i \text{ for } 1 \leq i \leq n
\end{aligned}$$

and the transposition scheme

$$T = (V \cup \bigcup_{i=1}^{n} \{c_i, c_i', d_i, d_i'\}, \{(c_i', d_i, d_i') \mid 1 \leq i \leq n\}).$$

Every string in the language $Tr_T(h_2(h_1^{-1}(L)))$ is either of the form

$$xc_iyd_ic'_id'_iz \quad \text{or} \quad xd_ic'_id'_iyc_iz$$

with

$$x, y, z \in (V \cup \bigcup_{i=1}^{n} \{c_ic'_i, d_id'_i\})^*.$$

Now, it is easy to see that

$$Dupl_D(L) = g(h_3^{-1}(Tr_T(h_2(h_1^{-1}(L)))))$$

Conversely, for the transposition scheme

$$T = (V, \{(x_i, y_i, z_i) \mid 1 \leq i \leq n\})$$

we construct the homomorphisms h_1 and h_2 as above and consider the homomorphisms

$$\begin{aligned}
&h'_3 \ : \ (V \cup \bigcup_{i=1}^{n} \{p_i, p'_i, q_i, q'_i\})^* \longrightarrow (V \cup \bigcup_{i=1}^{n} \{c_i, d_i, c'_i, d'_i\})^*,\\
&h'_3(a) = a \text{ for } a \in V,\\
&h'_3(q'_i) = d_id'_i,\ h'_3(q_i) = c_ic'_i,\ h'_3(p_i) = c_id_ic'_i,\\
&h'_3(p'_i) = d_id_ic'_id'_i \text{ for } 1 \leq i \leq n,\\
&g' \ : \ (V \cup \bigcup_{i=1}^{n} \{p_i, p'_i, q_i, q'_i\})^* \longrightarrow V^*,\\
&g'(a) = a \text{ for } a \in V,\\
&g'(q'_i) = y_ix_iz_i,\ g'(q_i) = x_i,\ g'(p'_i) = y_iz_i,\ g'(p_i) = \lambda \text{ for } 1 \leq i \leq n
\end{aligned}$$

and the duplication schemes

$$\begin{aligned}
D_1 &= (V'', \{(d_i, c_i, c'_i) \mid 1 \leq i \leq n\}),\\
D_2 &= (V'', \{(d_ic'_i, d_i, d'_i) \mid 1 \leq i \leq n\})
\end{aligned}$$

with

$$V'' = V \cup \bigcup_{i=1}^{n} \{c_i, c'_i, d_i, d'_i\}.$$

Then we obtain

$$Tr_T(L) = g'((h'_3)^{-1}(Dupl_{D_2}(Dupl_{D_1}(h_2(h_1^{-1}(L)))))).$$

ii) By i) it is sufficient to give a proof for transpositions. Let

$$I = (V, \{x_1, x_2, \dots, x_n\})$$

be an inversion scheme. Then we construct the homomorphisms h_2 and h_3 and the transposition scheme T as in the proof of i) and modify h_1 and g to

$$\begin{aligned}
&h_1' \ : \ (V \cup \bigcup_{i=1}^{n} \{c_i, d_i\})^* \longrightarrow V^*,\\
&h_1'(x) = x \text{ for } x \in V \cup \{c_i \mid 1 \leq i \leq n\}, \quad h_1'(d_i) = \lambda \text{ for } 1 \leq i \leq n,\\
&g'' \ : \ (V \cup \bigcup_{i=1}^{n} \{p_i, q_i', q_i, c_i \mid 1 \leq i \leq n\})^* \longrightarrow V^*,\\
&g''(a) = a \text{ for } a \in V,\\
&g''(q_i') = \lambda, \ g''(q_i) = x_i, \ g''(c_i) = mi(x_i), \ g''(p_i) = \lambda \text{ for } 1 \leq i \leq n.
\end{aligned}$$

Then we obtain

$$Inv_I(L) = g''(h_3^{-1}(Tr_T(h_2((h_1')^{-1}(L))))).$$

iii) Again, by i) it is sufficient to give a proof for transpositions. Obvously, if $T = (V, \{t_1, t_2, \dots, t_n\})$ is a transposition scheme and $T_i = (V, \{t_i\})$ for $1 \leq i \leq n$, then

$$Tr_T(L) = Tr_{T_1}(L) \cup Tr_{T_2}(L) \cup \dots \cup Tr_{T_n}(L).$$

By supposition, $\mathcal{L}$ is closed under union, and thus it is sufficient to show that $\mathcal{L}$ is closed under applications of transpositions schemes of the form $\bar{T} = (V, \{(x, y, z)\})$. We consider the homomorphisms

$$\begin{aligned}
&f_1 \ : \ (V \cup \{c, d\})^* \longrightarrow V^*,\\
&f_1(a) = a \text{ for } a \in V, \quad f_1(c) = x, \quad f_1(d) = yz,\\
&f_2 \ : \ (V \cup \{c, d\})^* \longrightarrow (V \cup \{c, d, c', d'\})^*,\\
&f_2(a) = a \text{ for } a \in V, \quad f_2(c) = cc', \quad f_2(d) = dd',\\
&f_3 \ : \ (V \cup \{q, q', p, p'\})^* \longrightarrow (V \cup \{c, d, c', d'\})^*,\\
&f_3(a) = a \text{ for } a \in V, \ f_3(q) = cc', \ f_3(q') = dd', \ f_3(p) = c'c, \ f_3(p') = d'd,\\
&f \ : \ (V \cup \{p, p'q, q'\})^* \longrightarrow V^*,\\
&f(a) = a \text{ for } a \in V, \quad f(q') = yz, \quad f(q) = x, \quad f(p) = \lambda, f(p') = yxz
\end{aligned}$$

and the inversion schemes

$$I_1 = (V \cup \{c, d, c', d'\}, \{cc'\}) \quad \text{and} \quad I_2 = (V \cup \{c, d, c', d'\}, \{dd'\})$$

and obtain

$$Tr_{\bar{T}}(L) = f(f_3^{-1}(Inv_{I_1}(Inv_{I_2}(f_2(f_1^{-1}(L)))))).$$

□

3 Closure properties of some families

We first study the closure under (non-iterated) inversion, duplication, and transposition.

Theorem 4 *Any trio is closed under duplications, transpositions and inversions.*

Proof. Let $\mathcal{F}$ be a trio and $L \subseteq V^*$ be a language in $\mathcal{F}$. Further let $D = (V, \{(x_i, y_i, z_i) \mid 1 \leq i \leq n\}$ be a duplication scheme. We define the homomorphisms

$$\begin{array}{ll}
h_1 : (V \cup \bigcup_{i=1}^{n}\{c_i, d_i\})^* \longrightarrow V^*, & h_1(a) = a \text{ for } a \in V, \\
& h_1(c_i) = x_i \text{ for } 1 \leq i \leq n, \\
& h_1(d_i) = y_i z_i \text{ for } 1 \leq i \leq n, \\
h_2 : (V \cup \bigcup_{i=1}^{n}\{c_i, d_i\})^* \longrightarrow V^*, & h_2(a) = a \text{ for } a \in V, \\
& h_2(c_i) = x_i \text{ for } 1 \leq i \leq n, \\
& h_2(d_i) = y_i x_i z_i \text{ for } 1 \leq i \leq n
\end{array}$$

and the regular set

$$R = \bigcup_{i=1}^{n}(V^*\{c_i\}V^*\{d_i\}V^* \cup V^*\{d_i\}V^*\{c_i\}V^*).$$

It is easy to see that

$$Dupl_D(L) = h_2(h_1^{-1}(L) \cap R)$$

which proves the closure of $\mathcal{F}$ under duplications.

Since the erasing homomorphisms used in the proof of Theorem 3 i) are 1-restricted and trios are closed under restricted homomorphisms (see [13]), the statement follows for transpositions, too.

Now let $I = (V, \{x_1, x_2, \ldots, x_n\})$ be an inversion scheme. We consider the homomorphisms

$$\begin{array}{ll}
h_1 : (V \cup \{c_i \mid 1 \leq i \leq n\})^* \longrightarrow V^*, & h_1(a) = a \text{ for } a \in V, \\
& h_1(c_i) = x_i \text{ for } 1 \leq i \leq n, \\
h_2 : (V \cup \{c_i \mid 1 \leq i \leq n\})^* \longrightarrow V^*, & h_2(a) = a \text{ for } a \in V, \\
& h_2(c_i) = mi(x_i) \text{ for } 1 \leq i \leq n
\end{array}$$

and the regular set

$$R = \bigcup_{i=1}^{n} V^*\{c_i\}V^*$$

and obtain

$$Inv_I(L) = h_2(h_1^{-1}(L) \cap R)$$

which proves the closure of $\mathcal{F}$ under inversion. □

Corollary 5 *All families in the Chomsky hierarchy are closed under duplications, transpositions and inversions.*

We now start the study of closure under iterated versions. The following lemma is a helpful tool.

Lemma 6 *Every family of languages closed under iterated inversions or iterated transpositions is closed under permutations.*

Proof. For any language $L \in V^*$ let us construct the inversion scheme $I = (V, \{ab \mid a, b \in V, a \neq b\})$ and the transposition scheme $T = (V, \{(a, \lambda, b), (a, b, \lambda) \mid a, b \in V\})$. The relations

$$Inv_I^*(L) = Tr_T^*(L) = Perm(L)$$

follow immediate. □

Theorem 7 *The family of regular languages is not closed under iterated inversions, iterated transpositions and iterated duplications.*

Proof. Since the family of regular languages is not closed under permutations, the nonclosure with respect to iterated inversions and iterated transpositions follows by Lemma 6.

In order to prove the non-closure under iterated duplications we consider the regular language L consisting of the only word *abab* and the duplication scheme $D = (\{a, b\}, \{(ab, a, b)\})$. It is easy to see that

$$Dupl_D^*(L) = \{a^n b^n a^m b^m \mid n \geq 1, m \geq 1\}$$

which is not a regular language. □

Theorem 8 *The family of context-free languages is closed neither under iterated inversions nor under iterated transpositions.*

Proof. Because the family of context-free languages is also not closed under permutations, the statement follows by Lemma 6, again. □

It remains as an *open problem* whether or not the family of context-free languages is closed under iterated duplications.

Theorem 9 *The families of context-sensitive and recursively enumerable languages are closed under iterated inversions, iterated transpositions and iterated duplications.*

Proof. Let L be a context-sensitive language generated by the context-sensitive grammar $G = (N, T, S, P)$ and let (V, I) be an inversion scheme. We construct the context-sensitive grammar $G' = (N', T, S, P')$, where

$$\begin{aligned} N' &= N \cup \{X_a \mid a \in T\}, \\ P' &= \{X_{a_1}X_{a_2}\ldots X_{a_k} \longrightarrow X_{a_k}\ldots X_{a_2}X_{a_1} \mid a_1a_2\ldots a_k \in I\} \\ &\quad \cup \{X_a \longrightarrow a \mid a \in T\} \\ &\quad \cup \{h(\alpha) \longrightarrow h(\beta) \mid \alpha \longrightarrow \beta \in P,\} \end{aligned}$$

and $h : (N \cup T)^* \longrightarrow N'^*$ is the homomorphism given by

$$h(A) = A \text{ for } A \in N \quad \text{and} \quad h(a) = X_a \text{ for } a \in T.$$

The equality $L(G') = Inv^*(L)$ can be easily checked.

We are going to prove that $Tr^*(L)$ is a context-sensitive language for any transposition scheme $(T, \{(x_i, y_i, z_i) \mid 1 \leq i \leq n\})$, $n \geq 1$. To this end, we construct a phrase-structure grammar $\bar{G}$ working in the following way. A string of the form X_iw, with $w \in L$ and $1 \leq i \leq n$ is firstly generated. The symbols X_i scan the string w from left to right in order to perform a transposition rule (x_i, y_i, z_i). During this process two situations may occur. In the first one, the substring x_i is identified in w, it is erased, and the substring y_iz_i is looked for further. If this substring is identified, then x_i is inserted between y_i and z_i and the current scanning symbol becomes Y.

The second situation assumes that the substring y_iz_i is firstly identified and then the substring x_i. Now the process may be iterated arbitrarily many times, afterwards the scanning symbols are erased. With the above explanations we infer that $L(\bar{G}) = Tr^*(L)$. Since the grammar $\bar{G}$ has a linear bounded working space, it follows that $Tr^*(L)$ is a context-sensitive language (see[13]).

The closure of the recursively enumerable languages class follows immediately. By a similar proof one can show the closure under iterated duplications. □

Finally we remark that in this paper the three operations inversion, transposition and duplication have been studied isolated from each other as this was done in the papers on splicing. However, if we want to model the evolution it is necessary to consider schemes which contain rules for inversion as well as for transposition, duplication, splicing and deletion. It remains to investigate operations based on such schemes. A grammatical approach in this direction is presented in[4] as well as[15].

References

1. V. Brendel, H. G. Busse, Genome Structure Described by Formal Languages, *Nucleic Acids Res.*, **12**, 2561(1984).
2. J. Collado-Vides, The Search for Grammatical Theory of Gene Regulations is Formally Justified by Showing the Inadequacy of Context-free Grammars, *CABIOS,* **7**, 321(1991).
3. N. G. Copeland et al. A Genetic Linkage Map of the Mouse: Current Applications and Future Prospects. *Science*, **262**, 57(1993).
4. J. Dassow, V. Mitrana. A Grammatical Model for Genome Evolution: Evolutionary Grammars. *Proc. GCB'96*, 1996.
5. J. Dassow, V. Mitrana. Self Cross-over Systems. Submitted, 1995.
6. D. J. McGeoch. Molecular Evolution of Large DNA Viruses of Eukaryotes. *Seminars in Virology*, **3**, 399(1992).
7. T. Head, Formal Language Theory and DNA: An Analisys of the Generative Capacity of Specific Recombinant Behaviours, *Bull. Math. Biology*, **49**, 737(1987).
8. T. Head, Gh. Păun, D. Pixton, Language Theory and Molecular Genetics, a chapter in the forthcoming *Handbook of Formal Languages* (G. Rozenberg, A. Salomaa, eds.).
9. S. Karlin, E. S. Mocarski, G. A. Schachtel. Molecular Evolution of Herpesviruses: Genomic and Protein Comparisons. *J. of Virology*, **68**, 1886(1994).
10. V. Mitrana, Crossover Systems: A Language-theoretic Approach to DNA Recombination. Submitted, 1995.
11. Gh. Păun, The Splicing as an Operation on Formal Languages, *Proc. of INBS-IEEE Conference*, Washington, 176(1995).
12. D. Pixton, Regularity of Splicing Languages, *Discrete Appl. Math.*, in press.
13. A. Salomaa, *Formal Languages*, Academic Press, New York, 1973.
14. D. Sankoff et al. Gene Order Comparisons for Phylogenetic Inference: Evolution of the Mitochondrial Genome. *Proc. Natl. Acad. Sci. USA*, **89**, 6575(1992).
15. D. B. Searls, The Computational Linguistics of Biological Sequences. In *Artificial Intelligence and Molecular Biology* (L. Hunter ed.), AAAI Press, The MIT Press, 47(1993).