Bounded Hairpin Completion

Masami Ito[1], Peter Leupold[1⋆], and Victor Mitrana[2⋆⋆]

[1] Department of Mathematics, Faculty of Science
Kyoto Sangyo University, Department of Mathematics
Kyoto 603-8555, Japan
ito@cc.kyoto-su.ac.jp, leupold@cc.kyoto-su.ac.jp

[2] University of Bucharest, Faculty of Mathematics and Computer Science
Str. Academiei 14, 010014, Bucharest, Romania
and
Department of Information Systems and Computation
Technical University of Valencia,
Camino de Vera s/n. 46022 Valencia, Spain
mitrana@fmi.unibuc.ro

Abstract. We consider a restricted variant of the hairpin completion called bounded hairpin completion. The hairpin completion is a formal operation inspired from biochemistry. Applied to a word encoding a single stranded molecule x such that either a suffix or a prefix of x is complementary to a subword of x, hairpin completion produces a new word z, which is a prolongation of x to the right or to the left by annealing. The restriction considered here concerns the length of all prefixes and suffixes that are added to the current word by hairpin completion. They cannot be longer than a given constant. Closure properties of some classes of formal languages under the non-iterated and iterated bounded hairpin completion are investigated. We also define the inverse operation, namely bounded hairpin reduction, and consider the set of all primitive bounded hairpin roots of a regular language.

1 Introduction

This paper is a continuation of a series of works started with [4] (based on some ideas from [3]), where a new formal operation on words inspired by the DNA manipulation called hairpin completion was introduced. That initial work has been followed up by a several related papers ([11–14]), where both the hairpin completion as well as its inverse operation, namely the hairpin reduction, were further investigated.

Several problems remained unsolved in these papers. This is the mathematical motivation for the work presented here. By considering a weaker variant of

⋆ This work was done, while the author was funded as a post-doctoral fellow by the Japanese Society for the Promotion of Science under number P07810.

⋆⋆ Work supported by the Grant-in-Aid No. 19-07810 by Japan Society for the Promotion of Science

the hairpin completion operation, called here the bounded hairpin completion, we hope to be able to solve some of the problems in this new setting that remained unsolved in the aforementioned papers. Another motivation is a practical one, closely related to the biochemical reality that inspired the definition of this operation. It seems more practical to consider that the prefix/suffix added by the hairpin completion cannot be arbitrarily long. In a laboratory every step of a computation will have to make do with a finite amount of resources and finite time; thus the length of the added string would be bounded by both the amount of additional nucleic acids in the test tube and the time given for one step of computation.

We briefly highlight some of the biological background that inspired the definition of the *Watson-Crick superposition* in [3]. The starting point is the structure of the DNA molecule. It consists of a double strand, each DNA single strand being composed by nucleotides which differ from each other in their bases: A (adenine), G (guanine), C (cytosine), and T (thymine). The two strands which form the DNA molecule are kept together by relatively weak hydrogen bonds between the bases: A always bonds with T and C with G. This phenomenon is usually referred to as *Watson-Crick complementarity.* The formation of these hydrogen bonds between complementary single DNA strands is called *annealing.*

A third essential feature from biochemistry is the PCR (Polymerase Chain Reaction). From two complementary, annealed strands, where one is shorter than the other, it produces a complete double stranded DNA molecule as follows: enzymes called polymerases add the missing bases (if they are available in the environment) to the shorter strand called *primer* and thus turn it into a complete complement of the longer one called *template.*

We now informally explain the superposition operation and how it can be related to the aforementioned biochemical concepts. Let us consider the following hypothetical biological situation: two single stranded DNA molecules x and y are given such that a suffix of x is Watson-Crick complementary to a prefix of y or a prefix of x is Watson-Crick complementary to a suffix of y, or x is Watson-Crick complementary to a subword of y. Then x and y get annealed in a DNA molecule with a double stranded part by complementary base pairing and then a complete double stranded molecule is formed by DNA polymerases. The mathematical expression of this hypothetical situation defines the superposition operation. Assume that we have an alphabet and a complementary relation on its letters. For two words x and y over this alphabet, if a suffix of x is complementary to a prefix of y or a prefix of x is complementary to a suffix of y, or x is complementary to a subword of y, then x and y bond together by complementary letter pairing and then a complete double stranded word is formed by the prolongation of x and y. Now the word obtained by the prolongation of x is considered to be the result of the superposition applied to x and y. Clearly, this is just a mathematical operation that resembles a biological reality considered here in an idealized way.

On the other hand, it is known that a single stranded DNA molecule might produce a hairpin structure, a phenomenon based on the first two biological principles mentioned above. Here one part of the strand bonds to another part of the same strand. In many DNA-based algorithms, these DNA molecules often cannot be used in the subsequent steps of the computation. Therefore it has been the subject of a series of studies to find encodings that will avoid the formation of hairpins, see e.g. [5–7] or [10] and subsequent work for investigations in the context of Formal Languages. On the other hand, those molecules which may form a hairpin structure have been used as the basic feature of a new computational model reported in [18], where an instance of the 3-SAT problem has been solved by a DNA-algorithm whose second phase is mainly based on the elimination of hairpin structured molecules.

We now consider again a hypothetical biochemical situation: we are given one single stranded DNA molecule z such that either a prefix or a suffix of z is Watson-Crick complementary to a subword of z. Then the prefix or suffix of z and the corresponding subword of z get annealed by complementary base pairing and then z is lengthened by DNA polymerases up to a complete hairpin structure. The mathematical expression of this hypothetical situation defines the hairpin completion operation. By this formal operation one can generate a set of words, starting from a single word. This operation is considered in [4] as an abstract operation on formal languages. Some algorithmic problems regarding the hairpin completion are investigated in [11]. In [12] the inverse operation to the hairpin completion, namely the hairpin reduction, is introduced and one compares some properties of the two operations. This comparison is continued in [13], where a mildly context-sensitive class of languages is obtained as the homomorphic image of the hairpin completion of linear context-free languages. This is, to our best knowledge, the first class of mildly context-sensitive languages obtained in a way that does not involve grammars or acceptors.

In the aforementioned papers, no restriction is imposed on the length of the prefix or suffix added by the hairpin completion. This fact seems rather unrealistic though this operation is a purely mathematical one and the biological reality is just a source of inspiration. On the other hand, several natural problems regarding the hairpin completion remained unsolved in the papers mentioned above. A usual step towards solving them might be to consider a bit less general setting and try to solve the problems in this new settings. Therefore, we consider here a restricted variant of the hairpin completion, called *bounded hairpin completion*. This variant assumes that the length of the prefix and suffix added by the hairpin completion is bounded by a constant.

2 Basic definitions

We assume the reader to be familiar with the fundamental concepts of formal language theory and automata theory, particularly the notions of grammar and finite automaton [16] and basics from the theory of abstract families of languages [19].

An alphabet is always a finite set of letters. For a finite set A we denote by $card(A)$ the cardinality of A. The set of all words over an alphabet V is denoted by V^*. The empty word is written λ; moreover, $V^+ = V^* \setminus \{\lambda\}$. Two languages are considered to be equal if they contain the same words with the possible exception of the empty word.

A concept from the theory of abstract families of languages that we will refer to is that of a *trio*. This is is a non-empty family of languages closed under non-erasing morphisms, inverse morphisms and intersection with regular languages. A trio is *full* if it is closed under arbitrary morphisms.

Given a word w over an alphabet V, we denote by $|w|$ its length, while $|w|_a$ denotes the number of occurrences of the letter a in w. If $w = xyz$ for some $x, y, z \in V^*$, then x, y, z are called prefix, subword, suffix, respectively, of w. For a word w, $w[i..j]$ denotes the subword of w starting at position i and ending at position j, $1 \leq i \leq j \leq |w|$. If $i = j$, then $w[i..j]$ is the i-th letter of w which is simply denoted by $w[i]$.

Let Ω be a "superalphabet", that is an infinite set such that any alphabet considered in this paper is a subset of Ω. In other words, Ω is the *universe* of the alphabets in this paper, i.e., all words and languages are over alphabets that are subsets of Ω. An *involution* over a set S is a bijective mapping $\sigma : S \longrightarrow S$ such that $\sigma = \sigma^{-1}$. Any involution σ on Ω such that $\sigma(a) \neq a$ for all $a \in \Omega$ is said to be a *Watson-Crick involution*. Despite the fact that this is nothing more than a fixed point-free involution, we prefer this terminology since the hairpin completion defined later is inspired by the DNA lengthening by polymerases, where the Watson-Crick complementarity plays an important role. Let $\bar{\cdot}$ be a Watson-Crick involution fixed for the rest of the paper. The Watson-Crick involution is extended to a morphism from Ω to Ω^* in the usual way. We say that the letters a and $\overline{a}$ are complementary to each other. For an alphabet V, we set $\overline{V} = \{\overline{a} \mid a \in V\}$. Note that V and $\overline{V}$ can intersect and they can be, but need not be, equal. Recall that the DNA alphabet consists of four letters, $V_{DNA} = \{A, C, G, T\}$, which are abbreviations for the four nucleotides and we may set $\overline{A} = T$, $\overline{C} = G$.

We denote by $(\cdot)^R$ the mapping defined by $^R : V^* \longrightarrow V^*$, $(a_1a_2 \ldots a_n)^R = a_n \ldots a_2a_1$. Note that ^{R} is an involution and an *anti-morphism* ($(xy)^R = y^Rx^R$ for all $x, y \in V^*$). Note also that the two mappings $\bar{\cdot}$ and $\cdot^R$ commute, namely, for any word x, $(\overline{x})^R = \overline{x^R}$ holds.

The reader is referred to [4] or any of the subsequent papers [11–14] for the definition of the (unbounded) *k-hairpin completion*; it is essentially the same as for the bounded variant defined below, only without the bound $|\gamma| \leq p$. Thus the prefix or suffix added by hairpin completion can be arbitrarily long. By the mathematical and biological reasons mentioned in the introductory part, in this work we are interested in a restricted variant of this operation that allows only prefixes and suffixes of a length bounded by a constant to be added. Formally, if V is an alphabet, then for any $w \in V^+$ we define the p-bounded k-hairpin completion of w, denoted by $pHC_k(w)$, for some $k, p \geq 1$, as follows:

$$\begin{aligned}
pHC \curvearrowleft_k (w) &= \{\overline{\gamma^R}w | w = \alpha\beta\overline{\alpha^R}\gamma, |\alpha| = k,\ \alpha, \beta \in V^+, \gamma \in V^*, |\gamma| \leq p\} \\
pHC \curvearrowright_k (w) &= \{w\overline{\gamma^R} | w = \gamma\alpha\beta\overline{\alpha^R}, |\alpha| = k,\ \alpha, \beta \in V^+, \gamma \in V^*, |\gamma| \leq p\} \\
pHC_k(w) &= HC \curvearrowleft_k (w) \cup HC \curvearrowright_k (w).
\end{aligned}$$

This operation is schematically illustrated in Figure 1.

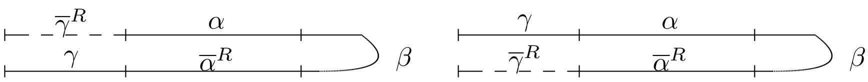

Figure 1: Bounded hairpin completion

The *p-bounded hairpin completion* of w is defined by

$$pHC(w) \;=\; \bigcup_{k\geq 1} pHC_k(w).$$

As above, the p-bounded hairpin completion operation is naturally extended to languages by

$$pHC_k(L) \;=\; \bigcup_{w\in L} pHC_k(w) \qquad pHC(L) \;=\; \bigcup_{w\in L} pHC(w).$$

The iterated version of the p-bounded hairpin completion is defined in a similar way to the unbounded case, namely:

$$\begin{array}{ll}
pHC_k^0(w) = \{w\}, & pHC^0(w) = \{w\}, \\
pHC_k^{n+1}(w) = pHC_k(pHC_k^n(w)), & pHC^{n+1}(w) = pHC(pHC^n(w)), \\
pHC_k^*(w) = \bigcup_{n\geq 0} pHC_k^n(w), & pHC^*(w) = \bigcup_{n\geq 0} pHC^n(w), \\
pHC_k^*(L) \;=\bigcup_{w\in L} pHC_k^*(w), & pHC^*(L) \;=\bigcup_{w\in L} pHC^*(w).
\end{array}$$

3 The non-iterated case

The case of bounded hairpin completion is rather different in comparison to the unbounded variant considered in [11–14]. As it was expected, the closure problem of any trio under bounded hairpin completion is simple: every (full) trio is closed under this operation.

Proposition 1 *Every (full) trio is closed under p-bounded k-hairpin completion for any $k, p \geq 1$.*

Proof. It is sufficient to consider a generalized sequential machine (*gsm*) that adds a suffix (prefix) of length at most p to its input provided that its prefix (suffix) satisfies the conditions from the definitions. As every trio is closed under *gsm* mappings, see [19], we are done. □

We recall that neither the class of regular languages nor that of context-free languages is closed under unbounded hairpin completion. By the previous theorem, both classes are closed under bounded hairpin completion.

On the other hand, in [11] it was proved that if the membership problem for a given language L is decidable in $\mathcal{O}(f(n))$, then the membership problem for the hairpin completion of L is decidable in $\mathcal{O}(nf(n))$ for any $k \geq 1$. Further, the factor n is not needed for the class of regular languages, but the problem of whether or not this factor is needed for other classes remained open in [11]. An easy adaption of the algorithm provided there shows that this factor is never needed in the case of bounded hairpin completion and thus membership is always decidable in $\mathcal{O}(f(n))$; presenting the adapted algorithm would exceed the scope of this work, though.

4 The iterated case

As in non-iterated case, the iterated bounded hairpin completion offers also a rather different picture of closure properties in comparison to the unbounded variant considered in the same papers cited above. We start with a general result.

Theorem 1 *Let $p, k \geq 1$ and $\mathcal{F}$ be a (full) trio closed under substitution. Then $\mathcal{F}$ is closed under iterated p-bounded k-hairpin completion if and only if $pHC_k^*(w) \in \mathcal{F}$ for any word w.*

Proof. The "only if" part is obvious as any trio contains all singleton languages.

For the "if" part, let $L \in \mathcal{F}$ be a language over the alphabet V. We write $L = L_1 \cup L_2$, where

$$L_1 = \{x \in L \mid |x| < 2(k+p)+1\},$$
$$L_2 = \{x \in L \mid |x| \geq 2(k+p)+1\}.$$

Clearly, $pHC_k^*(L) = pHC_k^*(L_1) \cup pHC_k^*(L_2)$. As any trio contains all finite languages, it follows that any trio closed under substitution is closed under union. Therefore, as L_1 is a finite language, we conclude that $pHC_k^*(L_1) \in \mathcal{F}$. Consequently, it remains to show that $pHC_k^*(L_2) \in \mathcal{F}$.

Let $\alpha, \beta \in V^+$ be two arbitrary words with $|\alpha| = |\beta| = k + p$. We define $L_2(\alpha, \beta) = L_2 \cap \{\alpha\}V^+\{\beta\}$. We have that

$$L_2 = \bigcup_{|\alpha|=|\beta|=k+p} L_2(\alpha, \beta) \quad \text{and} \quad pHC_k^*(L_2) = \bigcup_{|\alpha|=|\beta|=k+p} pHC_k^*(L_2(\alpha, \beta)).$$

On the other hand, it is plain that $pHC_k^*(L_2(\alpha, \beta)) = s(pHC_k^*(\alpha X \beta))$, where X is a new symbol not in V and s is a substitution $s : (V \cup \{X\})^* \longrightarrow 2^{V^*}$ defined by $s(a) = \{a\}$ for all $a \in V$ and $s(X) = \{w \in V^+ \mid \alpha w \beta \in L_2(\alpha, \beta)\}$. The language $\{w \in V^+ \mid \alpha w \beta \in L_2(\alpha, \beta)\}$ is in $\mathcal{F}$ (even $\mathcal{F}$ is not full) as it is the image of a language from $\mathcal{F}$, namely $L_2(\alpha, \beta)$, through a *gsm* mapping that deletes both the prefix and suffix of length $k + p$ of the input word. By the closure properties of $\mathcal{F}$, it follows that $pHC_k^*(L_2(\alpha, \beta))$ is in $\mathcal{F}$ for any α, β as above, which completes the proof. □

We recall that none of the families of regular, linear context-free, and context-free languages is closed under iterated unbounded hairpin completion. Here the bounded hairpin completion is much more tractable.

Corollary 1 *The family of context-free languages is closed under iterated p-bounded k-hairpin completion for any $k, p \geq 1$.*

Proof. By the previous result, it suffices to prove that $pHC_k(w)$ is context-free for any word w. Given $w \in V^+$, we construct the arbitrary grammar $G = (\{S, X\}, V \cup \{\#\}, S, P)$, where the set of productions P contains the following rules:

$$\begin{aligned} P = \{S \to yXz \mid w = zy\} &\cup \{\overline{z}^R yXz \to \overline{z}^R yX\overline{y}^R z \mid 1 \leq |y| \leq p, |z| = k\} \\ &\cup \{\overline{z}^R Xyz \to \overline{z^R y^R} Xyz \mid 1 \leq |y| \leq p, |z| = k\} \cup \{X \to \#\}. \end{aligned}$$

By a result of Baker (see [1]), the language generated by G is context-free. Further we have that $pHC_k^*(w) = h(cp(L(G)) \cap V^+\{\#\})$. Here cp maps every word in the set of all its circular permutations and every language in the set of all circular permutations of its words, while h is a morphism that erases $\#$ and leaves unchanged all letters of V. As the class of context-free languages is closed under circular permutation [17], we infer that $pHC_k^*(w)$ is context-free. □

The above argument does not work for the class of linear context-free languages as this class is known not to be closed under circular permutation. However, also this family is closed under iterated bounded hairpin completion.

Theorem 2 *The family of linear context-free languages is closed under iterated p-bounded k-hairpin completion for any $k, p \geq 1$.*

Proof. Let L be a language generated by the linear grammar $G = (N, T, S, P)$. We construct the linear grammar $G' = (N', T, S', P')$, where

$$\begin{aligned} N' = N \cup \{S'\} &\cup \{[\alpha, \beta] \mid \alpha, \beta \in T^*, 0 \leq |\alpha|, |\beta| \leq k + p\} \\ &\cup \{[\alpha, A, \beta] \mid \alpha, \beta \in T^*, 0 \leq |\alpha|, |\beta| \leq k + p, A \in N\}, \end{aligned}$$

and the set of productions P' is defined by (in the definition of every set $\alpha, \beta \in T^*, 0 \leq |\alpha|, |\beta| \leq k + p, A \in N$ holds):

$$\begin{aligned} P' = P &\cup \{S' \to S\} \cup \{S' \to [\alpha, \beta] \mid \alpha, \beta \in T^*, 0 \leq |\alpha|, |\beta| \leq k + p\} \\ &\cup \{[\alpha, \beta] \to [\alpha', \beta']\overline{y}^R \mid \alpha = \alpha' = yvw, \beta = u\overline{v^R y^R}, |v| = k, |y| \leq p, \\ &\quad \beta' = xu\overline{v}^R, x \in T^*, |\beta'| \leq k + p\} \\ &\cup \{[\alpha, \beta] \to y[\alpha', \beta']\overline{y}^R \mid \alpha = yvw, \beta = \beta' = u\overline{v^R y^R}, |v| = k, |y| \leq p, \\ &\quad \alpha' = vwx, x \in T^*, |\alpha'| \leq k + p\} \\ &\cup \{[\alpha, \beta] \to [\alpha, S, \beta] \mid \alpha, \beta \in T^*, 0 \leq |\alpha|, |\beta| \leq k + p\} \\ &\cup \{[\alpha, A, \beta] \to x[\alpha', B, \beta']y \mid A \to xBy \in P, \alpha = x\alpha', \beta = \beta' y, \alpha', \beta' \in T^*\} \\ &\cup \{[\alpha, A, \beta] \to \alpha x[\lambda, B, \beta']y \mid A \to \alpha xBy \in P, \beta = \beta' y, \beta' \in T^*\} \\ &\cup \{[\alpha, A, \beta] \to x[\alpha', B, \lambda]y\beta \mid A \to xBy\beta \in P, \alpha = x\alpha', \alpha' \in T^*\} \\ &\cup \{[\lambda, A, \lambda] \to A \mid A \in N\}. \end{aligned}$$

It is rather easy to note that we have the derivation

$$S' \Longrightarrow^* x[\alpha, \beta]y \Longrightarrow x[\alpha, S, \beta]y \Longrightarrow^* x\alpha w\beta y$$

in G' if and only if $S \Longrightarrow^* \alpha w\beta$ in G and $x\alpha w\beta y \in pHC_k^*(\alpha w\beta)$. This concludes the proof. $\square$

The problem of whether or not the iterated unbounded hairpin completion of a word is context-free is open. By the previous result, it follows that the iterated bounded hairpin completion of a word is always linear context-free. We do not know whether this language is always regular. More generally, the status of the closure under iterated bounded hairpin completion of the class of regular languages remains unsettled.

We finish this section with another general result.

Theorem 3 *Every trio closed under circular permutation and iterated finite substitution is closed under iterated bounded hairpin completion.*

Proof. We take two positive integers $k, p \geq 1$. Let $\mathcal{F}$ be a family of languages with the above properties and $L \subseteq V^*$ be a language in $\mathcal{F}$. Let L_1 be the circular permutation of $L\{\#\}$, where $\#$ is a new symbol not in V. Clearly, L_1 still lies in $\mathcal{F}$. We consider the alphabet $W = \{[x\#y] \mid x, y \in V^*, 0 \leq |x|, |y| \leq p + k\}$ and define the morphism $h : (W \cup V)^* \longrightarrow (V \cup \{\#\}^*$ by $h([x\#y]) = x\#y$, for any $[x\#y] \in W$, and $h(a) = a$, for any $a \in V$. We now consider the language $L_2 \in \mathcal{F}$ given by $L_2 = h^{-1}(L_1)$. By the closure properties of $\mathcal{F}$, the language $L_3 = s^*(L_2)$ is in $\mathcal{F}$, where s is the finite substitution $s : (W \cup V)^* \longrightarrow 2^{(W \cup V)^*}$ defined by $s(a) = \{a\}, a \in V$, and $s([x\#y]) = \{x\#y\} \cup R$ with

$$
\begin{aligned}
R = \{&[x\#\overline{u}^R y] \mid x = vzu, y = \overline{z}^R w, u, v, z, w \in V^*, |z| = k,\\
&\quad |\overline{u}^R y| \leq p + k, |u| \leq p\} \cup\\
&\{[x\#\overline{u}^R y']y'' \mid x = vzu, y = \overline{z}^R w = y'y'', u, v, z, w, y', y'' \in V^*, |z| = k,\\
&\quad |\overline{u}^R y'| = p + k, |u| \leq p\}\} \cup\\
&\{[x\overline{u}^R\#y] \mid x = w\overline{z}^R, y = uzv, u, v, z, w \in V^*, |z| = k,\\
&\quad |x\overline{u}^R| \leq p + k, |u| \leq p\}\} \cup\\
&\{x''[x'\overline{u}^R\#y] \mid x = w\overline{z}^R = x''x', y = uzv, u, v, z, w, x', x'' \in V^*, |z| = k,\\
&\quad |x'\overline{u}^R| = p + k, |u| \leq p\}\}.
\end{aligned}
$$

Finally, let L_4 be the circular permutation of $h(L_3)$. Then we obtain that $pHC_k^*(L) = g(L_4 \cap V^*\{\#\})$, where g is a morphism that removes $\#$ and leaves unchanged all symbols from V. $\square$

5 An inverse operation: the bounded hairpin reduction

We now define the inverse operation of the bounded hairpin completion, namely the bounded hairpin reduction in a similar way to [13], where the unbounded hairpin reduction is introduced. Let V be an alphabet, for any $w \in V^+$ we define the *p-bounded k-hairpin reduction* of w, denoted by $pHR_k(w)$, for some $k, p \geq 1$, as follows:

$$pHR \circlearrowleft_k (w) = \{\alpha\beta\overline{\alpha^R\gamma^R} | w = \gamma\alpha\beta\overline{\alpha^R\gamma^R}, |\alpha| = k,\ \alpha, \beta, \gamma \in V^+, 1 \leq |\gamma| \leq p\},$$

$pHR \circlearrowleft_k (w) = \{\gamma\alpha\beta\overline{\alpha^R} | w = \gamma\alpha\beta\overline{\alpha^R\gamma^R}, |\alpha| = k,\ \alpha, \beta, \gamma \in V^+, 1 \leq |\gamma| \leq p\}.$

$pHR_k(w) = pHR \circlearrowleft_k (w) \cup pHR \circlearrowright (w).$

The *p-bounded hairpin reduction* of w is defined by

$$pHR(w) = \bigcup_{k \geq 1} pHR_k(w).$$

The bounded hairpin reduction is naturally extended to languages by

$$pHR_k(L) = \bigcup_{w \in L} pHR_k(w) \qquad pHR(L) = \bigcup_{w \in L} pHR(w).$$

The iterated bounded hairpin reduction is defined analogously to the iterated bounded hairpin completion.

We recall that the problem of whether or not the iterated unbounded hairpin reduction of a regular language is recursively decidable is left open in [13]. The same problem for the iterated bounded hairpin reduction is now completely solved by the next more general result. Before stating the result, we need to recall a few notions about string-rewriting systems. To this aim, we follow the standard notations for string rewriting as in [2]. A string-rewriting system (SRS) over an alphabet V is a finite relation $R \subset V^* \times V^*$, and the rewrite relation induced by R is denoted by $\longrightarrow_R$. That is, we write $x \longrightarrow_R y$ if $x = uvw, y = uzw$, for some $u, v, z, w \in V^*$, and $(v, z) \in R$. As usual every pair $(v, z) \in R$ is referred to as a rule $v \to z$. The reflexive and transitive closure of $\longrightarrow_R$ is denoted by $\longrightarrow_R^*$. We use $R^*(L)$ for the closure of the language L under the string-rewriting system R. Formally, $R^*(L) = \{w \mid x \longrightarrow_R^* w,$ for some $x \in L\}$. A rule $v \to z$ is said to be *monadic* if it is length-reducing ($|v| > |z|$) and $|z| \leq 1$. A SRS is called monadic if all its rules are monadic. A class of languages $\mathcal{F}$ is closed under monadic SRS if for any language $L \in \mathcal{F}$ over some alphabet V and any monadic SRS R over V, $R^*(L) \in \mathcal{F}$ holds.

Theorem 4 *Every trio closed under circular permutation and monadic string-rewriting systems is closed under iterated bounded hairpin reduction.*

Proof. Let $\mathcal{F}$ be a trio and k, p be two positive integers. The central idea of the proof is as follows. We permute every word of a language in $\mathcal{F}$ in a circular way. Then the last and first letters are next to each other. Thus the hairpin reduction becomes a local operation and can be simulated by monadic string-rewriting rules. By our hypothesis, these are known to preserve the membership in $\mathcal{F}$.

To start with, we attach a new symbol X to the end of every word of a given $L \in \mathcal{F}$, $L \subseteq V^*$. Then we obtain the language L' by doing a circular permutation to all words in $L\{X\}$. Note that X marks the end and beginning of the original words. On this language we apply a gsm-mapping g that introduces redundancy by adding to every letter information about its neighboring letters in the following way:

1. The letter containing the X contains also the $k+p$ letters to the left and to the right of X in order.

2. Every letter left of X contains the letter originally at that position and the $k+p$ letters left of it in order.
3. Every letter right of X contains the letter originally at that position and the $k+p$ letters right of it in order.

At the word's end and its beginning, where there are not enough letters to fill the symbols, some special symbol signifying a space is placed inside the compound symbols.

Now we can simulate a step of p-bounded k-hairpin reduction by a string-rewriting rule with a right-hand side of length one, i.e. a monadic one. A straightforward approach would be to use rules of the form $u\overline{v}^R X v\overline{u}^R \to uXv\overline{u}^R$. But we see that u and $Xv\overline{u}^R$ are basically not changed, they only form a context whose presence is necessary. Through our redundant representation of the word, their presence can be checked by looking only at the corresponding image of X under g. Further, since the symbols of the image of u under g contain only information about symbols to their left, they do not need to be updated after the deletion of $\overline{v}^R$ to preserve the properties 1 to 3. The same is true for $v\overline{u}^R$. Only in the symbol corresponding to X some updating needs to be done and thus it is the one that is actually rewritten. So the string rewriting rules are

$$g_{\mathsf{left}}(z_0 z_1 u\overline{v})[1\ldots|v|][z_1 u\overline{v}Xv\overline{u}z_2] \to [z_0 z_1 uXv\overline{u}z_2],$$

where g_{left} does the part of g described by property 2, and where $z_0, z_1 \in V^*$, $u, v \in V^+$, $|u| = k$, $|v| \le p$, $|z_0 z_1 u| = p+k$. Analogously, rules that delete symbols to the right of X are defined. Let R be the string-rewriting system consisting of all such rules. It is immediate that $w' \in pHR_k(w) \iff g(cp(wX)) \to_R g(cp(w'X))$ and by induction $w' \in pHR_k^*(w) \iff g(cp(wX)) \to_R^* g(cp(w'X))$.

Therefore, at this point we have all circular permutations of words that can be reached by p-bounded k-hairpin reduction from words in L coded under g. To obtain our target language we first undo the coding of g by the gsm-mapping g' that projects all letters to the left of X to their last component, all letters to the right of X to their first component, and the symbol containing X to just X. This mapping is letter-to-letter, the gsm only needs to remember in its state whether is has already passes over the symbol containing X. Of the result of this we take again the circular permutation.

Now we filter out the words that have X at the last position and therefore are back in the original order of L and delete X. By the closure properties of $\mathcal{F}$, the result of this process lies in $\mathcal{F}$, which completes the proof. $\square$

As monadic SRSs are known to preserve regularity (see [8]) we immediately infer that

Theorem 5 *The class of regular languages is closed under iterated bounded hairpin reduction.*

In [12] one considers another concept that seems attractive to us, namely the *primitive hairpin root* of a word and of a language. Given a word $x \in V^*$ and a positive integer k, the word y is said to be the primitive k-hairpin root of x if the following two conditions are satisfied:

$$(i)\ \ y \in HR_k^*(x)(\text{or, equivalent, } x \in HC_k^*(y)),$$
$$(ii)\ \ HR_k(y) = \emptyset.$$

Here $HR_k^*(z)$ delivers the iterated unbounded hairpin reduction of the word z. In other words, y can be obtained from x by iterated k-hairpin reduction (maybe in zero steps) and y cannot be further reduced by hairpin reduction. The primitive bounded hairpin root is defined analogously. Clearly, a word may have more than one primitive bounded hairpin root; the set of all primitive p-bounded k-hairpin roots of a word x is denoted by $pH_k root(x)$. Naturally, the primitive p-bounded k-hairpin root of a language L is defined by $pH_k root(L) = \bigcup_{x \in L} pH_k root(x)$.

Clearly, to see whether a word is reducible, one has to look only at the first and last $k + p$ symbols. By Theorem 5 we have:

Theorem 6 *$pH_k root(L)$ is regular for any regular language L and any $p, k \geq 1$.*

Proof. For the regular language $L' \subseteq V^*$ obtained in the proof of Theorem 5 it suffices to consider the language

$$\{w \in L \mid |w| \leq 2k+2\} \cup (pHR_k^*(L) \cap \{\alpha x \beta \mid |\alpha| = |\beta| = k+1, \alpha \neq \overline{\beta}^R, x \in V^+\})$$

which is regular and equals $pH_k root(L)$. □

6 Final remarks

We have considered a restricted version of the hairpin completion operation by imposing that the prefix or suffix added by the hairpin completion are bounded by a constant. In some sense, this is the lower extreme case the upper extreme being the unbounded case that might be viewed as a linearly bounded variant. We consider that bounded variants by other sublinear mappings would be of theoretical interest.

Last but not least we would like to mention that hairpin completion and reduction resemble some language generating mechanisms considered in the literature like external contextual grammars with choice [15] or dipolar contextual deletion [9], respectively.

References

1. Baker, B.S: Context-sensitive grammars generating context-free languages. In Automata, Languages and Programming ICALP 1972. North-Holland, Amsterdam (1972) 501–506.
2. Book, R., Otto, F.: String-Rewriting Systems. Springer-Verlag (1993).
3. Bottoni, P., Labella, A., Manca, V., Mitrana, V.: Superposition based on Watson-Crick-like complementarity. Theory of Computing Systems **39** (2006) 503–524.

4. Cheptea, D., Martín-Vide, C., Mitrana, V.: A new operation on words suggested by DNA biochemistry: hairpin completion. In Transgressive Computing (2006) 216–228.
5. Deaton, R., Murphy, R., Garzon, M., Franceschetti, D.R., Stevens, S.E.: Good encodings for DNA-based solutions to combinatorial problems. In Proc. of DNA-based computers II. DIMACS Series **44** (1998) 247–258.
6. Garzon, M., Deaton, R., Neathery, P., Murphy, R.C., Franceschetti, D.R., Stevens, S.E.: On the encoding problem for DNA computing. In The Third DIMACS Workshop on DNA-Based Computing. Univ. of Pennsylvania (1997) 230–237.
7. Garzon, M., Deaton, R., Nino, L.F., Stevens, S.E., Wittner, M.: Genome encoding for DNA computing. In Proc. Third Genetic Programming Conference. Madison, MI (1998) 684–690.
8. Hofbauer, D., Waldmann, J.: Deleting string-rewriting systems preserve regularity. Theoretical Computer Science **327** (2004) 301-317.
9. Kari, L., Thierrin, G.: Contextual insertions/deletions and computability. Information and Computation **131** (1996) 47–61.
10. Kari, L., Konstantinidis, S., Sosík, P., Thierrin, G.: On hairpin-free words and languages. In Developments in Language Theory 2005. Vol. 3572 of LNCS. Springer-Verlag (2005) 296–307.
11. Manea, F., Martín-Vide, C., Mitrana, V.: On some algorithmic problems regarding the hairpin completion. Discrete Applied Mathematics (in press), doi:10.1016/j.dam.2007.09.022.
12. Manea, F., Mitrana, V.: Hairpin completion versus hairpin reduction. In Computation in Europe CiE 2007. Vol. 4497 of LNCS. Springer-Verlag (2007) 532-541.
13. Manea, F., Mitrana, V., Yokomori, T.: Two complementary operations inspired by the DNA hairpin formation: completion and reduction. Theoretical Computer Science, (in press), doi:10.1016/j.tcs.2008.09.049.
14. Manea, F., Mitrana, V., Yokomori, T.: Some remarks on the hairpin completion. In Proc. 12th International Conference on Automata and Formal Languages (2008) 302–313.
15. Marcus, S.: Contextual grammars. Rev. Roum. Math. Pures Appl. **14** (1969) 1525–1534.
16. Rozenberg, G., Salomaa, A. (Eds.): Handbook of Formal Languages. Springer-Verlag (1997).
17. Ruohonen, K.: On circular words and $(\omega^* + \omega)$-powers of words. Elektr. Inform. und Kybern. E.I.K. **13** (1977) 3–12.
18. Sakamoto, K., Gouzu, H., Komiya, K., Kiga, D., Yokoyama, S., Yokomori, T., Hagiya, M.: Molecular computation by DNA hairpin formation. Science **288** (2000) 1223–1226.
19. Salomaa, A.: Formal Languages. Academic Press (1973).