Applied Mathematics, 2012, 3, 188-197
http://dx.doi.org/10.4236/am.2012.32030 Published Online February 2012 (http://www.SciRP.org/journal/am)

Model of Overlapping Messages with Degenerate Coding

Valery Kirzhner[1], **Zeev Volkovich**[2]
[1]Institute of Evolution, University of Haifa, Haifa, Israel
[2]Software Engineering Department, ORT Braude College of Engineering, Karmiel, Israel
Email: valery@research.haifa.ac.il, vlvolkov@braude.ac.il

Received September 29, 2011; revised November 26, 2011; accepted December 4, 2011

ABSTRACT

Superposition of signals in DNA molecule is a sufficiently general principle of information coding. The necessary requirement for such superposition is the degeneracy of the code, which allows placing different messages on the same DNA fragment. Code words that are equivalent in the informational sense (*i.e.*, synonyms) form synonymous group and the entire set of code words is partitioned into synonymous groups. This paper is dedicated to constructing and analyzing the model of synonymous coding. We evaluate some characteristics of synonymous coding as applied to code words of length two although many definitions may be extended for words of arbitrary length.

Keywords: Overlapping Messages; Degenerate Coding; Sequence Compacting; De Bruijn Graph; DNA Triplet Code

1. Introduction

The article discusses the peculiarities of degenerate codes with the emphasis on their use for creating "overlapping messages". This study was motivated by the existence of such type of messages in DNA-protein coding. In what follows, we will describe some features of the DNA triplet code which are relevant for our study.

It is well known that the information contained in DNA is written using a 4-letter alphabet A, T, C, G. There exist a total of 64 3-letter words over this alphabet, encoding 20 amino acids. Consequently, the code is degenerate, *i.e.*, the same amino acid can be encoded by several alternative words. A group that contains all possible words encoding the same amino acid is referred to as a *synonymous group*. Any finite length succession of code words can be viewed as a message of DNA sequence. Such message is decoded starting from the first letter (start position), each successive triplet being transformed into the corresponding amino-acid. Usually, there exists only one start position to make a sensible message. However, in some cases the choice of another start position (e.g., shifted by one letter) also results in a sensible message. This is the situation of overlapping messages, which is quite common in genes [1-3] and there exist hundreds of overlapping pairs of protein coding genes in vertebrate genomes [4]. In particular, the start position of overlapping genes may be shifted by 1 or 2 positions, which is the case, e.g., in virus genomes [5-7].

In this study, the simplest coding model is suggested for a degenerate code with overlapping messages. This model, despite its simplicity, gives the opportunity to investigate the described above biological mechanisms. In particular, we evaluate the effectiveness of creating overlapping messages depending on the type of synonymous partitioning and assess the effectiveness using a specially constructed sequence. We obtain these characteristics for code words of length 2; however, some of the definitions given here can be extended also to words of arbitrary length [8].

Assume that there exists alphabet $\mathcal{A}_s$ where $s = |\mathcal{A}_s|$. Let the code words be the words of length 2 over this alphabet and let them belong to set $\mathcal{K}$. Set $\mathcal{K}$ does not necessarily coincide with the entire set of the words of length 2 over alphabet $\mathcal{A}_s$. Further, let $\mathcal{P}(\mathcal{K})$ be a certain partition of set $\mathcal{K}$ into non-empty subsets. Assume that, being assigned to each subset, there exists only one message, which can be transmitted via any element of this subset. Thus, the elements of each subset are *synonyms*, and any partition $\mathcal{P}(\mathcal{K})$ will be referred to as a *synonymous partition*. Let us consider a finite sequence of letters over alphabet $\mathcal{A}_s$, which we define as message M. We assume that a hypothetical decoding "device" reads the message word by word (the length of the words is 2), starting from the first position. As a result, an ordered sequence of separated words of length 2 is obtained. This "construction" is equivalent to the triplet coding and translation of protein.

Extending this analogy, we can note that decoding with a shift of the reading frame by 1 also creates a certain sequence of words of length 2. In the context of this model, we define both sequences as "functional" if the union of the words, constituting both sequences, contains just one word from each synonymous groups and does not contain non-code words of length 2. The latter condition is important in the situation where not all the words of length 2 over a given alphabet are code words. In other words, neither "functional" sequences contain synonymous words nor do they have common synonymous words.

We will call message M, considered above, a *dense sequence*. Respectively, a closed dense sequence will be called a *dense contour*. The mathematical methods that we are going to use lead to practically the same results in the cases of dense sequence or dense contour, but hereafter it will be more convenient to formulate propositions for a dense contour, rather than for a dense sequence. In this study, we investigate the existence criteria for a dense contour (dense sequence) for different synonymous partitions. In Section 2 we present the simplest examples of the connection between the composition of synonymous groups and the possibility to create a dense message. Cartesian synonymous partitions are introduced, which also may be used in the description of a standard triplet code [8]. These partitions comprise only a subset of all possible partitions, yet even in this simple case quite non-trivial properties are observed, which are the subject of the present study.

In the case of Cartesian partitions a new alphabet can be defined, in which each synonymous group may be substituted by a single word (in our case, of length 2). This alphabet is not a standard one, in the sense that its different letters are allowed to be superposed when they are used for creating a sequence. The rules of superposition are established by a special table of correspondence. Actually, each element of such an alphabet is a multivalued function, whose values belong to the original alphabet. The description of this alphabet, the main operational rules and the concept of a dense sequence over this alphabet are presented in Section 3. In the same section the main theorems on the existence of dense sequences are evaluated in the terms of abstract alphabet.

In Section 4, are specialized for the case of Cartesian synonymous partitions.

2. The Simplest Properties of Synonymous Partitions. Cartesian Synonymous Partitions

2.1. Example of a Synonymous Partition

Let us examine the simplest example of a synonymous partition. Let $s = 2$ $\left(\mathcal{A}_2 = \{A, B\}\right)$. There are four two-letter words $\{AA, AB, BA, BB\}$. Let us break this set arbitrarily into two non-empty synonymous classes, for example, $\{AA\}$ and $\{AB, BA, BB\}$. Our task is to determine if a dense sequence or contour exist for this partition. Since there are only two classes, the sequence must contain two words with the shift of 1, *i.e.* have length 3. It is obvious that the word AAB is the desired sequence, since it contains the two-letter word AA from the first synonymous set $\{AA\}$ and the word AB from the second set $\{AB, BA, BB\}$. There is one additional sequence BAA with the same properties: BA belongs to the second synonymous set and AA to the first one. Thus, there are two dense sequences for this synonymous partition (moreover, the synonym BB is not used), and packing by a contour is impossible at all. Indeed, such a contour would have to consist of two letters, which necessarily belonged to AA the only word from a synonymous group. But, obviously, this contour contains no words from the second synonymous group.

Let us examine now all possible synonymous partitions into two groups in this example. There are 7 such combinations: 5 partitions:

$$\begin{gathered}\{AA\},\ \{AB,\ BA,\ BB\};\ \{AB\},\ \{AA,\ BA,\ BB\};\\ \{AA,\ BA\},\ \{AB,\ BB\};\ \{AA,\ AB\},\ \{BA,\ BB\};\\ \{AA,\ BB\},\ \{AB,\ BA\};\end{gathered} \tag{2.1}$$

plus those obtained from (2.1) by transposition of symbols A and B (The last three partitions are preserved under this transposition of letters.) It is sufficient to solve our problem only for partitions in (2.1). In the partition $\{AB\}$, $\{AA,\ BA,\ BB\}$ there are four complete dense sequences: AAB, ABA, ABB and BAB. In this case all synonyms of both groups are used. There is also a contour *AB*, which includes two words *AB* and *BA* from different synonymous partitions. We can say that this synonymous partition is more effective than the first one in terms of quantity of possible versions of the dense packing. Further, for the partition $\{AA, BA\}$, $\{AB, BB\}$ there are four dense sequences: AAB, BAB, ABA, BBA and one contour AB. For the partition $\{AA, BB\}, \{AB, BA\}$ there are words AAB, BAA, ABB, BBA, and no contour. For the partition $\{AA, AB\}$, $\{BB, BA\}$ we have words BAA, ABB, ABA, BAB and a contour AB.

Our analysis of this example is now complete. We can see that for any synonymous partition into two sets there exist dense sequences, and the number of such sequences depends on the partition and varies from 2 to 4.

Statement 2.1. Let the set of code words $\mathcal{K}$ include all possible words of length 2 for any alphabet $\mathcal{A}_s$ where $s \geq 2$. Then for any partition into three non-empty synonymous groups there exists at least one dense sequence.

Proof. Let us prove that for $s \geq 5$ it is possible to select such four symbols of the alphabet $\mathcal{A}_s$ that the intersection of the set of the words over these symbols with each of three synonymous sets is not empty. In other words, the contraction of synonymous partition into this alphabet, reduced to the 4 symbols, also has three non-empty synonymous groups. Let us place the symbols of the alphabet $\mathcal{A}_s$ at the integral points of the X and Y axes of rectangular system of coordinates (Let us call such points symbolic). Then the words of length two are the corresponding points of Cartesian plane. Let us consider the following cases:

- Case 1: For any $A \in \mathcal{A}_s$ all words BA are synonyms. Then for any $B \in \mathcal{A}_s$ there exist $A_1, A_2, A_3 \in \mathcal{A}_s$ such that BA_1, BA_2, BA_3 belong to different synonymous groups. Four symbols: B, A_1, A_2, A_3 (see **Figure 1**).
- Case 2: There exists $A \in \mathcal{A}_s$ such that BA and CA are not synonyms, and DA is synonymous either to BA or to CA for any $D \in \mathcal{A}_s$. Let A_1A_2 be not synonymous to BA and CA. Then A_1A_2 is not synonymous to A_1A which is not synonymous to either BA or CA. Four symbols: A_1, A_2, A, B (or C) (see **Figure 2**).
- Case 3: There exists $A \in \mathcal{A}_s$ such that BA, CA and DA are not synonymous, and EA is synonymous to one of them for any $E \in \mathcal{A}_s$. Four symbols: B, C, D, A (see **Figure 3**).
- Thus, it remains to prove Statement 2.1 for cases where $s = 2, 3, 4$. This task was solved by sorting out all the possibilities on the computer. It is shown that set $\mathcal{S}$ is not empty in all cases for all nontrivial partitions into three synonymous groups. □

Further analysis of the number of synonymous partitions only in terms of *s* is ineffective. Indeed, let $s = 3\left(\mathcal{A}_3 = \{A, B, C\}\right)$. There exist 9 two-letter words. Let us examine the partition $\{AA\}$, $\{BB\}$, $\{CC\}$ and {everything else}. Here we have 4 synonymous groups. Three groups consist of one word each and cannot be connected directly. It means that for their connection it is necessary to use at least two words from the fourth group, so that the obtained word will have not less than two synonyms. Therefore, there is no dense sequence for this partition.

2.2. Two-Dimensional Cartesian Synonymous Partitions

Assume that there is given a finite alphabet $\mathcal{A}_s = \{\mathrm{a}_1, \cdots, \mathrm{a}_s\}$. We will use a Cartesian plane to represent all possible two-letter words over this alphabet. Let us arrange the letters of the alphabet $\mathcal{A}_s$ on the axes of the rectangular two-dimensional coordinate system, say, in the ascending order of numbers. Then the set of all possible two-letter words $\mathcal{K}$ can be identified with the set of points (i, j) $(1 \le i, j \le s)$ on the plane.

Example **2.2.1.** Let $\Omega = \{\mathrm{a}_1, \cdots, \mathrm{a}_r\}$ be a partition of the set $\mathcal{A}_s$ into the pairs of disjoint sets (classes) a_r $(1 \le r \le s; \mathrm{a}_i \mathrm{a}_j = \varnothing$, if $i \ne j$ and $\mathrm{a}_i \in \mathcal{A}_s$ for $i = 1, \cdots, r)$. Let us define a new alphabet. We will call the set a_r a *quotient letter*, the set Ω the *alphabet*, and the set $\mathcal{A}_s$ in this context, the *basic alphabet*. The above partition induces the partition $\mathcal{P}(\mathcal{K})$ of all code words into the groups. All words obtained by the direct product of a_i by a_j belong to one group. We will assume that each such group is synonymous, *i.e.* all code words included into this group contain the same message. It is convenient to denote synonymous group as a pair $(\mathrm{a}_i, \mathrm{a}_j)$ (see **Figure 4**). Let us call this pair of classes a *quotient word*. Let us define now the rules of the formation of the sequences of quotient words.

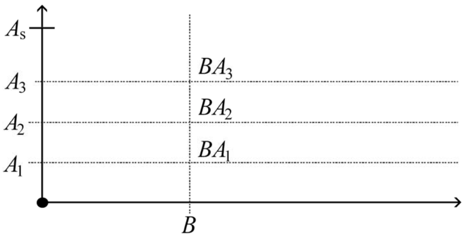

Figure 1. Geometrical illustration of proof of Statement 2.1, Case 1.

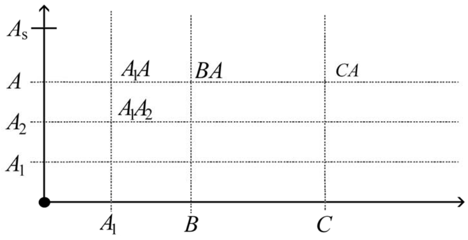

Figure 2. Geometrical illustration of proof of Statement 2.1, Case 2.

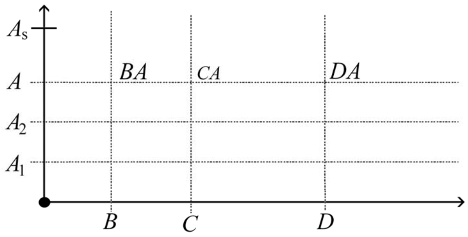

Figure 3. Geometrical illustration of proof of Statement 2.1, Case 3.

Namely, word $(\mathrm{a}_i, \mathrm{a}_j)$ follows the word $(\mathrm{a}_u, \mathrm{a}_v)$, if class a_v is equal to class a_i. Let us write down this chain as

$$(\mathrm{a}_u, \mathrm{a}_v)(\mathrm{a}_v, \mathrm{a}_j) \tag{2.2}$$

Clearly, such definition is coherent with the standard one for the sequence of letters. Let *x* be any element of the class a_v. Then (2.2) can be understood as any triplet, where and are any elements of the corresponding sets $(\mathrm{a}_u, \mathrm{a}_j)$. Let us call such set of sequences of letters matched with the chain of quotient words the *projection of the chain of quotient words*. Further, by analogy for

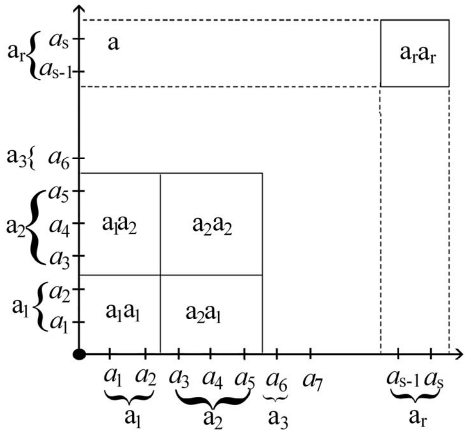
$\mathrm{a}_r\{a_s, a_{s-1}$
a
$\mathrm{a}_r\mathrm{a}_r$
$\mathrm{a}_3\{a_6$
$\mathrm{a}_2\{a_5, a_4, a_3$
$\mathrm{a}_1\mathrm{a}_2$
$\mathrm{a}_2\mathrm{a}_2$
$\mathrm{a}_1\{a_2, a_1$
$\mathrm{a}_1\mathrm{a}_1$
$\mathrm{a}_2\mathrm{a}_1$
a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_{s-1} a_s
a_1 a_2 a_3 a_r

Figure 4. Partition of the set $\mathcal{A}_s$ into the pairs of disjoint sets a_r; a pair (a_i, a_j) is a synonymous group.

the quotient words $(\mathrm{a}_v, \mathrm{a}_j)$ and $(\mathrm{a}_p, \mathrm{a}_q)$ sets a_j and a_p coincide and thus we can write down sequence of three words of the classes

$$(\mathrm{a}_u, \mathrm{a}_v)(\mathrm{a}_v, \mathrm{a}_j)(\mathrm{a}_j, \mathrm{a}_q). \tag{2.3}$$

Obviously, the projection of this chain consists of all possible words of the form, where *y* is any element from the set a_j, selected independently of x.

In the standard situation the alphabet Ω coincides withthe initial alphabet $\mathcal{A}_s = \{\mathrm{a}_1, \cdots, \mathrm{a}_s\}$, when the partition is trivial $(\mathrm{a}_i \equiv a_i)$. Thus, in this situation we can use a Cartesian plane too.

***Example* 2.2.2.** Let $\Omega_1 = \{\mathrm{a}_1, \cdots, \mathrm{a}_r\}$ and $\Omega_2 = \{\mathrm{b}_1, \cdots, \mathrm{b}_h\}$ be two partitions of the set $\mathcal{A}_s$. The pair $(\Omega_1\Omega_2)$ induces the partition of all code words into the groups as in the previous Example 2.1: all words obtained by the direct product of the set a_i by the set b_j belong to the same group. We will also assume that each such group is synonymous and will denote it as the quotient word $(\mathrm{a}_i, \mathrm{b}_j)$. Let us define now the rules of the formation of the sequences in the set of classes. Namely, word $(\mathrm{a}_i, \mathrm{b}_j)$ follows word $(\mathrm{a}_u, \mathrm{b}_v)$ if the intersection of the second element of the latter word (class b_v) with the first element of the former word (class a_i) is not empty. Let us write down this chain as

$$(\mathrm{a}_u, \mathrm{b}_v)(\mathrm{a}_i, \mathrm{b}_j), \mathrm{b}_v \cap \mathrm{a}_i \neq \varnothing \tag{2.4}$$

It is easy to see that such definition is coherent with the standard one for the sequence of letters. Namely, let *x* be any element from the intersection of sets b_v and a_i. Then (2.4) can be understood as any triplet, where $\alpha \in \mathrm{a}_u, \beta \in \mathrm{b}_j, x \in (\mathrm{b}_v \cap \mathrm{a}_i)$. Further, by analogy for the quotient words $(\mathrm{a}_i, \mathrm{b}_j)$ and $(\mathrm{a}_p, \mathrm{b}_q)$ we can write down a sequence of three quotient words

$$(\mathrm{a}_u, \mathrm{b}_v)(\mathrm{a}_i, \mathrm{b}_j)(\mathrm{a}_p, \mathrm{b}_q) \tag{2.5}$$

if the intersection of sets b_j and a_p is non-empty. It is obvious that the projection of this chain consists of all possible words of the form, where y can be any element from the intersection of sets b_j and a_p selected independently of x.

By definition, for any basic alphabet $\mathcal{A}_s$, a *Cartesian synonymous partition* is a partition of $\mathcal{A}_s\mathcal{A}_s$ into pairs of type $\mathrm{a}_i\mathrm{b}_j$ where $\mathrm{a}_i, \mathrm{b}_j$ are some subsets of the letters of the basic alphabet. We call such partitions *regular* Cartesian synonymous partitions, if as in the previous examples sets $\{\mathrm{a}_i\}$ and $\{\mathrm{b}_i\}$ form the partition of the letters of the basic alphabet $\mathcal{A}_s$.

***Example* 2.2.3.** Let a set of the code words over the alphabet $\mathcal{A}_s$ be divided into "rectangles", which do not intersect and cover the entire Cartesian square $\mathcal{A}_s\mathcal{A}_s$. The projections of the sides of these rectangles on *X* and *Y* axes form on these axes, generally speaking, two systems of intersecting intervals such that their union on each axis, obviously, is equal to $\mathcal{A}_s$. Let us denote both these systems as earlier $\Omega_1 = \{\mathrm{a}_1, \cdots, \mathrm{a}_r\}$ and $\Omega_2 = \{\mathrm{b}_1, \cdots, \mathrm{b}_h\}$. In this case, however, sets $\mathrm{a}_i, \mathrm{b}_j$ can intersect and the amount of sets *r* and *h* in each system can be greater than the length of the initial alphabet s. For example, let us examine a partition containing pairs $(\mathrm{a}_i, \mathrm{b}_1)$ with $\mathrm{i} = 1, \cdots, s$ and square $(\mathrm{a}_2, \mathrm{a}_3)(\mathrm{b}_2, \mathrm{b}_3)$ (see **Figure 5**). There exist already $s+1$ intervals on the *X*-axis. Nevertheless, it is possible to describe each "rectangle" (a synonymous set), as earlier, by an appropriate pair $(\mathrm{a}_i, \mathrm{b}_j)$ or by a single quotient word. The definition of a sequence of quotient words is transferred from Example 2.2.2. It is obvious that this is an *irregular Cartesian synonymous partition*.

3. Abstract Dense Sequences

The *quotient letters* introduced above (Example 2.2.1) may be viewed as letters belonging to some alphabet and possessing a definite inner structure. As a counter example, we will also consider the *abstract alphabet*, *i.e*. an alphabet without any assumptions about the nature of origins of its letters. In this chapter we investigate the problem of constructing a dense sequence in terms of the abstract alphabet. This problem is close to standard for combination theory of words (see, for example, [9]), however differs in some details. This more generalized approach allows us to include into consideration not only Cartesian synonymous partition with the known mechanism of projecting abstract (quotient) letters onto the letters of the initial alphabet, but also other potential methods of mapping a correspondence between the signals as they overlap.

Analysis of Cartesian synonymous partition demonstrates that it is natural to consider two abstract alphabets. Indeed, different alphabets already appeared in examples

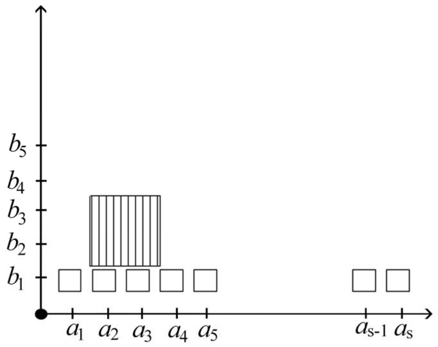

Figure 5. Pairs (a_i, b_1) with $i = 1,\cdots,s$ and a square (a_2, a_3) (b_2, b_3).

of Cartesian synonymous groups. For example, the collections of sets $\Omega_1 = \{a_1,\cdots,a_r\}$ and $\Omega_2 = \{b_1,\cdots,b_h\}$ in Example 2.2 can be considered as two different alphabets for constructing the quotient words. Letters of the first alphabet can stand only at the beginning of code word, and letters of the second alphabet can stand only at the end of the word. The rules of the construction of sequences of code words in these examples were connected with the origin of letters of the both alphabets, while these alphabets were the subsets of a certain initial set.

In line with this example, let us immediately define two different alphabets $\Omega_1 = \{a_1,\cdots,a_r\}$ and $\Omega_2 = \{b_1,\cdots,b_h\}$. That is, code word will be any pair (ab) of letters such that its first element (letter) belongs to the alphabet Ω_1 and the second one belongs to the alphabet Ω_2. To construct a sequence of such words it is necessary to define a table of correspondence of the letters of different alphabets. We define this correspondence using a many-valued mapping φ: set $\varphi(a_i)$ consists of the letters of the alphabet Ω_2, which are compatible with the symbol a_i. Let us call this mapping a *table of correspondence of symbols*. Note that in the case of Cartesian synonymous partitions, when the quotient letters are subsets of a certain set, the table of correspondence of symbols is defined by the relationship (2.4), *i.e.* by the non-empty intersection of the corresponding subsets. In our abstract case the table of correspondence of symbols makes possible to form sequences of words of the form

$$\cdots(a_i,b_j)(a_s,b_t)\cdots \tag{3.1}$$

If $b_j \in \varphi(a_s)$. Recall that in the case of the abstract alphabet the sequence is composed of the words following one another whereas the letters standing next to each other must satisfy the table of correspondence. In this case, a dense sequence is a sequence of words of the form (3.1), where each word is encountered exactly once. A dense contour is defined in a similar fashion.

Let us analyze sequences of words over the alphabets Ω_1 and Ω_2 with the help of the following bipartite graph G. The sets of vertices σ^- and σ^+ of this graph correspond to the letters of the alphabets Ω_1 and Ω_2, respectively. Arcs, leaving the vertices of the set σ^- and entering the vertices of the set σ^+, correspond to the code words. An arc connecting the vertex of the set σ^- associated with a letter $a \in \Omega_1$ with the vertex of the set σ^+ associated with a letter $b \in \Omega_2$, corresponds to the word (ab). We call these arcs *arcs of words*. Let us denote the set of all arcs of words by V. Arcs of another kind connect the vertices of the set σ^+ with the vertices of the set σ^-. These arcs are defined by the table of correspondence of symbols. Namely, if $b \in \varphi(a)$ then there is an arc leaving the vertex of the set σ^+, associated with a letter $b \in \Omega_2$ and entering the vertex of the set σ^- associated with a letter $a \in \Omega_1$. Let us call these arcs *arcs of recovery*. We denote the set of the arcs of recovery by U. Several arcs of recovery can leave each vertex, including none, depending on the table of correspondence.

Example 3.1. There are given two different alphabets $\Omega_1 = \{a_1,a_2,a_3,a_4\}$ and $\Omega_2 = \{b_1,b_2,b_3,b_4,b_5\}$ and a table of correspondence (Table 1). Let us assume that the code words are a_1b_1, a_2b_1, a_3b_3, a_3b_4, a_3b_5 and a_4b_2. Therefore, not all arcs of words should be drawn in the corresponding bipartite graph. **Figure 6** illustrates the corresponding bipartite graph G, where thin arrows depict arcs of words, and thick arrows depict arcs of recovery.

There exists a path in graph G that traverses all of the arcs of words only once. It may be written as a sequence of arcs

$$\begin{gathered}(a_1,b_1)\langle b_1,a_3\rangle(a_3,b_3)\langle b_3,a_2\rangle(a_2,b_1)\langle b_1,a_3\rangle \\ (a_3,b_4)\langle b_4,a_4\rangle(a_4,b_2)\langle b_2,a_3\rangle(a_3,b_5)\end{gathered} \tag{3.2}$$

where the symbol "$(\ )$" denotes arc of word and "$\langle\ \rangle$" —arc of recovery. Note that in construction of the above sequence the arc of recovery b_1a_3 was traversed twice.

Thus, the sequence of arcs (3.2) constitutes an Euler path conditional on the arc of recovery b_1,a_3 being of degree two. Furthermore, the arc of recovery $\langle b_1,a_3\rangle$ means that the words (a_1,b_1) and (a_3,b_3) may form a sequence $(a_1,b_1)(a_3,b_3)$ according to definition (3.1). The same is true for all the other arcs of recovery in the sequence of arcs (3.2). Hence, the sequence of words

$$(a_1,b_1)(a_3,b_3)(a_2,b_1)(a_3,b_4)(a_4,b_2)(a_3,b_5) \tag{3.3}$$

follows from the sequence of arcs (3.2). The sequence (3.3) contains all the code words only once and does not contain any other words. This is a dense sequence by definition. Note that in contrast to the standard De Bruijn graph, arcs of recovery do not correspond to words, and they, in some sense, are intermediate steps, which are represented neither by letters nor by words in the formed

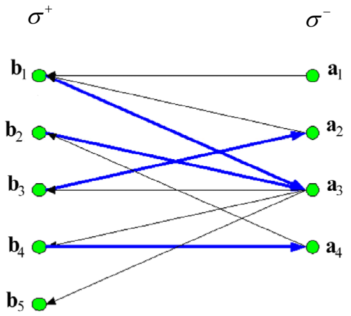

Figure 6. The bipartite graph *G* in Example.

sequence. Therefore, it is possible either to traverse the arc of recovery several times or not to traverse it at all, *i.e.*, these arcs (multi-arcs) can have an arbitrary non-negative multiplicity. The Table of correspondence (**Table 1**) of symbols defines the set U of the arcs of recovery; however, the multiplicities of these arcs can be assigned in several possible ways.

It follows from Good's Theorem ([10], Section 9) that for the existence of Euler cycle it is necessary and sufficient that the sum of the multiplicities of the entering arcs is equal to the sum of the multiplicities of the leaving arcs for all vertices of the graph G. Suppose that now the conditions of this theorem are violated in only two vertices of the graph. Specifically, the number of arcs with the first vertex as their initial vertex exceed by 1 than the number of arcs with the first vertex as their terminal vertex, and the reverse is true for the second vertex. It is easy to show that in this case there exists an Euler path which begins and ends with these two vertices. This fact reduces the problem of construction of a dense sequence to the problem of definition of the multiplicities of arcs of recovery with the given sets V and U, so that the graph G satisfies the Eulerian condition (see [10], Section 9). Naturally, only arcs of recovery may be assigned different multiplicities, because multiplicity of an arc of word always equals 1. In the Example 3.1, multiplicity of the arc $\langle b_1, a_3 \rangle$ should be set to 2, and multiplicities of all the other arcs of recovery should be set to 1. In this case, the number of incoming and outgoing arcs is the same (taking multiplicity into account). The number of arcs with the vertex a_1 as their initial vertex is greater than the number of arcs with a_1 as their terminal vertex by 1, and the reverse is true for the vertex b_5. Hence, the conditions for the existence of an Euler path in this graph are fulfilled.

Let us investigate the problem of construction of a dense sequence that contains all possible words over the alphabets Ω_1 and Ω_2 exactly once using graph G. Let the alphabets Ω_1 and Ω_2 have the same cardinality $(r = h)$. In this case d arcs leave each vertex of the set σ^-. Let $W \subseteq \sigma^-$. Let us denote the set of vertices in $\{\sigma^+\}$ adjacent to W via arcs of recovery by $S(W)$.

Table 1. Table of correspondence between the letters of alphabets Ω_1 and Ω_2.

$a_i \in \Omega_1$	$\varphi(a_i) \in \Omega_2$
a_1	-
a_2	b_3
a_3	b_1, b_2
a_4	b_4

Theorem 3.1. Let $d = |\sigma^-| = |\sigma^+|$. For the existence of a cyclic sequence of words including each word exactly once it is necessary and sufficient that for any set $W \subseteq \sigma^-$ the following inequality holds:

$$|S(W)| \geq |W|$$

Proof. Necessity. Since all arcs of words are drawn in this graph, each vertex of the set σ^- has the out-degree equal to d. For the same reason, each vertex of the set σ^+ has the in-degree also equal to d. Let us examine an arbitrary set $W \subseteq \sigma^-$. In order that the out-degrees and the in-degrees of all vertices of this set would be equal, it is necessary that the sum of multiplicities of the arcs of recovery of the set $S(W)$ incident to W would be equal $|W|d$. However, the vertices of the set σ^+ also must have equal out-degrees and in-degrees, therefore the multiplicity of the arc of recovery is limited by value d. (Multiplicity of the arc of recovery can be less than d, if there is more than one arc of recovery leaving that vertex.) Therefore, the sum of multiplicities of the arcs of recovery does not exceed the value of $|W|d$, hence it is necessary that $|S(W)|d \geq |W|d$.

Sufficiency. Conditions of this theorem coincide with the conditions of the Hall's theorem [10] of existence of perfect matching in the set of arcs U, *i.e.* matching, containing all vertices of graph. Let P be such a matching. Let us assume that the multiplicities of all arcs not in P equal zero, and the multiplicities of arcs in P equal d. It is obvious that in this case the out-degrees and the in-degrees of all vertices in G are equal. Therefore, an Euler cycle exists. The Theorem is proven. □

Note that the matching mentioned in the proof of Theorem 3.1 defines a one-to-one correspondence between letters of both alphabets. In this case it is possible to return to the initial situation when the first and the second letters of each word belong to the same alphabet. In particular, it is possible to identify the corresponding vertices of the graph G from σ^- and σ^+ and thus to obtain the standard De Bruijn graph.

Consider now the general case, when the alphabets Ω_1 and Ω_2 are of different cardinality $(r \neq h)$ and the set of the code words *M* does not necessarily coincide with the set of all possible words. It means that the sets of left and right vertices have different size, $\left|\sigma^-\right| \neq \left|\sigma^+\right|$, and not all arcs of words are drawn in the graph *G*. In this case let us denote the out-degree of vertex $\mathrm{i} \in \sigma^-$ by s_i and the in-degree of vertex $\mathrm{i} \in \sigma^+$ by t_i. It is obvious that

$$\sum s_\mathrm{i} = \sum t_\mathrm{i} \tag{3.4}$$

Let us define the *generalized out-degree* of the vertex as the number of arcs leaving it, each taken with its multiplicity and the *generalized in-degree* of the vertex as the number of arcs entering it, each taken with its multiplicity. Let us denote an arbitrary subset of vertices of graph *G* by *X* and the sum of generalized out-degrees of these vertices by $P(X)$. Let us denote, as earlier, the set of vertices in σ^- by W and the set of vertices in σ^+ that are adjacent to W via arcs of recovery, by $S(W)$.

Theorem 3.2. *For the existence of a cyclic sequence of words including each word from the set* M *exactly once it is necessary and sufficient that the graph* G *is connected, and for any set* $W \subseteq \sigma^-$ *the following inequality holds:*

$$P(S(W)) \geq P(W) \tag{3.5}$$

Proof. Necessity. *Indeed*, some of the arcs of recovery leaving the set *S*(*W*), possibly all of them, enter the vertices of the set *W*, but no other arc of recovery enters this set according to the definition of the set *S*(*W*). If graph G is a generalized Euler graph, *i.e.* each vertex has equal generalized out-degree and generalized in-degree, then the following inequality holds: $P(S(W)) \geq P(W)$.

Sufficiency. It is necessary to show that if (3.5) holds then there exist non-negative integral values on the arcs of recovery (multiplicities) such that the generalized out-degree and the generalized in-degree at each vertex of the graph G are equal. We apply the theory of flows in networks. Let us assume that for every *i* the *capacity of vertex* i *from the set* σ^- is equal to s_i (out-degree) and the *capacity of vertex* i *from the set* σ^+ is equal to t_i (in-degree). For convenience, let us add to the graph G two new vertices in the following way. Let us join all vertices of the set σ^- to a new vertex K by arcs directed to vertex K. Let us join a new vertex L to all vertices of the set σ^+ by arcs directed from vertex L. In the theory of network flows these vertices are conventionally called a *sink* and a *source* respectively. We will further examine only the arcs of recovery (set U) on the graph G discarding the arcs of words from the graph G, and denote the graph obtained in this way by G'. By construction, graph G' is oriented from the vertex L, the source, to the vertex K, the sink (see **Figure 7**).

In the network flow problem it is required to find a non-negative function on the arcs of the graph (called *the maximum flow*), possessing specific properties. By definition, the sum of the values of the flow function on all arcs incident to the sink is called the *flow value*. It is required that: 1) for each vertex, except for the sink and the source, the total value of the flow on all arcs entering the vertex and the total value on all arcs leaving the vertex must be equal to each other and must not exceed the capacity of the vertex, 2) it is not possible to increase the flow value without destroying condition (1). A vertex of graph is *saturated*, if the total value of the flow on entering (and, consequently, on leaving) arcs is equal to its capacity.

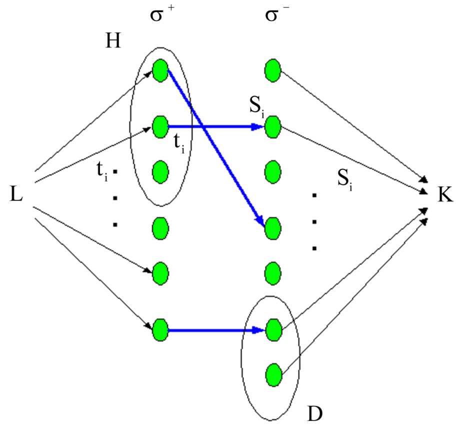

Figure 7. Graph G', oriented from the vertex L, the source, to the vertex K, the sink.

Let us show that if there is a flow of value $Q = \sum s_\mathrm{i}$ in the graph G', then all vertices of the sets σ^+ and σ^- are saturated. Indeed, according to the definition of the flow value, Q is equal to the sum of flows on all the arcs going from each vertex $i \in \sigma^-$ to the vertex K. However, the flow for each such arc does not exceed the capacity s_i of the corresponding vertex according to the property 1) of the flow. Consequently, the sum of all such flows does not exceed Q and is equal to Q only when the flow on arc is equal to the capacity of the vertex. When the flow value is Q, all the vertices of the set σ^- are saturated. Further, for any vertex $\mathrm{i} \in \sigma^-$ we will denote the sum of the flows on all arcs entering such a vertex by S_i. According to the property (1) of the flow, S_i is equal to the sum of the flows on all leaving arcs, *i.e.* $S_\mathrm{i} = s_\mathrm{i}$. Let us denote now the sum of flows on all arcs leaving any vertex $\mathrm{i} \in \sigma^+$ by T_i. According to the definition of the flow, $T_\mathrm{i} \leq t_\mathrm{i}$ *i.e.* $\sum T_\mathrm{i} \leq \sum t_\mathrm{i} = Q$ (see 3.4). On the other hand, it is clear that $\sum T_\mathrm{i} = \sum S_\mathrm{i} = Q$, since both sums are taken on the same set of arcs. From this follows that $T_\mathrm{i} = t_\mathrm{i}$ for any vertex $i \in \sigma^+$. Therefore, when the flow value is Q, all vertices of the set σ^+ are also saturated. It remains to assign flow t_i to each arc leaving vertex L and entering vertex $i \in \sigma^+$ of graph G'. In this case all the conditions of the flow are maintained.

We assume now that the value of flow Q not only exists, but also is integral, *i.e.* the values of the functions on the arcs of the graph are integral non-negative numbers. In this case for every arc of recovery its multiplicity must be defined as the value of flow on this arc. Since all vertices of the set σ^+ in graph G' are saturated, every vertex of the set σ^+ in the initial graph G has equal generalized out-degree and in-degree (number of arcs of words). The same is true for vertices of the set σ^- in the initial graph G. As it was proven earlier, the values of multiplicities are equal to the capacities of vertices, *i.e.* to the out-degrees for vertices of the set σ^- and to the in-degrees for vertices of the set σ^+.

With such multiplicities of the arcs of recovery the initial graph G is an Euler graph. It remains to prove that with the theorem condition the flow value Q exists and it is integral.

We call a set of vertices a *vertex-cut* (or just a *cut*) of the graph G' if it does not contain the sink and the source, and any path from L into K contains at least one vertex of this set. Let us call the sum of the capacities of the cut's vertices the *weight* of the cut. The cut is called *minimal* if it has the smallest weight. According to the Ford-Fulkerson theory [11], the value of the maximum flow is equal to the value of the minimal cut. Now we determine the value of the minimal cut in graph G'. In graph G' there are two obvious cuts: the entire set of vertices σ^- and the entire set of vertices σ^+. Both these cuts have an equal weight $P\left(\sigma^-\right)=P\left(\sigma^+\right)=Q$ (see Equality (3.4)). Let us demonstrate that the cuts σ^- and σ^+ are minimal. That is, the weight of any other cut is greater than or equal to Q. Consider a cut $\left(D,H\right)$, where $D\subset\sigma^-, H\subset\sigma^+$ respectively; its weight is equal to $P\left(D\right)+P\left(H\right)\cdot S\left(\sigma^-\setminus D\right)$ is a set of vertices in σ^+ that are adjacent to the vertices of $\sigma^-\setminus D$ via arcs of recovery and $S\left(\sigma^-\setminus D\right)\cap\left(\sigma^+\setminus H\right)=\varnothing$. Indeed, otherwise there would be an arc from the vertex of the set $\left(\sigma^+\setminus H\right)$, not included in the cut, into the vertex of the set $\sigma^-\setminus D$, also not included in the cut. This contradicts the definition of the cut. In other words, $S\left(\sigma^-\setminus D\right)\subset H$. Hence, taking into account the condition of the theorem $P\left(S\left(\sigma^-\setminus D\right)\right)^3 P\left(\sigma^-\setminus D\right)$, the following inequality holds:

$$P\left(D\right)+P\left(H\right)\ge P\left(D\right)+P\left(\sigma^-\setminus D\right)\ge$$
$$P\left(D\right)+P\left(\sigma^-\setminus D\right)=P\left(\sigma^-\right)=Q$$

Thus, the weight of the cut $P\left(D\right)+P\left(H\right)$ is not less than the value of Q. Therefore, the value of the maximum flow on graph G' is equal to Q.

The integrality of this flow easily follows from the standard algorithm of the construction of the maximum flow [11], which results in integral flow if the capacities of the vertices are integral numbers. The Theorem is proven. □

Corollary 3.1. Assume that in the graph G all arcs of words are drawn, and $\left(r\neq h\right)$. Then for the existence of a cyclic sequence of words that includes each word exactly once it is necessary and sufficient that for any set $W\subseteq\sigma^-$ the following inequality holds:

$$\left|S\left(W\right)\right|/\left|\sigma^+\right|\ge\left|W\right|/\left|\sigma^-\right| \quad (3.6)$$

Proof. Indeed, the in-degree of every vertex of the set σ^+ is equal to $\left|\sigma^-\right|$, and the out-degree of every vertex of the set σ is equal to $\left|\sigma^+\right|$. Therefore, inequality (3.5) can be re-written as

$$\left|\sigma^-\right|\left|S\left(W\right)\right|\ge\left|W\right|\left|\sigma^+\right|$$

From there follows (3.6). □

4. Cartesian Synonymous Partitions

Let us consider the case of Cartesian synonymous partitions and words of length two. Two different partitions of the basic alphabet $\mathcal{A}_s$ produce two quotient alphabets $\Omega_1=\left\{a_1,\cdots,a_r\right\}$ and $\Omega_2=\left\{b_1,\cdots,b_h\right\}$. We have described in detail the corresponding constructions of words and sequences in this case in Example 2.2.2 Let us examine the problem of construction of a dense sequence that includes all possible words of length two over the given alphabets. Let us use for this purpose the bipartite graph G with the sets of vertices σ^- and σ^+ introduced in Section 3; $\left|\sigma^-\right|=r, \left|\sigma^+\right|=h$ The set σ^- corresponds to the alphabet Ω_1 and contains the first letters of the words, and the set σ^+ corresponds to the alphabet Ω_2 and contains the second letters of the words. All possible arcs of words are drawn in this graph. The arcs of recovery are determined by the table of correspondence of symbols, which in this case is not a free parameter. Specifically, we assume that the arc of recovery occurs from vertex b_j into vertex a_i if the intersection of the corresponding sets is not empty: $a_i\cap b_j\neq\varnothing$. This definition is reasonable, since the sequences in the terms of a Cartesian synonymous partition must correspond to the sequences of letters of the basic alphabet in the sense of Section 2. Thus, the possibility of construction of a dense sequence is determined by the conditions given in Section 3. Some of the Ω_1 and Ω_2 partitions of the basic alphabet satisfy these conditions, and some of them do not.

Example 4.1. Let the alphabet Ω_1 consist only of two sets of elements $\left\{a_1,\ldots,a_{s-1}\right\}$, $\left\{a_s\right\}$, and let the alphabet Ω_2 coincide with $\mathcal{A}_s\left(\Omega_2\equiv\mathcal{A}_s\right)$. In this case $\left|\sigma^-\right|=2$ and the out-degree of each vertex of the set $\left|\sigma^-\right|$ is equal to $\boldsymbol{s}$; also $\left|\sigma^+\right|=s$ and the in-degree of each vertex of the set $\left|\sigma^+\right|$ is equal to 2. If $W=\left\{a_s\right\}\in\sigma^-$, then $\left|S\left(W\right)\right|=1$, since only one arc of recovery enters the vertex $\left\{a_s\right\}$ according to the definition of the arcs of recovery: only vertex $\left\{a_s\right\}\in\sigma^+$ has a non-empty intersection with the vertex $\{a_s\} \in \sigma^-$. Thus, inequality $2|S(W)| \geq s|W|$ is true only when $s = 1,2$. According to Corollary 3.1 this indicates an absence of an Euler cycle in the graph with $s > 2$. Consider now the same construction with the transposition of the alphabet: $\Omega_1 \equiv \mathcal{A}_s$ and Ω_2 consists only of two sets of elements $\{a_1,...,a_{s-1}\},\{a_s\}$. In this case $|\sigma^-| = s$ and the out-degree of each vertex of the set $|\sigma^-|$ is equal to 2; also $|\sigma^+| = 2$ and the in-degree of each vertex of the set $|\sigma^+|$ is equal to s. Let

$W = \{a_1, ..., a_{s-1}\} \in \sigma^-, |W| = s-1$. All arcs of recovery, by construction, are drawn from the sole vertex

$\{a_1,\cdots,a_{s-1}\}$ of the set $|\sigma^+|$ that has a non-empty intersection with each of the vertices $a_i \in \sigma^-, i = 1,\cdots,s-1$. Therefore, $|S(W)| = 1$. According to Corollary 3.1 an Euler cycle exists if $|\sigma^- |S(W)|^3 |W||\sigma^+|$. In our case this inequality yields $s \geq 2(s-1)$ and it is only fulfilled with $s = 1,2$.

Let us suppose that positive integers υ_1 and υ_2 are such that s/υ_1 and s/υ_2 are integers.

Statement 4.1. Let the set $\Omega_1 = \{a_1,\cdots,a_r\}$ and the set $\Omega_2 = \{b_1,\cdots,b_h\}$ be the partitions of the basic alphabet $\mathcal{A}_s$, where the set Ω_1 consists of subsets of size υ_1 and the set Ω_2 consists of subsets of size υ_2. There exists a cyclic dense sequence in the set of all possible words over these alphabets.

Proof. Let us examine the corresponding bipartite graph G. The set of vertices σ^- corresponds to the quotient letters of the set Ω_1 and the set of vertices σ^+ corresponds to the quotient letters of the set Ω_2. $|\sigma^-| = s/\upsilon_1$ and $|\sigma^+| = s/\upsilon_2$ by construction. Consider an arbitrary set $W \subseteq \sigma^-$. Then $S(W) \subseteq \sigma^+$, by definition, is a set of all vertices from which the arcs of recovery are drawn into the vertices of the set $W \subseteq \sigma^-$. Further, the number of different letters of the basic alphabet $\mathcal{A}_s$ in all words of the set W is equal to $\upsilon_1|W|$; the same letters, by the definition of the arcs of recovery, can be found in all words of the set $S(W)$. Thus, $|S(W)| \geq \upsilon_1|W|/\upsilon_2$. Therefore the following inequality holds:

$$|S(W)|s/\upsilon_1 \geq (\upsilon_1|W|/\upsilon_2)\ s/\upsilon_1 = |W|s/\upsilon_2$$

Since an Eulerian conditions, according to Corollary 3.1, take in this case the form $|S(W)|s/\upsilon_1 \geq |W|s/\upsilon_2$, the Statement is proven. □

Above we analyzed the case when the quotient letters of each alphabet consisted of the same number of letters of the basic alphabet. Now let us consider the case when the number of letters of the basic alphabet in the quotient letters is not fixed. For two partitions $\Omega_1 = \{a_1,\cdots,a_r\}$ and $\Omega_2 = \{b_1,\cdots,b_h\}$ of the basic alphabet $\mathcal{A}_s$ with an equal number of subsets $(r = h)$, the in-degree and out-degree of each vertex are equal. According to the Theorem 3.1, in this case the existence of an Euler cycle depends only on Cartesian partitions that produce alphabets Ω_1 and Ω_2, since the table of correspondence is determined by the composition of the sets. It is possible to reformulate the criterion of the existence of the dense cycle obtained in Theorem 3.1. Let us examine the set of words $W \subset \Omega_1$. Partition Ω_2 in an obvious manner induces the partition of the set of all the letters over the basic alphabet included in the words from the set W. Let us denote this partition by

$$W|\Omega_2 = \{b_1 \cap W, b_2 \cap W, \cdots, b_h \cap W\}$$

and let us denote the number of non-empty subsets of $W|\Omega_2$ by $|W|\Omega_2|$. By definition, the arc of recovery occurs from vertex b_i into vertex of the set W if $b_i \cap W \neq \varnothing$. Therefore, the following inequality always holds: $|S(W)| \geq |W|\Omega_2|$.

Statement 4.2. *Let the sets*

$$\Omega_1 = \{a_1,\cdots,a_r\},\ \Omega_2 = \{b_1,\cdots,b_h\}$$

be the partitions of the basic alphabet $\mathcal{A}_s$ into an equal quantity of subsets $(r = h)$). Let for each subset $W \subset \Omega_1$ the inequality $|W|\Omega_2| \geq |W|$ be fulfilled. Then there is $W \subset \Omega_1$ *a dense cyclic sequence in the set of all possible words over these alphabets.*

The proof is obvious in a view of the observations made above. Specifically, from the inequalities $|S(W)| \geq |W|\Omega_2|$ and $|W|\Omega_2| \geq |W|$ we obtain the following inequality: $|S(W)| \geq |W|$, and apply Theorem 3.1.

5. Conclusions

Unlike the traditional setting up a problem in the coding theory [12], coding in DNA is not aimed at the correction of possible mistakes made during transferring information. In the case of DNA the crucial point is the way of recording information. We have chosen a particular important case of overlapping messages together with the degeneracy of the code. This scheme is interesting also regardless of how common this phenomenon is in biological objects.

We suggest a simple model of such coding on the basis of the standard triplet code. The quality of the code was tested on a sequence, which contained one word from each synonymous group. A full analysis of the model is carried out and the criterion for the possibility of building the test sequence is obtained in relation to different Cartesian partitions.

The methods developed in this article may be applied to building sequences of more general types, where, for each synonymous group, the number of the words that are contained in the sequence is preset. The same methods are applicable for analysis in the case of the standard triplet code [8]. Noteworthy also is the concept of "abstract alphabet" and the sequences built over it. This alphabet allows for superposing not only the same letters, but also different ones, as determined by the table of correspondence.

5. Acknowledgements

The authors are grateful to Professors Ed. Trifonov, A. Vainshtein, A. Korol for fruitful discussions.

REFERENCES

[1] P. J. Cock and D. E. Whitworth, "Evolution of Gene Overlaps: Relative Reading Frame Bias in Prokaryotic Two-Component System Genes," *Journal of Molecular Evolution*, Vol. 64, No. 4, 2007, pp. 475-462. doi:10.1007/s00239-006-0180-1

[2] Z. I. Johnson and S. W. Chisholm, "Properties of Overlapping Genes are Conserved across Microbial Genomes," *Genome Research*, Vol. 14, No. 11, 2004, pp. 2268-2272. doi:10.1101/gr.2433104

[3] C. Kingsford, A. Delcher and S. L. Salzberg, "A Unified Model Explaining the Offsets of Overlapping and Near-overlapping Prokaryotic Genes," *Molecular Biology and Evolution*, Vol. 24, No. 9, 2007, pp. 2091-2098. doi:10.1093/molbev/msm145

[4] I. Makalowska, C. F. Lin and W. Makalowski. "Overlapping Genes in Vertebrate Genomes," *Computational Biology and Chemistry*, Vol. 29, No. 1, 2005, pp. 1-12. doi:10.1016/j.compbiolchem.2004.12.006

[5] D. Candotti, C. Chappey, M. Rosenheim, P. M'Pelé, J. M. Huraux and H. Agut, "High Variability of the Gag/Pol Transframe Region among HIV-1 Isolates," *Comptes Rendus de l'Académie des Sciences*: *Série III*, Vol. 317, No. 2, 1994, pp. 183-923.

[6] K. M. McGirr and G. C. Buehuring, "Tax & Rex: Overlapping Genes of the Deltaretrovirus Group," *Virus Genes*, Vol. 32, No. 3, 2006, pp. 229-239. doi:10.1007/s11262-005-6907-z

[7] H. L. Zaaijer, F. J. van Hemert, M. N. Koppelman and V. V. Lukashov, "Independent Evolution of Overlapping Polymerase and Surface Protein Genes of Hepatitis B Virus," *Journal of General Virology*, Vol. 88, 2007, pp. 2137-2143. doi:10.1099/vir.0.82906-0

[8] M. Gorel and V. M. Kirzhner, "Degenerate Coding and Sequence Compacting," ESI Preprints 1819, 2006. http://www.esi.ac.at/preprints/ESI-Preprints.html

[9] M. Lothaire, "Applied Combinatorics on Words," Encyclopedia of Mathematics and Its Applications, Cambridge University Press, Cambridge, 2005

[10] M. Hall, "Combinatorial Theory," John Wiley & Sons, Hoboken, 1976.

[11] L. R. Ford, Jr. and D. R. Fulkerson, "Flows in Networks," Princeton University Press, Princeton, 1962.

[12] I. F. Blake, "A Perspective on Coding Theory," *Information Sciences*, Vol. 57-58, 1991, pp. 111-118. doi:10.1016/0020-0255(91)90070-B