Bond-Free Languages: Formalizations, Maximality and Construction Methods

Lila Kari[1], Stavros Konstantinidis[2,⋆], and Petr Sosík[1,3]

[1] Department of Computer Science, The University of Western Ontario, London, ON, Canada, N6A 5B7
{lila, sosik}@csd.uwo.ca

[2] Dept. of Mathematics and Computing Science, Saint Mary's University, Halifax, Nova Scotia, B3H 3C3 Canada
s.konstantinidis@stmarys.ca

[3] Institute of Computer Science, Silesian University, Opava, Czech Republic

Abstract. The problem of negative design of DNA languages is addressed, that is, properties and construction methods of large sets of words that prevent undesired bonds when used in DNA computations. We recall a few existing formalizations of the problem and then define the property of sim-bond-freedom, where sim is a similarity relation between words. We show that this property is decidable for context-free languages and polynomial-time decidable for regular languages. The maximality of this property also turns out to be decidable for regular languages and polynomial-time decidable for an important case of the Hamming similarity. Then we consider various construction methods for Hamming bond-free languages, including the recently introduced method of templates, and obtain a complete structural characterization of all maximal Hamming bond-free languages. This result is applicable to the θ-k-code property introduced by Jonoska and Mahalingam.

1 Introduction

Most of the operations involved in DNA computations rely on the capability of controlling the *bonds* that can be formed between *(single-stranded) DNA molecules*. Such bonds are created due to the well-known Watson-Crick *complementarity* property of the four nucleotides A, C, G, T, which are the building blocks of DNA molecules. This property is important in conjunction with the fact that every molecule has a certain *orientation*, which is denoted by placing the symbols '$5'-$' and '$-3'$' at the two ends of the sequence of nucleotides comprising the molecule. For example, the molecules $5'-ACCGT-3'$ and $3'-ACCGT-5'$ are different – they have different chemical properties. In practice, the collection of DNA molecules exists as a 'soup' inside a test *tube*. Under favorable physical

⋆ Corresponding author.

tube conditions, if a molecule of the form $5' - X_1X_2 \cdots X_k - 3'$, where each X_i is a nucleotide, encounters the molecule $5' - Y_k \cdots Y_2Y_1 - 3'$ in which each nucleotide Y_i is the complement of X_i, then the pairs (X_i, Y_i) will form k chemical bonds and a double-stranded structure will be created – see Figure 1(a).

```
5'- A G T T C C -3'    5'- v A G T T C C w -3'    5'- v A G T T C C w
    | | | | | |                | | | | | |                | | | | | |  x
3'- T C A A G G -5'    3'- z T C A A G G y -5'    3'- z T C A A G G y
        (a)                      (b)                        (c)
```

Fig. 1. Vertical bars represent bonds between complementary nucleotides. In (b), the complementary parts $5' - AGTTCC - 3'$ and $5' - GGAACT - 3'$ of the DNA molecules $5' - vAGTTCCw - 3'$ and $5' - yGGAACTz - 3'$ bind together. In (c), the molecule $5' - vAGTTCCwxyGGAACTz - 3'$ is twisted at x and its complementary parts bind together

It is important to note that bonds can be formed even between complementary *parts* of two molecules, provided that these parts are *sufficiently long* – see Figure 1(b). Moreover, a molecule containing two complementary parts can bind to itself, or to a copy of itself – see Figure 1(c).

The bonds shown in Figure 1 are formed between parts that are *perfect* complements of each other. In practice, however, it is possible that two parts of molecules will bind together even if some of their corresponding nucleotides are not complementary to each other – see Figure 2.

```
5'- v A G A T T C C G T G w -3'    5'- v A G A T T C C G T G w
      | | |   | | |   | |                | | |   | | |   | |  x
3'- z T C T C A G G A A C y -5'    3'- z T C T C A G G A A C y
              (a)                                (b)
```

Fig. 2. In (a), parts of two DNA molecules bind together although these parts are not perfect complements of each other. In (b), the same parts appear in one molecule

1.1 The Problem of Undesirable Bonds

The success of a DNA operation relies on the assumption that no accidental bonds can be formed between molecules in the tube before the operation is initiated, or even during the operation. With this motivation, one of the foremost problems in DNA computing today is the following.

Problem 1. Define a large, potential collection of DNA molecules such that there can be no (sufficiently long and possibly imperfect) complementary parts in any two molecules, and no (sufficiently long and possibly imperfect) complementary parts in any one molecule.

In many cases in the literature, this problem is addressed in conjunction with the *uniqueness* problem, which involves designing molecules whose parts are different between each other. The motivation here is that, usually, a DNA operation is intended only for molecules containing a specific pattern (or specific patterns) of nucleotides. In this paper, however, we focus on Problem 1.

1.2 Notation for Molecules and Bonds

We proceed now with establishing the notation that would allow us to describe formalizations of Problem 1. Specifically, we define the terms *word, subword, language, involution, and codeword.*

A given alphabet can be used to form sequences of symbols that are called *words.* For example, 01001 is a word over the alphabet $\{0,1\}$. The *length* of a word w is denoted by $|w|$. For example, $|01001| = 5$. The prime example of an alphabet will be the DNA alphabet $\{A,C,G,T\}$. In this case, we agree that the left end of a *DNA word* represents the $5'$−end of the corresponding DNA molecule. For example, the word $CCATGT$ represents the molecule $5' - CCATGT - 3'$. If a word w can be written in the form xyz – this is the catenation of some words x, y and z – then we say that y is a *subword* of w. A *language* is any set of words. We shall use the expression 'x is a subword of a language' as a shorthand for x is a subword of some word in the language. One use of a language L is to represent all the possible distinct copies of DNA molecules that might appear in a tube. In this case, we refer to L as a *tube language* and we assume that every word in L is of length at least k, for some *parameter* k. This parameter represents the smallest length of two molecule parts for which it is possible to form a stable bond.

To represent the complementarity of nucleotides we use the concept of antimorphic involution introduced in [13]. In general an *involution* of an alphabet Σ is a function $\theta : \Sigma \to \Sigma$ such that $\theta(\theta(a)) = a$, for all symbols a in Σ. The involution is called *antimorphic* if we extend it to words such that $\theta(a_1 \cdots a_n) = \theta(a_n) \cdots \theta(a_1)$, where each a_i is a symbol in Σ. The prime example of an antimorphic involution will be the *DNA involution* τ such that

$$\tau(A) = T,\ \tau(T) = A,\ \tau(C) = G,\ \tau(G) = C.$$

For example, $\tau(ACCGTT) = AACGGT$. In general, for two DNA words x and y of length k, *the identity $\tau(x) = y$ represents the fact that the molecules (or parts of molecules) $5' - x - 3'$ and $5' - y - 3'$ could bind to each other.* According to the requirement in Problem 1, if $k = 6$, the words $ACCGTT$ and $AACGGT$ should not be subwords of the tube language L.

In the literature on DNA encodings, the tube language L is usually equal to, or a subset of, K^+, where K is a finite language whose elements are called *codewords.* The language K^+ consists of all words that are obtained by concatenating one or more codewords from K. For a nonnegative integer n, the notation K^n is used for the set of all words that are obtained by concatenating any n codewords from K. In general, K might contain codewords of different lengths.

In many cases, however, the set K consists of words of a certain fixed length l. In this case, we shall refer to K as a *code of length l*.

1.3 Formalizations of the Problem of Undesirable Bonds

With the preceding terminology in mind, Problem 1 is called the negative word design problem in [18]. Now we recall a few existing formalizations of Problem 1 and we propose a new one, which appears to be closer to the intuition behind the problem. It should be noted, however, that all formalizations are inter-related in some interesting ways.

One of the most recent attempts to address Problem 1 appears in [10]. In that paper, the authors require that a tube language L must satisfy the following property.

P1$[k]$**:** If x and y are any subwords of L of length k then $x \neq \tau(y)$.

A language satisfying this property is called a τ-k-code in [10]. An advantage of this formalization is that the property is defined *independently* of the structure of L. This property is also considered implicitly in [3] and [6]. In particular, reference [3] considers tube languages of the form $(sZ)^+$ satisfying **P1**$[k]$, where s is a fixed word of length k and Z is a code of length k – the notation sZ represents the set of all words sz such that z is in Z.

In [9], the authors introduce the concept of a *strictly τ-free code K*, which is a generalization of the notion of comma-free code [12], and show that the language K^+ must be strictly τ-free as well. Here we shall assume that K is of fixed length k. In this formalization the tube language L is equal to K^+. Using the tools of [9], it can be shown that L is a strictly τ-free language *iff* (if and only if) L satisfies the following property

P2$[k]$**:** If x is a subword of L of length k and v is a codeword in K then $x \neq \tau(v)$.

We note that similar properties are considered also in [15] and [16].

As noted earlier, parts of DNA molecules can bind to each other even if they are not perfect complements of each other. Hence, although sufficient, the condition $\tau(x) = y$ might not be necessary for the DNA words x and y to stick together. The common approach to deal with this is to modify the above condition by using the Hamming distance function $H(\cdot,\cdot)$. More specifically, for two words x and y of length k, *the relation $H(x, \tau(y)) \leq d$ represents the fact that the molecules (or parts of molecules) $5' - x - 3'$ and $5' - y - 3'$ could bind to each other.* Here, d is a nonnegative integer less than k.

In [5] and [21], the authors consider codes K of length k satisfying the following property

P3$[d, k]$**:** If u and v are any codewords in K then $H(u, \tau(v)) > d$.

In fact the above property is studied in conjunction with the uniqueness property $H(K) > d$.

Reference [2] considers a measure between two words, which is applied to codes of length k whose words can be concatenated in arbitrary ways. Thus, the tube language here is $L = K^+$. The code K satisfies certain uniqueness conditions as well as conditions related to Problem 1. In particular, the tube language $L = K^+$ satisfies the following property.

P4$[d, k]$**:** If x is a subword of L of length k and v is a codeword in K then $H(x, \tau(v)) > d$.

We note that also reference [19] considers this property for tube languages of the form $K_1K_2 \cdots K_m$, where each K_i is a certain code of length k.

With '$H(x, \tau(y)) \leq d$' as the criterion for x and y to bind together, it appears that **P4**$[d, k]$ is the strictest property in the literature for addressing Problem 1. This property, however, is not sufficient in general for avoiding undesirable bonds in the tube. To see this, consider the case where

$$d = 1, \quad k = 5, \quad K = \{ACGAT, CCGAA\}.$$

One can verify that the language K^+ satisfies **P4**$[d, k]$ and that the DNA words $ACGATACGATCCGAA$ and $ACGATCCGAACCGAA$ are in K^+ and contain the subwords $GATCC$ and $CGATC$ such that

$$H(GATCC, \tau(CGATC)) \leq 1.$$

Motivated by the above observation, we introduce the following property of a tube language L.

P5$[d, k]$**:** If x and y are any subwords of L of length k then $H(x, \tau(y)) > d$.

Note that, as in the case of **P1**$[k]$, the new property is defined independently of the structure of L. Any tube language satisfying this property will be called a $(\tau, H_{d,k})$*-bond-free language.*

We list now a few interesting connections among the properties **P1**–**P5**. We note that the condition $x \neq \tau(y)$ is equivalent to $H(x, \tau(y)) > 0$.

P3 and P5: In this paper we introduce the *subword closure operation* $\otimes$ and we show that if a code K satisfies **P3**$[d, k]$ then the language $K^{\otimes}$ satisfies **P5**$[d, k]$.

P4 and P2: It is evident that any language K^+ satisfying **P4**$[d, k]$ also satisfies **P2**$[k]$. Moreover, **P4**$[0, k]$ is identical to **P2**$[k]$. We can show that, for every code Q of length q, if the language Q^+ satisfies **P2**$[q]$ then the language $(Q^{d+1})^+$ satisfies **P4**$[d, q(d + 1)]$, for any $d > 0$.

P4 and P5: It is evident that any language K^+ satisfying **P5**$[d, k]$ also satisfies **P4**$[d, k]$. Moreover, it can be shown that if K^+ satisfies **P4**$[d, k]$ then $(K^2)^+$ satisfies **P5**$[d, k]$.

P5 and P1: Obviously, any language satisfying **P5**$[d, k]$ also satisfies **P1**$[k]$. Moreover, the property **P1**$[k]$ coincides with **P5**$[0, k]$. It can be shown that every language satisfying **P1**$[q]$, for some positive integer q, also satisfies **P5**$[d, q(d+1)]$ for every $d > 0$, and conversely, if the language is of the form K^+ and satisfies **P5**$[d, k]$ then it satisfies **P1**$[k - d]$ as well.

Important Note. Proofs of the results obtained in this paper can be found in the full version [17].

1.4 A More General Formalization: $(\boldsymbol{\theta}, \mathbf{sim})$-Bond-Freeness

The choice of the Hamming distance in the condition '$H(x, \tau(y))$' for *similarity* between words is a very natural one and has attracted a lot of interest in the literature. One might argue, however, that parts of two DNA molecules could form a stable bond even if they have different lengths – hybridizations of this type are addressed in [1]. Based on this observation, the condition for two subwords x and y to bind together should be

$$|x|, |y| \geq k \quad \text{and} \quad Lev(x, \tau(y)) \leq d.$$

The symbol $|u|$ denotes the length of the word u and $Lev(u, v)$ is the *Levenshtein distance* between the words u and v – this is the smallest number of substitutions, insertions and deletions of symbols required to transform u to v. With this formulation, the condition for similarity based on the Hamming distance can be rephrased as follows

$$|x|, |y| \geq k \quad \text{and} \quad H(x, \tau(y)) \leq d,$$

where we assume that $H(u, v) = \infty$ if the words u and v have different lengths. In general, for any similarity relation $\text{sim}(\cdot, \cdot)$ between words and for every involution θ, we define the following property of a language L.

P$[\theta, \text{sim}]$**:** If x and y are any nonempty subwords of L then $\text{sim}(x, \theta(y))$ is false.

Any language satisfying **P**$[\theta, \text{sim}]$ is called a (θ, sim)-*bond-free language.*

The precise definition of a similarity relation is given in Section 2. It can be shown that the relations '$|u|, |v| \geq k$ and $H(u, v) \leq d$' and '$|u|, |v| \geq k$ and $Lev(u, v) \leq d$' are indeed similarity relations. *For these relations we shall use the notation* $H_{d,k}$ *and* $Lev_{d,k}$*, respectively.*

2 Decidability Questions About $(\boldsymbol{\theta}, \mathbf{sim})$-Bond-Freedom

In this section, we use the general tools about language operations and trajectories obtained in [16] and we show that one can decide in quadratic time whether a given regular language is (θ, sim)-bond-free. Moreover, we show that this problem is decidable even when the given language is context-free. Then, we use also the general tools about maximal solutions of language inequations developed in [14] and [16] to establish the decidability of whether a given regular language is maximal with respect to the (θ, sim)-bond-free property. The acronyms *DFA* and *NFA* stand for deterministic and nondeterministic, respectively, finite automaton. A relation between words (binary relation) is *rational* if it is realized by a finite-state transducer.

Definition 1. *A binary relation* sim *is called a* similarity relation *with parameters* (t, l)*, where* t *and* l *are nonnegative integers, if the following conditions are satisfied. (i) If* $\text{sim}(u, v)$ *is true then* $\text{abs}(|u| - |v|) \leq t$*. (ii) If* $\text{sim}(u, v)$ *is true and* $|u|, |v| > l$ *then there are proper subwords* x *and* y *of* u *and* v*, respectively, such that* $\text{sim}(x, y)$ *is true.*

We can interpret the above conditions as follows: (i) the lengths of two similar words cannot be too different and (ii) if two words are similar and long enough, then they contain two similar proper subwords. *In the rest of the section we shall assume that* sim *is a fixed, but arbitrary, similarity relation with parameters* (t, l). It is evident that the relation $H_{d,k}$ defined in Subsection 1.4 is an example of a similarity relation with parameters $(0, k)$. It can be shown that also $Lev_{d,k}$ is a similarity relation, with parameters $(d, d + k)$ – see [17].

Theorem 1. *Assume that* sim *is a rational relation. The following problem is decidable in quadratic time.* ***Input:*** *NFA A.* ***Output:*** *Y/N, depending on whether* $L(A)$ *is a* (θ, sim)*-bond-free language.*

We note that for most of the DNA language properties considered in [9, 15, 16] the above problem is undecidable for context-free languages. As the (θ, sim)-bond-free property seems to be rather general, it might be surprising that the same problem is decidable.

Theorem 2. *If the similarity relation* sim *is computable, then it is decidable whether a given context-free language is (*θ, sim*)-bond-free.*

Corollary 1. *Let* d *and* k *be nonnegative integers with* $k \geq 1$. *It is decidable whether a given context-free language is* $(\theta, H_{d,k})$*-bond-free (or* $(\theta, Lev_{d,k})$*-bond-free).*

Theorem 3. *Assume that the similarity relation* sim *is rational. Then the following problem is decidable.* ***Input:*** *NFAs A and B such that* $L(A)$ *is a* (θ, sim)*-bond-free subset of* $L(B)$. ***Output:*** *Y/N, depending on whether* $L(A)$ *is a maximal* (θ, sim)*-bond-free subset of* $L(B)$.

3 Decidability of Maximality in the Hamming Case

In the literature on DNA encodings, and in coding theory in general, the set of words that are involved in the application of interest are usually formed by concatenating shorter words of a certain fixed length. Following this practice, we consider languages that are subsets of $(\Sigma^k)^+$, for some positive integer k. We call such languages *k-block languages*. Naturally, any regular k-block language can be represented by a special type of lazy DFA, which we call *k-block DFA*. This is a deterministic finite automaton such that, for every production $pu \to q$, the word u is of length k.

The decision method for maximality of the previous section is not of polynomial time. In this section, however, we are able to show a polynomial time algorithm for testing whether a given regular langauge is $(\theta, H_{d,k})$-bond-free, for $d = 0$ or $d = 1$. We remind the reader that, in the case of $d = 0$, the property coincides with **P1**$[k]$ – see Subsection 1.3. Next we illustrate the concept of maximality with an example. *The notation* $\mathrm{Sub}_k(L)$ *represents the set of all subwords of length* k *of the language* L.

Example 3.1. Consider the code $K_1 = \{AA, AC, CA, CC\}$ over the DNA alphabet and the 2-block language K_1^+. Let $S_1 = \mathrm{Sub}_2(K_1^+)$. Then, S_1 is equal to K_1 and $S_1 \cap \tau(S_1)$ is empty. Hence, the language K_1^+ is a $(\tau, H_{0,2})$-bond-free subset of $(\Sigma^2)^+$. Moreover, there is no word v in $\Sigma^2 - K_1$ such that the language $(K_1 \cup \{v\})^+$ is $H_{0,2}$-bond-free. For example, if $v = AG$ then GC would be a subword of length 2 of $(K_1 \cup \{v\})^+$ such that $GC = \tau(GC)$. On the other hand, it is possible to add AG as a subword with the constraint that AG cannot be followed by CA or CC. In fact we can add also GA as a subword, provided that GA cannot be preceded by AC, CC, or AG. More specifically, consider the language L_2 accepted by the 2-block DFA $(\Sigma, \{1,2,3,4\}, 1, \{2,3,4\}, P_2)$, where $2, 3, 4$ are the final states and the set of productions P_2 is equal to

$\{1u \to 2,\ 1v \to 3,\ 1(AG) \to 4 \mid u = AA, CA, GA \text{ and } v = AC, CC\} \cup$
$\{2u \to 2,\ 2v \to 3,\ 2(AG) \to 4 \mid u = AA, CA, GA \text{ and } v = AC, CC\} \cup$
$\{3u \to 2,\ 3v \to 3,\ 3(AG) \to 4 \mid u = AA, CA \text{ and } v = AC, CC\} \cup$
$\{4(AG) \to 4,\ 4(AC) \to 3,\ 4(AA) \to 2\}$.

The language L_2 is a proper superset of K_1^+ and is a $(\tau, H_{0,2})$-bond-free subset of $(\Sigma^2)^+$. In fact it can be shown that L_2 is maximal with this property [17].

Theorem 4. *Let d be a fixed value in $\{0, 1\}$. The following problem is computable in polynomial time. **Input:** k-block DFA A such that $L(A)$ is a $(\theta, H_{d,k})$-bond-free subset of $(\Sigma^k)^+$. **Output:** Y/N, depending on whether $L(A)$ is maximal with that property. Moreover, if $L(A)$ is not maximal, output a minimal-length word $w \in (\Sigma^k)^+ - L(A)$ such that $L(A) \cup \{w\}$ is a $(\theta, H_{d,k})$-bond-free subset of $(\Sigma^k)^+$.*

4 Construction Methods for the Hamming Case

In this section we describe methods for constructing $(\tau, H_{d,k})$-bond-free languages. We focus on languages that are subsets of $(\Sigma^k)^+$ or $\Sigma^k\Sigma^*$. *We assume throughout that k and d are integers, with $k \geq 1$ and $0 \leq d < k$, and τ is the DNA involution.* Moreover, we introduce the subword closure operation, $\otimes$, which plays an important role in the sequel.

Let d be a nonnegative integer and let S and S_1 be languages containing only words of the same length k, for some positive integer k. The *Hamming ball* $H_d(S)$ of S is the set $\{v \mid H(v, z) \leq d$, for some $z \in S\}$. Note that $H_d(S) = S$ when $d = 0$. The *subword closure* $S^\otimes$ of S is the set $\{w \in \Sigma^* \mid |w| \geq k,\ \mathrm{Sub}_k(w) \subseteq S\}$. We note that $S_1 \subseteq S$ iff $S_1{}^\otimes \subseteq S^\otimes$. This implies that, if $S_1 \neq S$ then $S_1{}^\otimes \neq S^\otimes$. Moreover we note that, given S, one can construct in linear time a DFA accepting $S^\otimes$ [17].

4.1 Direct Methods

Here we consider analytical methods that do not rely on previously constructed languages. The first method is based on the concept of a template and the operation '$\cdot$': $0 \cdot 0 = C$, $0 \cdot 1 = G$, $1 \cdot 0 = T$, $1 \cdot 1 = A$ [2]. This operation is extended to binary words of the same length in a natural manner. A *k-template* is any binary word of length k. If x is a k-template and E is subset of $\{0,1\}^k$ then $x \cdot E = \{x \cdot v \mid v \in E\}$. The construction method of [2] involves choosing a k-template x and a code E such that $x \cdot E$ satisfies a desired property. In our case, we are interested in k-templates x such that

$$H(x_2x_1,(x_4x_3)^R) > d, \tag{1}$$

for all prefixes x_1 and x_3 of x and suffixes x_2 and x_4 of x.

Theorem 5. *Let x be a k-template satisfying (1). Then the language $(x\cdot\{0,1\}^k)^+$ is $(\tau, H_{d,k})$-bond-free.*

Observe that the cardinality of the code $x \cdot \{0,1\}^k$ is 2^k. The advantage of the method of templates is that properties of the template x, which is a simple object, are passed gracefully to the code $x \cdot E$, where E is any subset of $\{0,1\}^k$. We note that many of the templates listed in [2] satisfy (1).

We introduce now another direct construction method. The bond-free language is again of the form K^+, where K is a code of fixed-length. Moreover there is a set I of positions in which the codeword symbols are always in $\{A,C\}$. The method is described more formally in the next theorem. The notation $k\,\%\,2$ stands for the remainder of the integer division $k/2$, and $v[i]$ stands for the symbol of the word v at position i.

Theorem 6. *Let I be a nonempty subset of $\{1,\ldots,k\}$ of cardinality $\lfloor k/2\rfloor + 1 + \lfloor (d+k\,\%\,2)/2\rfloor$. Then the language K^+ is $(\tau, H_{d,k})$-bond-free, where*

$$K = \{v \in \Sigma^k \mid \text{ if } i \in I \text{ then } v[i] \in \{A,C\}\}.$$

Let l be the quantity $\lfloor k/2\rfloor + 1 + \lfloor (d+k\,\%\,2)/2\rfloor$ that appears in the above theorem. The size of the code K is 2^l4^{k-l}. On the other hand the method of k-templates produces codes K of size 2^k. Obviously, $2^l4^{k-l} \geq 2^k$. Moreover, one can verify that $k = l$ iff d is in $\{k-2, k-3\}$. An advantage of the method of Theorem 6 is that we can construct $(\tau, H_{d,k})$-bond-free languages with a large ratio d/k. Another advantage of some codes K defined in the previous theorem is that one can encode and decode information in linear time [17].

4.2 Methods Based on the Catenation Closure

The main idea here is that the catenation closure of Q^{d+1}, that is the language $(Q^{d+1})^+$, is $(\tau, H_{d,k})$-bond-free if Q is of length q with the property that Q^+ is $(\tau, H_{0,q})$-bond-free. The correctness of the method is based on the following theorem.

Theorem 7. *Let j and q be positive integers and let L be a subset of $\Sigma^{jq}\Sigma^*$. If L is $(\tau, H_{t,q})$-bond-free, for some integer $t \geq 0$, then it is also $(\tau, H_{d,k})$-bond-free, where $d = j(t+1) - 1$ and $k = jq$.*

Observe that for $t = 0$, the above theorem says that nearly every language that is $(\tau, H_{0,q})$-bond-free is *inherently* $(\tau, H_{d,k})$-bond-free for any $d > 0$ and any $k \geq q(d+1)$. This is a connection between the properties **P1** and **P5** considered in Section 1.

With the notation of the above theorem, let Q be a code of length q such that the language Q^+ is $(\tau, H_{t,q})$-bond-free. Let $K = Q^j$ and let $k = jq$. The code Q could be defined by some direct method, or by brute force for small values of q and t. In either case, the language K^+ is $(\tau, H_{d,k})$-bond-free.

In the case of $t = 0$, we have that $j = d + 1$ and the cardinality of the code K is $|Q|^{d+1}$, which can be larger than the cardinality of the codes defined in Theorem 6 with the same parameters. For example, if $Q = \{A, C\}^2 \Sigma \cup \{A, C\}\{G, T\}\{A, C\}$, then the code $K = Q^{d+1}$ consists of 24^{d+1} codewords. On the other hand, if the code K is defined using Theorem 6 for $k = 3(d + 1)$ then the cardinality of K is equal to 16^{d+1}.

The following observation can be viewed as a converse type of Theorem 7.

Theorem 8. *Let K be any set of words such that the language K^+ is $(\tau, H_{d,k})$-bond-free, for some integers $d \geq 0$ and $k \geq 1$. Then the language K^+ is also $(\tau, H_{0,k-d})$-bond-free.*

4.3 All Maximal (Hamming) Bond-Free Languages

With the results of Section 3 in mind [17], we understand that the languages of the form K^+ obtained by the preceding methods are not necessarily maximal. In what follows we discuss methods of obtaining new bond-free languages, possibly maximal, from old ones using the subword closure operation $\otimes$. We also need the following, slightly restricted, version of the subword closure of S, where S is any code of fixed length k, $S^\oplus \triangleq S^\otimes \cap (\Sigma^k)^+$. We call $S^\oplus$ the *block closure* of S.

Theorem 9. *Let S be a set of words of fixed length k. Then each of the languages $S^\otimes$ and $S^\oplus$ is $(\tau, H_{d,k})$-bond-free iff*

$$\tau(S) \cap H_d(S) = \emptyset. \tag{2}$$

Using the above observation we can extend $(\tau, H_{d,k})$-bond-free languages of the form K^+, such as those constructed earlier, as follows – we assume the words of K are of fixed length k. Let $S = \mathrm{Sub}_k(K^2) = \mathrm{Sub}_k(K^+)$. Then S satisfies (2) and, therefore, the language $S^\otimes$ is a $(\tau, H_{d,k})$-bond-free language that includes K^+. Next consider any code K of length k satisfying property **P3**$[d, k]$ – recall from Section 1 that such codes have been studied in [5] and [21]. Using again the above theorem it follows that $K^\otimes$ is a $(\tau, H_{d,k})$-bond-free language.

The question that arises now is when the bond-free languages of Theorem 9 are maximal. The following result addresses this question. In fact we show a complete structural characterization of all maximal $(\tau, H_{d,k})$-bond-free subsets of $(\Sigma^k)^+$ and $\Sigma^k \Sigma^*$.

Theorem 10. *The class of all maximal $(\tau, H_{d,k})$-bond-free subsets of $(\Sigma^k)^+$ is finite and equal to $\{S^\oplus \mid S \subseteq \Sigma^k$ and S is maximal satisfying $\tau(S) \cap H_d(S) = \emptyset\}$. In particular, if $d = 0$ then this class is equal to $\{S^\oplus \mid S \cup \tau(S) = \{v \in \Sigma^k \mid v \neq \tau(v)\},\ \tau(S) \cap S = \emptyset\}$.*

Note: *The above theorem holds also for subsets of $\Sigma^k\Sigma^*$ if we replace $S^\oplus$ with $S^\otimes$.*

According to Theorem 10, if K is a maximal subset of Σ^k satisfying $\tau(K) \cap H_d(K)$ then the language $K^\oplus$ is a maximal $(\tau, H_{d,k})$-bond-free subset of $(\Sigma^k)^+$. In the case of $d = 0$ the characterization of the maximal bond-free languages is quite explicit: Define any partition $\{S, \tau(S)\}$ of the set $\{v \in \Sigma^k \mid v \neq \tau(v)\}$ and then compute $S^\oplus$; this language will be maximal. The language L_2 considered in Example 3.1 is a particular instance of this type of construction [17].

The above theorem implies that every k-block $(\tau, H_{d,k})$-bond-free language L is included in a *regular* maximal such language. Statements of this type with L being regular have been obtained for various code-related properties and are of particular interest in the theory of codes [12]. In our case it is also interesting to note that the language L is not necessarily regular.

5 Discussion

We have considered the problem of undesirable bonds and proposed the property of (θ, sim)-bond-freedom for DNA languages, which addresses this problem when bonds between imperfect complements of DNA molecules are permitted. Using recent language theoretic tools, we were able to establish various decidability results about (θ, sim)-bond-freedom. The case where sim is the Hamming similarity has been considered by many authors. In this case, we have demonstrated interesting connections between our property and those of other authors, and have identified general construction *methods*. In particular, we have identified all DNA languages that are maximal with respect to the new property. This result is also applicable to the case of the θ-k-code property of [10]. Directions for future research include the following: (i) Derive a methodology for defining properties of DNA languages that would be able to address the uniqueness problem – called positive design problem in [18] – as independently of the application as possible. (ii) Elaborate on the proposed construction methods to obtain concrete constructions of languages that, in addition to being bond-free, they satisfy additional properties such as uniqueness and fixed GC-ratio. (iii) Explore further the subword closure operation from a theoretical at least point of view.

Acknowledgements

Research was partially supported by the Canada Research Chair Grant to L.K., NSERC Discovery Grants R2824A01 to L.K. and R220259 to S.K., and by the Grant Agency of Czech Republic, Grant 201/02/P079 to P.S.

References

1. Andronescu, M., Dees, D., Slaybaugh, L., Zhao, Y., Condon, A., Cohen, B., Skiena, S.: Algorithms for testing that sets of DNA words concatenate without secondary structure. In: [7], 182-195
2. Arita, M., Kobayashi, S.: DNA sequence design using templates. New Generation Computing **20** (2002) 263–277
3. Baum, E.: DNA sequences useful for computation. Pre-proceed. 2nd DIMACS Workshop on DNA-based computers, (1996) 122–127
4. Chen, J., Reif, J. (eds): Pre-proceed. 9th International Workshop on DNA-Based Computers, Madison, Wisconsin, 2003. Lecture Notes in Computer Science, Vol. 2943. Springer-Verlag, Berlin Heidelberg New York (2004)
5. Condon, A.E., Corn, R.M., Marathe, A.: On combinatorial DNA word design. Journal of Computational Biology, **8**:3 (2001) 201–220
6. Feldkamp, U., Saghafi, S., Rauhe, H.: DNASequenceGenerator - A program for the construction of DNA sequences. In: [11], 179–189
7. Hagiya, M., Ohuchi, A. (eds): Pre-proceed. 8th International Workshop on DNA-Based Computers, Saporo, Japan, 2002. Lecture Notes in Computer Science, Vol. 2568. Springer-Verlag, Berlin Heidelberg New York (2003)
8. Head, T.: Relativised code concepts and multi-tube DNA dictionaries. In: Calude, C.S., Păun, G. (eds.): Finite Versus Infinite: Contributions to an Eternal Dilemma. Springer-Verlag, London (2000) 175–186
9. Hussini, S., Kari, L., Konstantinidis, S.: Coding properties of DNA languages. In: [11], 107-118
10. Jonoska, N., Mahalingam, K.: Languages of DNA based code words. In: [4], 58–68
11. Jonoska, N., Seeman, N.C. (eds): Pre-proceed. 7th International Workshop on DNA-Based Computers, Tampa, Florida, 2001. Lecture Notes in Computer Science, Vol. 2340. Springer-Verlag, Berlin Heidelberg New York (2002)
12. Jürgensen, H., Konstantinidis, S.: Codes. In: G. Rozenberg, A. Salomaa (eds): Handbook of Formal Languages, Vol. I. Springer-Verlag, Berlin Heidelberg New York (1997) 511–607
13. Kari, L., Kitto, R., Thierrin, G.: Codes, involutions and DNA encoding. In: Brauer, W., Ehrig, H., Karhumäki, J., Salomaa, A. (eds.): Lecture Notes in Computer Science, Vol. 2300. Springer-Verlag, Berlin Heidelberg New York (2002) 376–393
14. Kari, L., Konstantinidis, S.: Language equations, maximality and error detection. Journal of Computer and System Sciences (to appear)
15. Kari, L., Konstantinidis, S., Losseva, E., Wozniak, G.: Sticky-free and overhang-free DNA languages. Acta Informatica **40** (2003) 119–157
16. Kari, L., Konstantinidis, S., Sosík, P.: On properties of bond-free DNA languages. Tech. report 609, Dept. Computer Science, Univ. Western Ontario, Canada (2003)
17. Kari, L., Konstantinidis, S., Sosík, P.: Bond-free languages: formalizations, maximality and construction methods. Tech. report 2004-01, Dept. Mathematics and Computing Science, Saint Mary's University, Halifax, Canada (2004). Electronic form available at `http://www.stmarys.ca/academic/science/compsci/`
18. Mauri, G., Ferretti, C.: Word Design for Molecular Computing: A Survey. In: [4], 37–46
19. Reif, J.H., LaBean, T.H., Pirrung, M., Rana, V.S., Guo, B., Kingsford, C., Wickham, G.S.: Experimental construction of very large scale DNA databases with associative search capability. In: [11], 231–247

20. Tanaka, F., Nakatsugawa, M., Yamamoto, M., Shiba, T., Ohuchi, A.: Developing support system for sequence design in DNA computing. In: [11], 129–137.
21. Tulpan, D.C., Hoos, H.H., Condon, A.E.: Stochastic Local Search Algorithms for DNA Word Design. In: [7], 229–241.