Journal of Harbin Engineering University Vol 44 No. 9
ISSN: 1006-7043 September 2023

DNA Languages Favourable for Computation: Involutive Factor-Free Languages

C Annal Deva Priya Darshini
Department of Mathematics,
Madras Christian College, University of Madras, Chennai.
annaldevapriyadarshini@gmail.com,

Abstract

Formal languages play a central role in many branches of study, including DNA computing. A DNA strand, the basic unit of every living cell, is a sequence of nucleotides Adenine, Guanine, Cytosine, Thymine denoted by the symbols A, G, C, and T with the nucleotides A bindingto T and C binding to G. A DNA strand can thus be viewed as a word over the four-letter DNA alphabet {A, T, C, G} and the relation between the nucleotides can be expressed in terms of an involution mapping, also known as the Watson Crick involution[1]. Generalizing the notion ofWatson-Crick complementary which is central in DNA computing, involutively bordered wordsand involutively unbordered words were introduced and investigated[3]. Also, the combinatorial properties of words have been investigated over DNA computing. Let θ be a morphic or antimorphic involution of *.

A word w ∈ * is a θ- factor free or involutive factor free if none of its factors is a θ- factorof the word w[2]. In this paper, we propose to study the involutive factor-free languages over the Watson-Crick alphabet and their properties.

Keywords: Unbordered words, θ- factor words, θ- factor free words.

1 Introduction

A finite word is a finite sequence of symbols over an alphabet. Words play a central rolein many branches of study like Formal Languages and Automata Theory, computability and DNA computing[1,4]. A DNA strand which is the basic unit of every living cell is a sequence of nucleotides, adenine, guanine, cytosine, and thymine denoted by the symbols "a, t, c, g"; with these nucleotides binding to each other in a specific manner. DNA computation involves encoding problems in DNA strands, performing bio-operations for computation, and decoding solutions. Watson-Crick pairing and annealing ensure complementary strands twist into the double helix. DNA polymerase copies data, ligases bind molecules, and gel electrophoresis separates DNA by length. Adleman showcased DNA's parallel computation capability in 1994, solving a computationally 'hard' problem using molecular biology techniques. Conventional electronic computers solve problems involving large calculations but they become extremely slow when asked to sample billions of possible answers. But Adleman was able to demonstrate that molecules can solve the same problem quickly by simultaneously carrying out billions of operations in parallel. Adleman chose the Hamiltonian Path Problem for his experiment. The problem is to find a path through a directed graph that starts and ends at defined nodes and visits each number exactly once. Adleman identified a 7 nodes Hamiltonian path using some of the basic tools of molecular biology. The nodes were each encoded by a 20-base DNA nucleotide and routes between them by a complementary DNA nucleotide strand related to both the adjacent nodes. When all the nodes and route nucleotide strands were mixed together they annealed forming double–stranded DNA of varying lengths. They were ligated together to find the required solution. The Hamiltonian Path was verified in several ways. The author L. Kari[5] has classified the notions of bordered and unbordered words over the DNAalphabet $\Sigma = \{a, t, c, g\}$ by modelling the relation between the nucleotides as an anti-morphic involution

function and introduced the notion of bordered and unbordered words. A non-empty word over an alphabet is called bordered if it has a proper prefix which is also a suffix of a word. A non-empty word is called unbordered if it is not bordered. Also θ – bordered words were investigated, where θ is a morphic or antimorphic involution.The author has proved that the set of all θ – bordered words is regular, when θ is an antimorphic involution and the set of all θ – bordered words is context-sensitive when θ is morphic involution. The properties of θ – bordered and θ – unbordered words and relations between θ -bordered and θ – unbordered words was studied. In [3], the properties of language which ensure that the words of such language will not form undesirable bonds when used in DNA computations were introduced. Also, several characterizations of the desired properties and the methods for obtaining languages with such properties was investigated. In [6], the authors have introduced and investigated the idea of θ – bordered factors, the concept of θ – valence of a bordered factor and – sub-word complexity of a word. Some properties of involutively θ – bordered factors of a word were also investigated. In [2], the author has introduced and investigated the involutive factor of a word and obtained several combinatorial properties of such words under a morphic or antimorphic involution. Properties of the complementary notion of involutive factor-free words are also obtained.

In this paper, motivated by [2], we extend the involutive factor-free words to the involutive factor free language and study their properties. This helps to form a sequence of DNA strands with no twists in it making it feasible for DNA computing.

This paper is organized as follows: section 2 reviews the basic concepts, definitions and properties of involutively factor-free words. Section 3 deals with the involutive factor-free language. Section 4 gives the concluding remarks.

2 Basic Concepts

We first recall certain basic notions. An alphabet is a finite set of symbols Σ. A word w over Σ is a finite sequence of symbols of Σ. For example, *aabab* is a word over Σ = {*a, b*}. We denote by Σ^*, the set of all words over Σ, including the empty word λ and by $\Sigma^+$the set of all non-empty words over Σ. A language L over Σ is a subset of Σ^*. For a word $w \in \Sigma^+$, $alph(w)$ is a set of symbols of Σ. For example, if Σ = {*a, t, c, g*}, w = *attat*, then $alph(w)$ = {*a, t*}. For a word, $w \in \Sigma^*$, the length of is the number of non-empty symbols in w andit is denoted by $|w|$. For a word $w \in \Sigma^+$, the word $u \in \Sigma^+$ is a factor of w if there are words $\alpha, \beta \in \Sigma^*$ such that $w = \alpha$ u β. If $\alpha = \lambda$ then u is called the prefix of w and if $\beta = \lambda$ then u is called a suffix of w. The set of all factors of w is denoted by $F(w)$.

A mapping θ: $\Sigma^* \rightarrow \Sigma^*$satisfying the property that $\theta(xy) = \theta(x)\theta(y)$ is a morphism on Σ^*If $\theta(xy) = \theta(y)\theta(x)$, then θ is an antimorphism on Σ^*. The mapping θ is an involution on Σ^* if $\theta(\theta(x)) = x$,for all $x \in \Sigma^*$. The mapping $\theta: \Sigma^* \rightarrow \Sigma^*$defined on the DNA alphabet is given by $\theta(a) = t$, $\theta(t) = a$, $\theta(c) = g$, $\theta(g) = c$. θ is an involution as the complement of a sequence equals the sequence itself.

We now recall the notions of bordered and unbordered words. Also, the notions of involutive factors of a word and the involutive factor- free words.

Definition 1. *Let θ be a morphic or an antimorphic involution on Σ^*. A word $u \in \Sigma^+$ is said to be θ – bordered if there exists a word $v \in \Sigma^+$such that $u = vx = y\theta(v)$ for $x, y \in \Sigma^+$. Then v is a θ – border of u. Otherwise, the word u is called θ – unbordered.*

Note:

$D_\theta(1)$ is the set of all unbordered words.

Lemma 2. *Let θ be an antimorphic*

involution. Then $x \in \Sigma^+$ *is* θ *- bordered if and only if*

$x = ay\theta(a)$ *for some* $a \in \Sigma$ *and* $y \in \Sigma^*$.

Proposition 3. *When* θ *is an antimorphic involution on* Σ^*, $D_\theta(1)$ *is a regular language.*

Definition 4. *Let* θ *be a morphic or an antimorphic involution on* Σ^*. *A word* $u \in \Sigma^+$ *is said to be* θ *– factor or an involutive factor of* w *if* u *and* $\theta(u) \in F(w)$. *.i.e. both* u *and* $\theta(u)$ *are the factors of the word.*

Example 5. *Let* $\Sigma = \{a, t, c, g\}$ *be an alphabet. Let* θ *be a mapping on* Σ^* *defined by*
$\theta(a) = t, \theta(t) = a, \theta(c) = g, \theta(g) = c.$

(i) *If* θ *is a morphic involution then the factor* v = *gact of the word* w = ***a gact*** *attg* ***ctga*** *t is a* θ *–factor since* $\theta(v)$ = *ctga is also a factor of w.*
(ii) *If* θ *is an antimorphic involution then the factor* v = *aagt of the word* w = *tga* ***ttca*** *g* ***aagt*** *is a*

θ *– factor since* $\theta(v) = \theta(aagt)$ = *ttca is also a factor of w.*

Definition 6. *Let* θ *be a morphic or an antimorphic involution on* Σ^*. *A word* $w \in \Sigma^*$*is*
θ *–factor free or involutive factor free if none of its factors is a* θ *– factor of w. i.e. the word w willnot contain its* θ *- image.*

Example 7. *Consider a word w = tgggttg over* $\Sigma = \{a, t, c, g\}$. *Let* θ *be a morphic involution on*

defined by $\theta(a) = t, \theta(t) = a, \theta(c) = g, \theta(g) = c.$ *No factor of u of w is a* θ *– factor of w as* $\theta(u)$

is not a factor of w.

Definition 8. *Let* L_1 *and* L_2 *be languages in the same alphabet. Then the right quotient of* L_1*is defined as* $L_1/L_2 = \{x : xy \in L_1$ *, for some* $y \in L_2\}$ *and the left quotient of* L_1 *is defined as* $L_1 \setminus L_2 = \{x: xy \in L_2$ *for some* $y \in L_1\}$.

For properties and other results on involutively factor free words, we refer [3].

3 Languages Avoiding Involutively Bordered Factor

We are interested in studying languages over the DNA alphabet with "nice" coding properties. The main challenge of working with the DNA alphabet is the presence of the Watson – Crick complementarity relation. This property should be avoided during computations so that a strand does not "stick" to itself creating regions that are not available for further computation.

In [3], Priya Darshini and Rajkumar Dare, have introduced θ - factor free words and studiedthe properties. Motivated by this we define $F = \cup_{a\in\Sigma} \Sigma^+ a\Sigma^* \theta(a)\Sigma^+$, then F is the language of all words with θ - factors. The

language $\Sigma^* \setminus F$ is the complement of F containing all θ - factor free words. For a θ – factor freeword $w = \Sigma^+\theta(a)\,\Sigma^+$ if Σ^+ is replaced by Σ^*then w is a involutively bordered word.

Proposition 9. *The language F is a regular language.*

Proof. The conclusion follows from the fact that the expression $\Sigma^+ a\Sigma^*\theta(a)\Sigma^+$is a regularexpression. Hence F is regular.

Corollary 1. *The language* $\Sigma^* \setminus F$ *is a regular language. That is set of all* θ *– factor free word isregular.*

Consider, $L \subseteq \Sigma^2$ as $L = \{aa, tt, gg, cc, ac, ag, tc, tg\}$. L is the set of θ – factor free words of length 2(smallest length). The words $w_1 = aa$ and $w_2 = tt$ are θ – complements and hence concatenation of the type w_1w_2 or w_2w_1 or $w_1w_2^*$ will result in involutively bordered words or words with θ – factor. such a language L may not be useful in bio-

computation. L contains θ – factor free words. Our goal is to form languages that do not contain such words. In fact, we would like to study languages that would contain involutively factor free words alone even after the concatenation of words in the language.

We give a condition under which the concatenation of the words will always be θ – factor free. Recall in[2] that the concatenation of two θ – factor free words is also θ – factor free by "Let θ be a morphic or an antimorphic involution on Σ^*. Let $u, v \in \Sigma^+$ be θ – factor free. Then$uv \in \Sigma^+$ is θ – factor free if and only if $\theta(alph(u)) \cup alph(v) = \varphi$. Also, let θ be a morphic or an antimorphic involution on Σ^*. Let $u, v \in \Sigma^+$ be θ – factor free with $\theta(alph(u)) \cup alph(v) = \varphi$. Then u^+, v^+, $(uv)^+$ are θ – factor free.

In the next proposition, we give one more condition under which the concatenation of words will be θ - factor free.

Proposition 10. *For any $w_i, w_j, i \neq j, w_i, w_j \in \Sigma^+$. Let w_i, w_j be θ – factor free words. Then*

$w_i w_j$ is θ – factor free if and only if

(i) $\forall v \in F(w_i), \theta(v) \notin F(w_j)$

(ii) $\forall u \in F(w_j), \theta(u) \notin F(w_i)$

We are interested in involutive factor-free languages as $L \subseteq \Sigma^*$ such that

$L = \{w / \forall v \in F(w)$; (i) $\theta(v) \notin F(w)$

$ii)$ $\theta(v) \notin F(x)$, for any other $x \in L\}$.

For example, the language $L_1 \subseteq \Sigma^+$, $L_1 = \{a^n\} \cup \{(ac)^n\}$ will be an involutive factor-free languagebut $L_2 = \{a^n\} \cup \{t^n\}$ will not be an involutive factor-free language. Thus, a set of involutive factor-free words is not an involutive factor-free language. A simple extension of the notion of words to languages is not feasible in the context of DNA computing.

To this end, we partition the alphabet $= \{a, t, c, g\}$ into four alphabets, $\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4$ such that $\forall a \in \Gamma_i, \theta(a) \notin \Gamma_i$ for each i. Let $\Gamma_1 = \{a, c\}, \Gamma_2 = \{a, g\}, \Gamma_3 = \{t, c\}$ and $\Gamma_4 = \{t, g\}$. Then any $L \subseteq \Gamma_i^*$ will have words which are θ – factor -free and concatenation of any two words in L will not result in involutively bordered words or a word with θ – factors. The alphabet Γ_i is formed by putting together an alphabet of and another alphabet which is not its θ - image. This partition ensures the 'θ factor freeness' of the words in the alphabet.

Any θ – factors free word $w \in \Sigma^*$ is in one and only one alphabet Γ_i.

We infer some immediate consequences of this partition.

Lemma 11. *The following relations are true between the alphabets:*

(i) $\Gamma_1 = \theta(\Gamma_4), \Gamma_4 = \theta(\Gamma_1), \Gamma_2 = \theta(\Gamma_3)$ *and* $\Gamma_3 = \theta(\Gamma_2)$

(ii) $\Gamma_i \cup \Gamma_j = \Sigma$ *if* $\Gamma_i = \theta(\Gamma_i)$; $i \neq j$

$\Gamma_i \cap \Gamma_j = \varphi$ *if* $\Gamma_i = \theta(\Gamma_i)$; $i \neq j$

(iii) $\Gamma_i \cup \Gamma_j \cup \Gamma_k = \Sigma$ *if* $\Gamma_i = \theta(\Gamma_i)$; $i \neq j \neq k$

$\Gamma_i \cap \Gamma_j \cap \Gamma_k = \varphi$ *if* $\Gamma_i = \theta(\Gamma_i)$; $i \neq j \neq k$

Note that the involtive mapping θ - decides the partition of the alphabet.

For example, if $\Sigma = \{a, b, c\}$ and $\theta : a \to b$, $b \to a$,c $\to c$ then the partition $\Gamma_1 = \{a, c\}, \Gamma_2 = \{b\}$ will not be conducive for DNA computing. We require, $\Gamma_1 = \{a\}, \Gamma_2 = \{b\}, \Gamma_3 = \{c\}$for our purpose. Our study focuses on languages within this frameworks of DNA computing.

Definition 12. *The language avoiding involutively bordered words as factor is $L = \{w / w \in \Gamma_i^*\}$, for given i.*

We give some properties of such languages.

Proposition 13. *If $L \subseteq \Sigma^*$ is a language avoiding involutively bordered words i.e., $L \subseteq \Gamma_i^*$ forgiven i, then $\theta(L), L^c, L^*$, $L^+, L^c \subseteq \Gamma_i^*$ are also languages avoiding involutivley bordered words.*

Proof. Follows from the definition of L.

Proposition 14. *Let* L *be a family of languages in* Γ_i^* *for a given* i. *The family* L *is closed under union, intersection, concatenation, right and left quotient.*

Proof. Follows from the definition of Γ_i^*

Proposition 15. *For any* $L_1, L_2 \subseteq \Sigma^*$

(a) The union, intersection and the complement over union and intersection are languagesavoiding involutively bordered words as factors if and only if $L_1, L_2 \in \Gamma_i^*$ *for given* i.

(b) If $L_1 \in \Gamma_i$, $L_2 \in \theta(\Gamma_i)$ *for* $i \neq j$, *then*

i) $L_1 \cap L_2 = \varphi$

ii) $L_1 \cup L_2$ *will be a language avoiding involutively bordered word.*

Proposition 16. *For any three languages* $L_i, L_j, L_k \subset \Sigma^+, (i \neq j \neq k)$

(i) $L_i \cup (L_j \cap L_k)$ *is a language containing* θ *- factor free words if and only if* $L_i, L_j \in \Gamma_p$ *and* $L_k \in \Gamma_q$, $p \neq q$

. *(ii)* $L_i \cap (L_j \cup L_k)$ *is a language containing* θ *- factor free words if and only if* $L_i, L_j \in \Gamma_p$ *and* $L_k \in \Gamma_q$, $p \neq q$

. *(iii)* $L_i \cap (L_j \cup L_k)$ *is a language containing* θ *- factor free words if and only if* $L_j, L_k \in \Gamma_p$ *and* $L_i \in \Gamma_q$, $q \neq p$

. *(iv)* $L_i \cup (L_j \cap L_k)$ *is a language containing* θ *- factor free words if and only if* $L_i \in \Gamma_p$, $L_j \in \Gamma_q$ *and* $L_k \in \Gamma_\theta(q)$

.

(v) $L_i \cup (L_j \cap L_k)$ *is a language containing* θ *- factor words if and only if* $L_j \cap L_k \in \Gamma_p$ *and* $L_i \in \Gamma_q$, $q \neq p$.

.

Proof. *(i)* Let $L_j \subset \Gamma_p{}^+$ and $L_k \subseteq \Gamma_q{}^+$; $p \neq q$, then $(L_j \cap L_k) \subseteq \{a^+/a \in \Sigma\}$ where $\Gamma_p \cap \Gamma_q = \{a\}$.

For $L_i \subset \Gamma_p{}^+$, we have $L_i \cup (L_j \cap L_k) = \{w/w \in \Gamma^+\}$, since Γ^+ is a language of θ - factor free words.

(ii) Let $L_j \subset \Gamma^+$ and $L_k \subset \Gamma^+$; $p \neq q$, then $L_j \cap L_k$ contains words with θ - factors. The language

$L_i \cap (L_j \cup L_k)$ is however a language of θ - factor free words for $L_i \subset \Gamma^+$.

However, for $L_j, L_k \subseteq \Gamma^+$, $L_j \cup L_k \subseteq \Gamma^+$, then for any other language $L_i \subset \Gamma^+$; $L_i \cap (L_j \cup L_k)$

is however a language of θ - factor free words for $L_i \subset \Gamma^+$.

(iii) Let $L_j \in \Gamma_q$ and $L_k \in \theta(\Gamma_q)$, then $L_j \cap L_k$ will be empty. For any $L_i \in \Gamma_q$, $L_i \cup (L_j \cap L_k)$ willcontain words with θ - factor free.

(iv) Let $L_j, L_k \subseteq \Gamma^+$, then we have $L_j, L_k \cap \Gamma^+$. For any other language, $L_i \subseteq \Gamma^+$, $q \neq p$, the language $L_i \cup (L_j \cap L_k) \subseteq \Sigma^+$. Hence $L_i \cup (L_j \cap L_k)$ will always contains words with θ - factor.

Definition 17. *A language L is* θ *- factor closed if* θ *- factors of words in L are also in L. We have* $F(L) = \cup_{w \in L} F(w)$ *and* $F_\theta(L) = \cup_{w \in L} F_\theta(w)$. *Hence L is* θ *- factor closed if* $F_\theta(L) = F(L)$.

Example 18. *Let* = {a, c, t, g} *be an alphabet. Let* θ *be an antimorphic involution defined by*

$\theta(a) = t, \theta(t) = a, \theta(c) = g, \theta(g) = c$. *Let* $L = \{a^n t/n \geq 1\}$. *Then we get* $F(L) = \{a^n t/n \geq 1\} \cup \{a^n/n \geq 1\}$ *and* $F_\theta(L) = \{a^n t/n \geq 1\}$, *then* $L = F_\theta(L)$. *L is* θ *– factor closed.*

Proposition 19. *The language L is* θ *– factor closed if* $F_\theta(L) = F(L)$. *Let* $L^J = F(L) - F_\theta(L)$ *is always a language avoiding involutively bordered words as* θ *- factors. For a non-empty languageL* $\subseteq \Gamma_i^*$ *is a language avoiding words with* θ *- factors if and only if* $L^J = \varphi$.

4 Conclusion

Thus, we have introduced a method of generating DNA language avoiding involutively bor- dered factors and studied some of its properties especially conditions under which concatenation also leads to languages avoiding involutively bordered factors. From DNA computing is evidentthat molecular computers based on DNA have many appealing properties. They provide extremely dense information storage. For instance, considering a gram of DNA, when dried would consume a volume of approximately one cubic centimeter and can store data as such of one trillion CDs. They correspond in enormous parallel processing.

The originality of the contribution in this paper lies in introducing a general notion of involutive factor free or θ– factor-free languages and studying several language theoretic properties of involutive factor-free languages. We have explored the possibilities of treating the DNA alphabet as the binary alphabets by partitioning into four subsets. This would ensure that languages have the requisite encoding properties. We propose to extend this work by studying the closure properties of languages over Γ_i under the bio – operations like cutting, splicing, annealing, pasting, shuffling, deleting, inserting. This paper shows results related to concatenation of words alone. Generalization to other bio – operations is an interesting problem. The enormous potential of DNA computing is still in its nascent stage.

References

[1] Adleman L.M., “Computing with DNA”, Scientific American, pp. 34-41.

[2] Annal Deva PriyaDarshini C, Rajkumar Dare V, Ibrahim Venkat and Subramaniam K G, (2013-2014), “Factors of Words under an Involution”, Journal of Mathematics and Infor- matics, Vol-1, pp. 52-59.

[3] Hussini S, Kari L and Konstantinidis S, (2002), “Coding properties of DNA languages”, proceeding of DNA Based Computers, Vol- 7, pp. 57 - 69.

[4] Jonoska N, (2003), “Computing with b

[5] iomolecules: Trends and Challenges”, XVI Tarrang- ona Seminar on Formal Syntax and Semantics, Vol- 27.

[6] Kari L, Mahalingam K, (2007), “Involutively Bordered Words”, Intern. J. Found Comp. Sci, Vol-18, pp.1089-1106.

[7] Rajkumar Dare V, Annal Deva Priya Darshini C, (2011), “Bordered Factors of a Finite Word”, Proceedings of the 2011 Sixth International Conference on Bio-Inspired Computing: Theories and Applications, pp. 163 – 166.