On Hairpin-Free Words and Languages

Lila Kari[1], Stavros Konstantinidis[2], Petr Sosík[3,⋆], and Gabriel Thierrin[4]

[1] Department of Computer Science, The University of Western Ontario, London, ON, N6A 5B7, Canada
lila@csd.uwo.ca
[2] Dept. of Mathematics and Computing Science, Saint Mary's University, Halifax, Nova Scotia, B3H 3C3, Canada
s.konstantinidis@smu.ca
[3] Institute of Computer Science, Silesian University, 74601 Opava, Czech Republic
petr.sosik@fpf.slu.cz
[4] Department of Mathematics, The University of Western Ontario, London, ON, N6A 5B7, Canada
thierrin@uwo.ca

Abstract. The paper examines the concept of hairpin-free words motivated from the biocomputing and bioinformatics fields. Hairpin (-free) DNA structures have numerous applications to DNA computing and molecular genetics in general. A word is called hairpin-free if it cannot be written in the form $xvy\theta(v)z$, with certain additional conditions, for an involution θ (a function θ with the property that θ^2 equals the identity function).
We consider three involutions relevant to DNA computing: a) the mirror image function, b) the DNA complementarity function over the DNA alphabet $\{A, C, G, T\}$ which associates A with T and C with G, and c) the Watson-Crick involution which is the composition of the previous two. We study elementary properties and finiteness of hairpin (-free) languages w.r.t. the involutions a) and c). Maximal length of hairpin-free words is also examined. Finally, descriptional complexity of maximal hairpin-free languages is determined.

Keywords: DNA computing, DNA hairpin, involution, finite automaton

1 Introduction

The primary motivation for study of hairpin-free structures in this paper arises from the areas of DNA computing and bioinformatics, where such structures are important for the design of information-encoding DNA molecules. A single strand DNA molecule can be formally described as a string over the DNA alphabet $\Delta = \{A, C, T, G\}$. These four symbols correspond to nucleotides attached to a sugar-phosphate backbone. Two single strands can bind (anneal) to each other if they have opposite polarity (the strand's orientation in space) and

⋆ Corresponding author

C. De Felice and A. Restivo (Eds.): DLT 2005, LNCS 3572, pp. 296–307, 2005.

are pairwise Watson-Crick complementary: A is complementary to T, and C to G. The ability of DNA strands to anneal to each other allows for creation of various secondary structures. A *DNA hairpin* is a particular type of secondary structure important in many applications. An example of a hairpin structure is shown in Figure 1. The figure characterizes the case when θ is the Watson-Crick antimorphic involution (see the next section for exact definition).

Fig. 1. A single-stranded DNA molecule forming a hairpin loop

Hairpin-like secondary structures play an important role in insertion/deletion operations with DNA. Hairpin-freedom is crucial in the design of primers for the PCR reaction [4]. Hairpins are the main tool used in the Whiplash PCR computing techniques [17]. In [19] hairpins serve as a binary information medium for DNA RAM. Last, but not least, hairpins are basic components of recently investigated "smart drugs" [1]. Therefore, in the above mentioned applications, one needs to construct (sets of) hairpin(-free) DNA molecules, or to test existing sets of DNA molecules for hairpin-freedom and study their properties. We refer e.g. to [16] for an overview of design of DNA languages without hairpins and other undesired bonds. Coding properties of hairpin-free languages have been studied in [11, 12]. Hairpins have also been studied in the context of bio-operations occurring in single-celled organisms (see the hairpin inversion operation defined as one of the three molecular operations that accomplish gene assembly in ciliates [6, 8]).

In addition, the presented results also contribute to mathematical characterization of regularities in formal words and languages. In this sense the definition of hairpin-free words can be understood as a generalization of repetition-freedom. A word u is called *hairpin-k-free* if $u = xvy\theta(v)z$ implies $|v| < k$, for a chosen involution θ. Considering the special case when $k = 1$, θ is the identity involution and y is the empty word, we obtain the square-freedom (see below).

For a general overview and fundamental results in combinatorics on words, the reader is referred to [5, 13]. If w is an nonempty word, then ww is called a square and www is called a cube. Important questions about avoiding squares and cubes in infinite words have been answered in [7]. See [14] for combinatorics on finite words. Words of the form $uvyvz$ with a bounded length of y have been studied e.g. in [3]. Unfortunately, many techniques and results known in combinatorics on words are non-applicable in the case of hairpin-free words. One of the main reasons is that in the case of an antimorphic involution, analogies of the famous defect theorem and its consequences are no longer valid.

The paper is organized as follows. Section 2 introduces basic formal concepts and definitions. In Section 3 we present the concept of hairpin-free words and languages and study their elementary properties. Problems related to the finite-

ness of hairpin-free languages are addressed in Section 4. Finally, in Section 5 we study descriptional complexity of hairpin (-free) languages with regards to possible applications.

2 Formal Language Prerequisites

We will use X to denote a finite alphabet and X^* its corresponding free monoid. The cardinality of the alphabet X is denoted by $|X|$. The empty word is denoted by 1, and $X^+ = X^* - \{1\}$. A *language* is an arbitrary subset of X^*. For a word $w \in X^*$ and $k \geq 0$, we denote by w^k the word obtained as catenation of k copies of w. Similarly, X^k is the set of all words from X^* of length k. By convention, $w^0 = 1$ and $X^0 = \{1\}$. We also denote $X^{\leq k} = X^0 \cup X^1 \cup \ldots \cup X^k$. By convention, $X^{\leq k} = \emptyset$ for $k < 0$.

A mapping $\psi : X^* \rightarrow X^*$ is called a *morphism* (*anti-morphism)* of X^* if $\psi(uv) = \psi(u)\psi(v)$ (respectively $\psi(uv) = \psi(v)\psi(u)$) for all $u, v \in X^*$, and $\phi(1) = 1$. See [9] for a general overview of morphisms. An involution $\theta : X \longrightarrow X$ is defined as a map such that θ^2 is the identity function. An involution θ can be extended to a morphism or an antimorphism over X^*. In both cases θ^2 is the identity over X^* and $\theta^{-1} = \theta$. If not stated otherwise, θ refers to an arbitrary morphic or antimorphic involution in this paper.

In our examples we shall refer to the DNA alphabet $\Delta = \{A, C, T, G\}$. By convention, DNA strands are described by strings over this alphabet in orientation from 5' to 3' end. On this alphabet several involutions of interest are defined. The simplest involution is the identity function ϵ. An antimorphic involution which maps each letter of the alphabet to itself is called a *mirror involution* and it is denoted by μ. The DNA *complementarity involution* γ is a morphism given by $\gamma(A) = T$, $\gamma(T) = A$, $\gamma(C) = G$, $\gamma(G) = C$. For example, $\epsilon(ACGCTG) = ACGCTG = \mu(GTCGCA) = \gamma(TGCGAC)$.

Finally, the antimorphic involution $\tau = \mu\gamma$ (the composite function of μ and γ, which is also equal to $\gamma\mu$), called the *Watson-Crick involution*, corresponds to the DNA bond formation of two single strands. If for two strings $u, v \in \Delta^*$ it is the case that $\tau(u)v$, then the two DNA strands represented by u, v anneal as Watson-Crick complementary sequences.

A nondeterministic finite automaton (*NFA*) is a quintuple $A = (S, X, s_0, F, P)$, where S is the finite and nonempty set of states, s_0 is the start state, F is the set of final states, and P is the set of productions of the form $sx \rightarrow t$, for $s, t \in S$, $x \in X$. If for every two productions $sx_1 \rightarrow t_1$ and $sx_2 \rightarrow t_2$ of an NFA we have that $x_1 \neq x_2$ then the automaton is called a *DFA* (deterministic finite automaton). The language accepted by the automaton A is denoted by $L(A)$. The *size* $|A|$ of the automaton A is the number $|S| + |P|$. We refer to [18] for further definitions and elements of formal language theory.

3 Involutions and Hairpins

Definition 1. *If θ is a morphic or antimorphic involution of X^* and k is a positive integer, then a word $u \in X^*$ is said to be θ-k-hairpin-free or simply hp(θ,k)-free if $u = xvy\theta(v)z$ for some $x, v, y, z \in X^*$ implies $|v| < k$.*

Notice that the words 1 and $a \in X$ are hp(θ,1)-free. More generally, words of length less than $2k$ are hp(θ,k)-free. If we interpret this definition for the DNA alphabet Δ and the Watson-Crick involution τ, then a hairpin structure with the length of bond greater than or equal to k is a word that is not hp(τ,k)-free.

Definition 2. *Denote by $hpf(\theta, k)$ the set of all hp(θ,k)-free words in X^*. The complement of $hpf(\theta, k)$ is $hp(\theta, k) = X^* - hpf(\theta, k)$.*

Notice that $hp(\theta, k)$ is the set of words in X^* which are hairpins of the form $xvy\theta(v)z$ where the length of v is at least k. It is also the case that $hp(\theta, k+1) \subseteq hp(\theta, k)$ for all $k > 0$.

Definition 3. *A language L is called θ-k-hairpin-free or simply $hp(\theta, k)$-free if $L \subseteq hpf(\theta, k)$.*

It is easy to see from the definition that a language L is hp(θ, k)-free if and only if $X^*vX^*\theta(v)X^* \cap L = \emptyset$ for all $|v| \geq k$. An analogous definition was given in [11] where a θ-k-hairpin-free language is called θ-subword-k-code. The authors focused on their coding properties and relations to other types of codes. Restrictions on the length of a hairpin were also considered, namely that $1 \leq |y| \leq m$ for some $m \geq 1$. The reader can verify that our Proposition 3 remains valid and the results in Section 5 change only slightly if we apply this additional restriction.

Example.

1. Let $X = \{a, b\}$ with $\theta(a) = b, \theta(b) = a$. Then $hpf(\theta, 1) = a^* \cup b^*$.
   This example shows that in general the product of hp$(\theta, 1)$-free words is not an hp$(\theta, 1)$-free word. Indeed, a and b are hp$(\theta, 1)$-free, but the product ab is not.
2. If $\theta = \gamma$ is the DNA complementary involution over Δ^*, then:
$$hpf(\theta, 1) = \{A, C\}^* \cup \{A, G\}^* \cup \{T, C\}^* \cup \{T, G\}^*$$
3. Let $\theta = \mu$ be the mirror involution and let $u \in hpf(\theta, 1)$. Since $\theta(a) = a$ for all $a \in X$, u cannot contain two occurrences of the same letter a. This implies that $hpf(\theta, 1)$ is finite. For example, if $X = \{a, b\}$, then:
$$hpf(\theta, 1) = \{1, a, b, ab, ba\}$$

We focus first on the important special case when $k = 1$. Observe that $hp(\theta, 1) = \bigcup_{a \in X} X^*aX^*\theta(a)X^*$. Recall also the definition of an embedding order: $u \leq_e w$ if and only if

$$u = u_1u_2 \cdots u_n,\ w = v_1u_1v_2u_2 \cdots v_nu_nv_{n+1}$$

for some integer n with $u_i, v_j \in X^*$.

A language L is called *right $\leq_e$-convex* [20] if $u \leq_e w,\ u \in L$ implies $w \in L$. The following result is well known: *All languages (over a finite alphabet) that are right $\leq_e$-convex are regular.*

Proposition 1. *The language $hp(\theta, 1)$ is right $\leq_e$-convex.*

Proof. If $u = u_1u_2 \in hp(\theta, 1)$ and $v_1, v_2, v_3 \in X^*$, then $w = v_1u_1v_2u_2v_3 \in hp(\theta, 1)$. Therefore, if $u \in hp(\theta, 1)$ and $u \leq_e w$, then w can be constructed from u by a sequence of insertions, and hence $w \in hp(\theta, 1)$. □

Let $L \subseteq X^*$ be a nonempty language and let:

$$S(L) = \{w \in X^* | u \leq_e w, u \in L\}.$$

Hence $S(L)$ is the set of all the words $w \in X^*$ that can be expressed in the form $w = x_1u_1x_2u_2 \cdots x_nu_nx_{n+1}$ with $u = u_1u_2 \cdots u_n \in L$ and $x_i \in X^*$.

Recall further that a set H with $\emptyset \neq H \subseteq X^+$ is called a *hypercode* over X^* *iff* $x \leq_e y$ and $x, y \in H$ imply $x = y$. That is, a hypercode is an independent set with respect to the embedding order.

Proposition 2. *Let θ be a morphic or antimorphic involution. Then there exists a unique hypercode H such that $hp(\theta, 1) = S(H)$.*

Proof. Let $H = \bigcup_{a \in X} a\theta(a)$, then $S(H) = \bigcup_{a \in X} X^*aX^*\theta(a)X^* = hp(\theta, 1)$. The uniqueness of H is immediate. □

Example. Consider the hypercodes for the earlier three examples.

1. For $X = \{a, b\}$ and the involution (morphic or antimorphic) $\theta(a) = b$, $\theta(b) = a$, the hypercode is $H = \{ab, ba\}$.
2. For the DNA complementarity involution γ we have $H = \{AT, TA, CG, GC\}$.
3. The mirror involution over $\{a, b\}^*$ gives the hypercode $H = \{aa, bb\}$.

Proposition 1, true for the case $k = 1$, cannot in general be extended to the case $k > 1$ as the language $hp(\theta, 2)$ is not $\leq_e$-convex. However, the weaker regularity property remains valid. Note that $hp(\theta, k) = \bigcup_{|w| \geq k} X^*wX^*\theta(w)X^*$.

Proposition 3. *The languages $hp(\theta, k)$ and $hpf(\theta, k)$, $k \geq 1$, are regular.*

Proof. One can easily derive $hp(\theta, k) \bigcup_{|w|=k} X^*wX^*\theta(w)X^*$. Every language $X^*wX^*\theta(w)X^*$with $|w| = k$ is regular, hence $hp(\theta, k)$ is a union of a finite number of regular languages. Therefore both $hp(\theta, k)$ and its complement $hpf(\theta, k)$ are regular. □

4 Finiteness of Hairpin-Free Languages

In this section we give the necessary and sufficient conditions under which the language $hpf(\theta, k)$ is finite, for a chosen $k \geq 1$. We study first the interesting special case of μ, the mirror involution, over a binary alphabet X.

Recall that $hp(\mu, k)$ is the set of all words containing two non-overlapping mirror parts of length at least k. In the next proposition we show that the longest hp$(\mu, 4)$-free word is of length 31. This also implies that the language $hpf(\mu, 4)$ is finite. The proof requires several technical lemmata whose proofs are omitted due to page limitations and can be found in [15]. In these lemmata we assume that $|X| = 2$.

Definition 4. *A* run *in a word w is a subword of w of the form c^k, with $c \in X$ and $k \geq 1$, such that $w = uc^kv$ for some word u that does not end with c, and some word v that does not start with c.*

Lemma 1. *Suppose that w is a word in $hpf(\mu, 4)$. The following statements hold true.*

1. *If a^i is any run in w then $i \leq 7$. If the run is internal then $i \leq 5$.*
2. *The word w cannot contain three different runs $a^{i_1}, a^{i_2}, a^{i_3}$ with $i_1, i_2, i_3 \geq 3$. If w contains two runs a^j and a^i with $i, j \geq 3$ then w starts with a^jba^i, or w ends with a^iba^j. Moreover not both i and j can be greater than 3.*
3. *The word w cannot contain three different internal runs a^2. If w contains two internal runs a^2 then they occur as in $\cdots ba^2ba^2b\cdots$.*
4. *The above statements also hold if we replace a with b and vice-versa.*

Lemma 2. *Suppose that a word in $hpf(\mu, 4)$ contains a subword w of the form*

$$ab^{x_1}a^{y_1}\cdots b^{x_n}a^{y_n}b,$$

with $n \geq 3$ and $x_i, y_i \geq 1$ for each i. Then there are at most three indices i such that $x_i = y_i = 1$.

Lemma 3. *Suppose that a word w is in $hpf(\mu, 4)$ and contains two runs c^j and c^i with $i, j \geq 3$ and $c \in X$. Then $|w| \leq 31$.*

Lemma 4. *Suppose that a word w is in $hpf(\mu, 4)$ and contains no two runs c^j and c^i with $i, j \geq 3$ and contains two internal runs b^2 and one internal run b^y with $y \geq 3$ and w is of the following form*

$$a^{y_0}b^{x_1}a^{y_1}\cdots b^{x_n}a^{y_n}(b^{x_{n+1}}a^{y_{n+1}}),$$

where all y_i's and x_j's are positive except possibly for y_{n+1}. Then $|w| \leq 31$.

Lemma 5. *If a word w is in $hpf(\mu, 4)$ and of the form*

$$a^{y_0}b^{x_1}a^{y_1}\cdots b^{x_n}a^{y_n}(b^{x_{n+1}}a^{y_{n+1}}),$$

such that $y_0, x_{n+1} \geq 3$, and $2 \geq y_{n+1} \geq 0$, and $2 \geq x_i, y_i > 0$ for all $i = 1, \ldots, n$, then $|w| \leq 30$. Moreover, the following word of length 30 satisfies the above premises:

$$a^7b^2ab^2ababaa^2ba^2b^7.$$

Proposition 4. *Let X be a binary alphabet. For every word $w \in X^*$ in $hpf(\mu, 4)$ we have that $|w| \leq 31$. Moreover the following word of length 31 is in $hpf(\mu, 4)$*

$$a^7ba^3bababab^2ab^2a^2b^7.$$

Proof. Without loss of generality we can assume that w starts with a. Then w would be of the form

$$a^{y_0}b^{x_1}a^{y_1}\cdots b^{x_n}a^{y_n}(b^{x_{n+1}}a^{y_{n+1}}),$$

where all y_i's and x_j's are positive except possibly for y_{n+1}. We distinguish the following cases.

Case 1: There are two runs c^i and c^j in w with $i, j \geq 3$. By Lemma 3, $|w| \leq 31$ as required.

In the next 7 cases, we assume that the first case does not hold and that there is exactly one run a^δ in w with $\delta \geq 3$.

Case 2: The run a^δ is a^{y_0} and there is a run b^i with $i \geq 3$. If $x_{n+1} \geq 3$ then Lemma 5 implies that $|w| \leq 30$. So assume that $x_{n+1} \leq 2$. If there are two internal runs b^2 in w then Lemma 4 implies that $|w| \leq 31$. So assume further that there is at most one internal run b^2. Note that if $x_{n+1} = 2$ and $y_{n+1} > 0$ then $b^{x_{n+1}}$ is the run b^2. Let g be the quantity $e|b^2a| + x_{n+1} + y_{n+1}$, where $e = 0$ if $x_{n+1} = 2$ and $y_{n+1} > 0$, and $e = 1$ if $x_{n+1} = 1$ or $y_{n+1} = 0$. Hence, $g \leq 6$. Moreover, $|w| \leq 7 + 3|ba| + 2|ba^2| + |b^5a| + g \leq 31$.

Case 3: The run a^δ is a^{y_0} and there is no run b^i with $i \geq 3$. Using again the quantity g of Case 2, we have that $|w| \leq 7 + 3|ba| + 2|ba^2| + |b^2a| + g \leq 28$.

Case 2': The run a^δ is $a^{y_{n+1}}$ and there is a run b^i with $i \geq 3$. Then the word $\mu(w)$ is of the same form as the word w is and the run b^i occurs in $\mu(w)$. Hence, Case 2 applies to $\mu(w)$ and, therefore, both $\mu(w)$ and w are of length at most 31.

Case 3': The run a^δ is $a^{y_{n+1}}$ and there is no run b^i with $i \geq 3$. Then the word $\mu(w)$ is of the same form as the word w is and no run b^i, with $i \geq 3$, occurs in $\mu(w)$. Hence, Case 3 applies to $\mu(w)$ and, therefore, both $\mu(w)$ and w are of length at most 28.

Case 4: The run a^δ is internal and there is one internal run b^j with $j \geq 3$. Then $j, \delta \leq 5$. If w contains two internal runs b^2 then Lemma 4 implies that $|w| \leq 31$. Next assume that w contains at most one internal run b^2 and consider the quantity $g = e|b^2a| + x_{n+1} + y_{n+1}$ as in Case 2. If w contains at most one internal run a^2 then

$$|w| \leq y_0 + 3|ba| + |ba^2| + |ba^5| + |b^5a| + g \leq 2 + 6 + 3 + 6 + 6 + 6 = 29.$$

Next assume further that w contains two internal runs a^2. Then Lemma 1 implies that w contains ba^2ba^2b. Also,

$$|w| \leq 2 + 6 + 2|ba^2| + 6 + 6 + g \leq 26 + g.$$

If $x_{n+1} = 2$ and $y_{n+1} > 0$ then $e = 0$ and $|w| \leq 30$. If $y_{n+1} = 0$ then $e = 1$ and $|w| \leq 31$. If $x_{n+1} = 1$ and $y_{n+1} = 1$ then $|w| \leq 31$. Finally, if $x_{n+1} = 1$ and $y_{n+1} = 2$ then w ends with aba^2, which contradicts the fact that w contains ba^2ba^2b.

Case 4': The run a^δ is internal and there is one external run b^j with $j \geq 3$. Then $y_{n+1} = 0$ and the run b^j is $b^{x_{n+1}}$, as $y_0 > 0$. Let w' be the word resulting by exchanging the letters a and b in w. Then the word $\mu(w')$ satisfies the premises of Case 2, which implies that w is of length at most 31.

Case 5: The run a^δ is internal and there is no run b^j with $j \geq 3$. Using again the quantity g of Case 2, we have that $|w| \leq y_0+3|ba|+|ba^5|+2|ba^2|+|b^2a|+g \leq 29$.
Case 6: Here the first case does not hold and there is no run a^δ with $\delta \geq 3$. If there is an internal run b^j with $j \geq 3$ then $|w| \leq y_0+3|ba|+|b^5a|+2|ba^2|+|b^2a|+g \leq 29$. If there is an external run b^j with $j \geq 3$ then $b^j = b^{x_{n+1}}$ and $y_{n+1} = 0$, and one can verify that $|w| \leq 27$. If there is no run b^j with $j \geq 3$ then one can verify that $|w| \leq 23$.

Finally, by inspection one verifies that $a^7ba^3bababab^2ab^2a^2b^7$ is indeed in $hpf(\mu, 4)$. □

Corollary 1. *Consider a binary alphabet X. Then $hpf(\mu, k)$ is finite if and only if $k \leq 4$.*

Proof. Denote $X = \{a, b\}$. By Proposition 4, the set $hpf(\mu, 4)$ is finite. Now consider the language $L_5 = (aabbab)^+$. The set of its subwords of length 5 is $Sub_5(L_5) = \{aabba, abbab, bbaba, babaa, abaab, baabb\}$. For its mirror image $\mu(L_5)$ we obtain $Sub_5(\mu(L_5)) = \{abbaa, babba, ababb, aabab, baaba, bbaab\}$. As these two sets are mutually disjoint, $L_5 \subseteq hpf(\mu, 5)$.

Finally, notice that for $k > 1$, finiteness of $hpf(\mu, k)$ implies also finiteness of $hpf(\mu, k-1)$. Hence the facts that $hpf(\mu, 4)$ is finite and $hpf(\mu, 5)$ is infinite conclude the proof. □

Proposition 5. *Let θ be a morphic or antimorphic involution. The language $hpf(\theta, k)$ over a non-singleton alphabet X is finite if and only if one of the following holds:*
(a) $\theta = \epsilon$, the identity involution;
(b) $\theta = \mu$, the mirror involution, and either $k = 1$ or $|X| = 2$ and $k \leq 4$.

Proof. (a) Let θ be a morphism. Assume first that $\theta \neq \epsilon$. Then there are $a, b \in X$, $a \neq b$, such that $\theta(a) = b$. Then $a^+ \subseteq hpf(\theta, k)$ for any $k \geq 1$, hence $hpf(\theta, k)$ is infinite.
Assume now that $\theta = \epsilon$ and let w be any word of length $\geq k|X|^k + k$. Since there exist $|X|^k$ distinct words of length k, there are at least two non-overlapping subwords of length k in w which are identical. Hence $w = xvyvz$ for some $v \in X^k$ and $x, y, z \in X^*$. Therefore $hpf(\epsilon, k)$ is finite since it cannot contain any word longer than $k|X|^k + k$.
(b) Let θ be an anti-morphism. Assuming that $\theta \neq \mu$, the same arguments as above show that $hpf(\theta, k)$ is infinite.
Assume now that $\theta = \mu$. Apparently $hpf(\mu, 1)$ is finite as shown in the examples above. For $|X| = 2$ we know that $hpf(\mu, k)$ is finite *iff* $k \leq 4$ by Corollary 1. Finally, for $|X| > 2$ and $k > 1$ the language $hpf(\mu, k)$ is infinite as it always contains the $\mathrm{hp}(\mu, 2)$-free set $(abc)^+$ (regardless to renaming the symbols). □

5 Descriptional Complexity of Hairpin(-Free) Languages

The regularity of the languages $hp(\theta, k)$ and $hpf(\theta, k)$ shown in Section 3 indicates an existence of fast algorithms deciding problems related to hairpin-

freedom. For such algorithms, a construction of automata (NFA or DFA) accepting the languages $hp(\theta, k)$ and $hpf(\theta, k)$ would be important. Therefore we investigate minimal size of these automata. We recall the following technical tools from [2], see also [10].

Definition 5. *A set of pairs of strings* $\{(x_i, y_i) \mid i = 1, 2, \ldots, n\}$ *is called a* fooling set *for a language* L *if for any* i, j *in* $\{1, 2, \ldots, n\}$,
(1) $x_i y_i \in L$, *and*
(2) if $i \neq j$ *then* $x_i y_j \notin L$ *or* $x_j y_i \notin L$.

Lemma 6. *Let* $\mathcal{F}$ *be a fooling set of a cardinality* n *for a regular language* L. *Then any NFA accepting* L *needs at least* n *states.*

Now we can characterize the minimal size of automata accepting languages $hp(\theta, k)$ and $hpf(\theta, k)$. We use the operator $\prod$ for catenation.

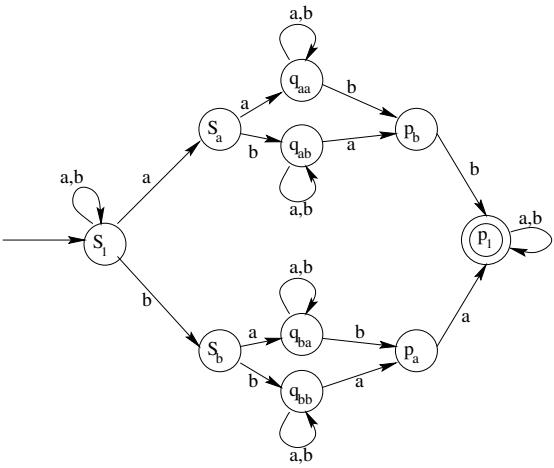

Fig. 2. An example of an NFA accepting the language $hp(\theta, 2)$

Proposition 6. *The number of states of a minimal NFA accepting the language* $hp(\theta, k)$, $k \geq 1$, *over an alphabet* X *of the cardinality* $\ell > 1$, *is between* ℓ^k *and* $3\ell^k$, *its size is at most* $3(\ell^k + \ell^{k+1})$.

Proof. Let $M_k = (S, X, s_1, F, P)$ be an NFA accepting $hp(\theta, k)$.

(i) The reader can easily verify that the set $\mathcal{F} = \{(w, \theta(w)) | w \in X^k\}$ is a fooling set for $hp(\theta, k)$. Therefore $|S| \geq \ell^k$.
(ii) Let

$$S = \{s_w, p_w \mid w \in X^{\leq k-1}\} \cup \{q_w \mid w \in X^k\}.$$

Let further $F = \{p_1\}$. The set of productions P is defined as follows:

$$\begin{array}{ll}
s_v a \to s_w \text{ if and only if } va = w, & \text{for each } v \in X^{\leq k-2},\ a \in X;\\
s_v a \to q_w \text{ if and only if } va = w, & \text{for each } v \in X^{k-1},\ a \in X;\\
q_w a \to q_w & \text{for all } w \in X^k,\ a \in X;\\
q_w a \to p_v \text{ if and only if } \theta(av) = w, & \text{for each } v \in X^{k-1},\ a \in X;\\
p_w a \to p_v \text{ if and only if } av = w, & \text{for each } v \in X^{\leq k-2},\ a \in X.
\end{array}$$

Finally, let $s_1 a \to s_1$ and $p_1 a \to p_1$ for all $a \in X$. An example of the automaton M_k for the case $X = \{a, b\}$, $k = 2$ and θ being an antimorphism, $\theta(a) = b$, $\theta(b) = a$, is at Fig. 2. The reader can verify that $L(M_k) = hp(\theta, k)$, and that $|S| \leq 3\ell^k$, $|P| \leq 3\ell^{k+1}$, therefore $|M_k| \leq 3(\ell^k + \ell^{k+1})$. □

Note that for $\ell = 1$ we have $hp(\theta, k) = X^{2k}X^*$, therefore the size of the minimal automaton accepting $hp(\theta, k)$ is $|M_k| = 4k + 2$.

Proposition 7. *Assume that there are distinct letters $a, b \in X$ such that $a = \theta(b)$. Then the number of states of a minimal NFA accepting $hpf(\theta, k)$, $k \geq 1$, over an alphabet X with the cardinality ℓ, is at least $2^{(\ell-2)^k/2}$.*

Proof. We take into the account only the cases $\ell \geq 3$, the case $\ell = 2$ is trivial. Denote $X_1 = X \setminus \{a, b\}$. We can factorize the set $X_1^k = C_1 \cup C_2 \cup C_3$, where C_1, C_2, C_3 are mutually disjoint sets such that $\theta(C_1) = C_2$ and $\theta(x) = x$ for all $x \in C_3$. Obviously $|C_1| = |C_2|$.

Denote $m = |C_1 \cup C_3|$, then $m \geq (\ell - 2)^k/2$. Consider the set of pairs of strings

$$\mathcal{F} = \left\{ \left(\prod_{w \in D} aw, \prod_{w \in (C_2 \cup C_3) \setminus \theta(D)} aw \right) \;\middle|\; D \subseteq (C_1 \cup C_3) \right\}. \tag{1}$$

We show that $\mathcal{F}$ is a fooling set for $hpf(\theta, k)$.

(i) Consider an arbitrary pair $(x, y) \in \mathcal{F}$. Let $z \in X^k$ be a substring of xy. If z contains a, then $\theta(z)$ cannot be in xy as $\theta(a) = b$ and b is not in xy. If z does not contain a, then $z \in X_1^k$ and z is a subword of either x or y. Assume that z is a part of x. Then, by definition of C_1 and C_3, there is no occurrence of $\theta(z)$ in x which would not overlap z. Also, $\theta(z)$ is not a subword of y as $z \in D$ and hence $\theta(z) \notin (C_2 \cup C_3) \setminus \theta(D)$. If z is a subword of y, the situation is analogous. Therefore, $xy \in hpf(\theta, k)$.

(ii) Let (x, y), (x', y') be two distinct elements of $\mathcal{F}$, associated with the sets $D, D' \subseteq (C_1 \cup C_3)$ in the sense of (1). Let us assume without loss of generality that there is a $z \in D \setminus D'$. Then $\theta(z) \in (C_2 \cup C_3)$ and $\theta(z) \notin \theta(D')$, hence $\theta(z)$ is a subword of y'. Simultaneously z is a subword of x, therefore $xy' \notin hpf(\theta, k)$.

We can conclude that $|\mathcal{F}| = 2^m \geq 2^{(\ell-2)^k/2}$, and hence the statement follows by Lemma 6. □

Corollary 2. *Let X be an alphabet such that $|X| = \ell$, $\ell \geq 2$. Let there be distinct letters $a, b \in X$ such that $a = \theta(b)$. Then the number of states of a minimal DFA over the alphabet X, accepting either $hp(\theta, k)$ or $hpf(\theta, k)$, $k \geq 1$, is between $2^{(\ell-2)^k/2}$ and $2^{3\ell^k}$.*

Proof. Observe that the numbers of states of minimal DFA's accepting $hp(\theta, k)$ and $hpf(\theta, k)$ are the same since these languages are mutual complements. Then the lower bound follows by Proposition 7. The upper bound follows by Proposition 6 and by the subset construction of a DFA equivalent to the NFA M_k mentioned there. □

Corollary 3. *Consider the DNA alphabet $\Delta = \{A, C, T, G\}$ and the Watson-Crick involution τ.*

(i) The size of a minimal NFA accepting $hp(\tau, k)$ is at most $15 \cdot 4^k$. The number of its states is between 4^k and $3 \cdot 4^k$.
(ii) The number of states of either a minimal DFA or an NFA accepting $hpf(\tau, k)$ is between $2^{2^{k-1}}$ and $2^{3 \cdot 2^{2k}}$.

Note: after careful inspection of the automaton in the proof of Proposition 6, one can derive that the actual size is at most $\frac{25}{3} \cdot 4^k + \frac{14}{3}$ and the number of states do not exceed $\frac{5}{3} \cdot 4^k - \frac{2}{3}$.

The above results indicate that the size of a minimal NFA for $hp(\tau, k)$ grows exponentially with k. However, one should recall that k is the *minimal* length of bond allowing for a stable hairpin. Therefore k is rather low in practical applications and the construction of the mentioned automaton can remain computationally tractable.

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. We are indebted to Elena Losseva for drawing the figures and for valuable comments to the paper.

References

1. Y. Benenson, B. Gil, U. Ben-Dor, R. Adar, E. Shapiro, An autonomous molecular computer for logical control of gene expression. *Nature* 429 (2004), 423–429.
2. J.C. Birget, Intersection and union of regular languages and state complexity. *Information Processing Letters* 43 (1992), 185–190.
3. G.S. Brodal, R.B. Lyngsø, C.N.S. Pedersen, J. Stoye, Finding maximal pairs with bounded gap. In *Proceedings of the 10th Annual Symposium on Combinatorial Pattern Matching (CPM)*, M. Crochemore and M. Paterson, Eds., LNCS 1645 (1999), 134–149.

4. C.R. Calladine, H.R. Drew, *Understanding DNA: The Molecule and How It Works.* 2nd edition, Academic Press, San Diego, 1999.
5. C. Choffrut, J. Karhumäki, Combinatorics of words. In [18], 329–438.
6. M. Daley, L. Kari, Some properties of ciliate bio-operations. In *Procs. of DLT 2002*, M. Ito, M. Toyama, Eds., LNCS 2450 (2003), 116–127.
7. F.M. Dekking, On repetitions of blocks in binary sequences. *J. Combin. Theory Ser. A* 20 (1976), 292–299.
8. A. Ehrenfeucht, T. Harju, I. Petre, D. Prescott, G. Rozenberg, *Computation in Living Cells: Gene Assembly in Ciliates.* Springer-Verlag, Berlin, 2004.
9. T. Harju, J. Karhumäki, Morphisms. In [18], 439–510.
10. J. Jirásek, G. Jirásková, A. Szabari, State complexity of concatenation and complementation of regular languages. In *CIAA 2004, Ninth International Conference on Implementation and Application of Automata*, M. Domaratzki, A. Okhotin, K. Salomaa, S. Yu, Eds., Queen's University, Kingston, 2004, 132–142.
11. N. Jonoska, D. Kephart, K. Mahalingam, Generating DNA code words. *Congressus Numerantium* 156 (2002), 99–110.
12. N. Jonoska, K. Mahalingam, Languages of DNA based code words. In *DNA Computing, 9th International Workshop on DNA Based Computers*, J. Chen and J.H. Reif, Eds., LNCS 2943 (2004), 61–73.
13. M. Lothaire, *Algebraic Combinatorics on Words.* Cambridge University Press, 2002.
14. A. de Luca, On the combinatorics of finite words. *Theoretical Computer Science* 218 (1999), 13–39.
15. L. Kari, S. Konstantinidis, P. Sosík, G. Thierrin, *On Hairpin-Free Words and Languages.* TR 639, Dept. of Computer Science, University of Western Ontario, 2005, `http://www.csd.uwo.ca/ lila/involuti.ps`.
16. G. Mauri, C. Ferretti, Word Design for Molecular Computing: A Survey. In *DNA Computing, 9th International Workshop on DNA Based Computers*, J. Chen and J.H. Reif, Eds., LNCS 2943 (2004), 37–46.
17. J. A. Rose, R. J. Deaton, M. Hagiya, A. Suyama, PNA-mediated Whiplash PCR. In *DNA Computing, 7th International Workshop on DNA Based Computers*, N. Jonoska and N. C. Seeman, Eds., LNCS 2340 (2002), 104–116.
18. G. Rozenberg, A. Salomaa, Eds., *Handbook of Formal Languages*, vol. 1, Springer Verlag, Berlin, 1997.
19. N. Takahashi, A. Kameda, M. Yamamoto, A. Ohuchi, Aqueous computing with DNA hairpin-based RAM. In *DNA 10, Tenth International Meeting on DNA Computing*, G. Mauri, C. Ferretti, Eds., University of Milano-Bicocca, 2004, 50–59.
20. G. Thierrin, Convex languages. In *Proc. IRIA Symp. North Holland* 1972, 481–492.