On the Hairpin Completion of Regular Languages

Volker Diekert[1], Steffen Kopecki[1], and Victor Mitrana[2]

[1] Universität Stuttgart, FMI, Germany
[2] Faculty of Mathematics, University of Bucharest, Romania
and
Department of Information Systems and Computation
Technical University of Valencia, Spain
diekert@fmi.uni-stuttgart.de, steffen.kopecki@web.de,
mitrana@fmi.unibuc.ro

Abstract. The hairpin completion is a natural operation of formal languages which has been inspired by molecular phenomena in biology and by DNA-computing. The hairpin completion of a regular language is linear context-free and we consider the problem to decide whether the hairpin completion remains regular. This problem has been open since the first formal definition of the operation.

In this paper we present a positive solution to this problem. Our solution yields more than decidability because we present a polynomial time procedure. The degree of the polynomial is however unexpectedly high, since in our approach it is more than n^{14}. Nevertheless, the polynomial time result is surprising, because even if the hairpin completion $\mathcal{H}$ of a regular language L is regular, there can be an exponential gap between the size of a minimal DFA for L and the size of a smallest NFA for $\mathcal{H}$.

1 Introduction

The origin of this paper is motivated by biological and DNA-computing. But although our motivation is based on biological phenomena, the present paper is more about an interesting decidability result on regular languages. Let us explain the background first and the connection to Formal Language Theory later.

Single-stranded DNA (ssDNA) are composed by nucleotides which differ from each other by their bases: A (adenine), G (guanine), C (cytosine), and T (thymine). Therefore each ssDNA may be viewed as a finite string over the four-letter alphabet $\{A, C, G, T\}$. Two single strands can bind to each other forming the secondary structure of DNA if they are pairwise Watson-Crick complementary: A is complementary to T, and C to G. The binding of two strands is also called *annealing*.

An intramolecular base pairing, known as *hairpin*, is a pattern that can occur in single-stranded DNA and, more commonly, in RNA. Hairpin or hairpin-free structures have numerous applications to DNA computing and molecular genetics. In many DNA-based algorithms, these DNA molecules cannot be used

M. Leucker and C. Morgan (Eds.): ICTAC 2009, LNCS 5684, pp. 170–184, 2009.

in the subsequent computations. Therefore, it is important to design methods for constructing sets of DNA sequences which are unlikely to lead to "bad" hybridizations. This problem was considered in a series of papers, see e.g. [2,3,4,7,8].

In [1,12] a new formal operation on words is introduced, namely the *hairpin completion*. It consists of three biological principles. Besides the Watson-Crick complementarity and annealing the third biological phenomenon is that of *lengthening DNA by polymerases*. In our case the phenomenon produces a complete molecule as follows: one starts with hairpins which are here single strands such that for each of them one end is annealed to a part of itself by Watson-Crick complementarity; and a *polymerization buffer* with many copies of the four nucleotides. Then polymerases will concatenate to the hairpin by complementing the template.

What happens in this situation is, informally, best explained in Fig. 1. In that picture as in the rest of the paper we mean by putting a *bar* on a word (like $\overline{\alpha}$) to read it from right-to-left in addition to replacing a by $\overline{a}$ for letters.

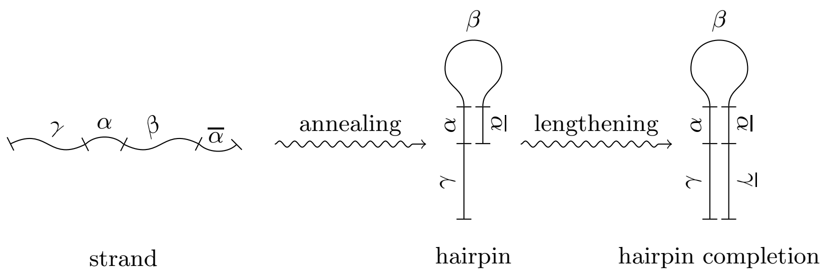

Fig. 1. Hairpin completion of a strand

This is a good starting point to translate the biologically inspired motivation to a purely abstract formalism. On that level, we have just a finite alphabet Σ together with an *involution*. This is a bijection $\overline{\ } : \Sigma \to \Sigma$ such that $\overline{\overline{a}} = a$ for all $a \in \Sigma$. In the concrete situation above $\Sigma = \{A, C, G, T\}$ and $\overline{A} = T$ and $\overline{C} = G$. We extend the involution to words $a_1 \cdots a_n$ by $\overline{a_1 \cdots a_n} = \overline{a_n} \cdots \overline{a_1}$. (Just like taking inverses in groups.)

We start with a (formal) language $L \subseteq \Sigma^*$ (the set of strands). Then hairpin completion can arise in one-sided way. The *right-sided hairpin completion* of L is formally defined by the set of words $\gamma\alpha\beta\overline{\alpha}\,\overline{\gamma}$ with $\gamma\alpha\beta\overline{\alpha} \in L$ being the strand and $\gamma\alpha\beta\overline{\alpha}\,\overline{\gamma}$ being the completion, see again Fig. 1. Still inspired by biological facts, a binding in a hairpin can be *stable*, only if α is long enough, say $|\alpha| \geq 10$. Formally we fix a (small) constant k and ask $|\alpha| \geq k$. The *left-sided hairpin completion* can be defined analogously.

Clearly, the hairpin completion of a finite language is finite. If L is regular then, sometimes the right/left-sided k-hairpin completion is regular again, sometimes it is not. But then it is a linear context-free language as the reader will immediately recognize. For example, if $L = ab^*b^kc\overline{b}^{\,k}$, then the right-sided k-hairpin completion is not regular, but linear context-free, because it is:

$$\left\{ab^mb^kc\overline{b}^{\,k}\overline{b}^{\,n}\overline{a} \;\middle|\; m \geq n\right\}.$$

This leads to a first natural decidability problem:

Problem 1. Is it decidable whether the right-sided k-hairpin completion of a regular language is regular again?

We can see directly from the hairpin picture that it is not always natural to distinguish between left and right. Therefore we consider the two-sided case, too. The *(two-sided) hairpin completion* of L is therefore defined by the set of words $\gamma\alpha\beta\overline{\alpha}\,\overline{\gamma}$ with either $\gamma\alpha\beta\overline{\alpha} \in L$ or $\alpha\beta\overline{\alpha}\,\overline{\gamma} \in L$ or both. If we simply speak about the *hairpin completion* we always mean the two-sided case. As above we see two possibilities, and, moreover, we see that the behaviors are different. Let us consider $L = ab^*b^kc\overline{b}^{\,k} \cup b^kc\overline{b}^{\,k}\overline{b}^{\,*}\overline{a}$. The right- and left-sided k-hairpin completion is still not regular, but the two-sided is.

However, if we consider $L = a^+b^kc\overline{b}^{\,k}$, then neither the right- nor the two-sided k-hairpin completion is regular. They are identical and equal to:

$$\left\{a^nb^kc\overline{b}^{\,k}\overline{a}^{\,n} \;\middle|\; n \geq 1\right\}.$$

This leads to a second natural decidability problem:

Problem 2. Is it decidable whether the k-hairpin completion of a regular language is regular again?

The initial work [1] has been followed up by several related papers [6,9,10,11,12], where both the hairpin completion as well as its inverse operation, namely the hairpin reduction, considered as formal operations on strings and languages were further investigated. But the decidability status of Problems 1 and 2 remained open. Actually, the difficulty in solving Problems 1 and 2 is perhaps not that surprising since we are immediately confronted with decidability questions on linear context-free languages. Every linear context-free language is a weak code image of an hairpin completion of some regular language. (A *weak code* is a homomorphism which is the identity on a subset of letters and maps the other letters to the empty word.) To see this let us quote a theorem from [1]:

Theorem 1. *A language is linear context-free if and only if it is the weak-code image of the hairpin completion of a regular language.*

Natural problems well-known to be undecidable for context-free languages are already undecidable for linear context-free languages, see e.g. [5] for a classical reference. In particular it is undecidable whether linear context-free languages are universal or equal to a given regular language or whether a linear context-free language is regular.

Thus, Problems 1 and 2 are problems about a subclass of linear context-free languages where no general results were known to solve them. In this paper we give positive answers to both problems. Actually, they are decidable in polynomial time (if the input size is given as the size of a DFA for L plus the size of a

DFA accepting the reversal language of L. Clearly, there might be an exponential gap between these sizes.)

The history of the solution shows several steps. First we solved Problem 1 and we realized that, retrospective, it was not difficult to find the solution, but we had no good estimation for the complexity. The solution to Problem 2 was much more difficult, and it became rather technical. The complexity was again unclear. A very rough estimation led us to something like triple exponential, but we worked in syntactic monoids and raised, whenever possible, elements to idempotent powers. So it was clear that there was room for improvement, and the intermediate results were never published.

The present solution is more ambitious. We prove a polynomial time result, which is more than expected when we started our work. What we find also quite amazing is the following: We treat natural problems about regular languages which we now know to be decidable in polynomial time. But the degree for the polynomial as we present the algorithm here might be about 20. So it is very high. With more efforts we were able to bring the degree down to 14, but this is not shown here. Such a huge time complexity is however no indication that for real life examples the problem is difficult. For most regular languages L it is probably very easy to decide whether the k-hairpin completion is regular again. Being regular is the exception and puts many constraints on L as we will see below. The formal statement of our result is in Section 3.

2 Notation

We assume the reader to be familiar with the fundamental concepts of formal language theory, context-free grammars and automata theory, see [5]. We also use *syntactic monoids*, but very little of this rich theory. What we use is the following elementary fact. If L is a regular language then there is a constant $s \in \mathbb{N}$ such that for all words x, y, z we have $xy^s z \in L$ if and only if $xy^{2s} z \in L$. Note that this implies $xy^s z \in L$ if and only if $x(y^s)^+ z \subseteq L$.

We use non-deterministic finite automata (NFA) and deterministic finite automata (DFA). Whenever convenient we use that all states are reachable and co-reachable. Thus, if g is a state then there is a path from the initial state to g and a path from g to some final state.

An *alphabet* is a finite set of *letters*. Here the alphabet is Σ. The set of words over Σ is denoted Σ^*, as usual, and the *empty word* is denoted by 1. Given a word w, we denote by $|w|$ its length. If $w = xyz$ for some $x, y, z \in \Sigma^*$, then x, y, z are called *prefix, factor, suffix*, respectively. For the prefix relation we also use the notation $x \leq w$. By a proper factor y of w we mean a factor such that $x \neq w$, but in our paper we allow $x = 1$.

As said above, Σ is equipped with an involution such that $\overline{\overline{a}} = a$ for all letters $a \in \Sigma$. The involution is extended to words by $\overline{1} = 1$ and $\overline{uv} = \overline{v}\,\overline{u}$, thus the involution reverses the order as well. Due to this law some authors call it an anti-involution, but we prefer our convention (which is also the more standard one).

If L is a language, then its *reversal language* is given by reading words right-to-left, i.e. by the set of words $a_n \cdots a_1$ where $a_1 \cdots a_n \in L$ and $a_i \in \Sigma$. Note that a DFA of minimal size for the reversal language yields also a DFA for $\overline{L} = \{\overline{w} \in \Sigma^* \mid w \in L\}$ of exactly the same size, and vice versa.

We intend to solve Problem 1 and 2 simultaneously, therefore we introduce a more general notion of hairpin completion.

Throughout the paper L and R denote two regular languages and $k > 0$ is a positive integer. We define the *hairpin completion* $\mathcal{H}(L, R, k)$ by

$$\mathcal{H}(L, R, k) = \{\gamma\alpha\beta\overline{\alpha}\,\overline{\gamma} \mid (\gamma\alpha\beta\overline{\alpha} \in L \vee \alpha\beta\overline{\alpha}\,\overline{\gamma} \in R) \wedge |\alpha| = k\}$$

Note that the definition does not change if we replace $|\alpha| = k$ by $|\alpha| \geq k$. For simplicity of the presentation we treat k as a (small) constant.

3 Main Result

Note that the right-sided k-hairpin completion is nothing but $\mathcal{H}(L, \emptyset, k)$, whereas the two-sided version appears as $\mathcal{H}(L, L, k)$. Thus, the notion $\mathcal{H}(L, R, k)$ is adopted to treat both cases simultaneously.

Problem 3. *Input:* A DFA accepting L of at most n states and a DFA accepting the reversal language of L (or for $\overline{L}$) of at most n states.

Question: Is the hairpin completion $\mathcal{H}(L, R, k)$ regular?

The purpose of this paper is to prove the following theorem.

Theorem 2. *Let Σ be a fixed alphabet and $k > 0$ be a constant. Let L and R be regular languages. Then it is decidable whether the hairpin completion $\mathcal{H}(L, R, k)$ is regular.*

As we have explained above, Problem 3 is more general than Problem 1 and 2. Obviously, for Problem 1 we do not need a DFA for the reversal language.

An NFA of minimal size accepting the hairpin completion may have exponentially more states than a DFA for L and $\overline{L}$. Thus, although we have a polynomial time decision algorithm there is no time to construct the NFA (in plain form).

Indeed let

$$L_n = \{bv\overline{a}^k ba^k \mid v \in \{a, b\}^n\}.$$

Then we have $\mathcal{H}(L_n, \emptyset, k) = \mathcal{H}(L_n, L_n, k) = \{bv\overline{a}^k ba^k \overline{v}\overline{b} \mid v \in \{a, b\}^n\}$.

Thus, the sizes of a minimal DFA accepting L_n and $\overline{L_n}$ are in $\mathcal{O}(n)$. But every NFA accepting $\mathcal{H}(L_n, \emptyset, k)$ must keep track of v and thus its size is in $\Omega(2^n)$.

The proof of Theorem 2 is quite technical and relies on some non-standard constructions for finite automata and context-free grammars.

The key idea is to use a linear grammar which produces exactly those $\gamma\alpha\beta\overline{\alpha}\,\overline{\gamma}$ where $|\gamma|$ is minimal. We show that, due to the minimality of $|\gamma|$, the context-free grammar has either a very special structure or the hairpin completion is not regular. This leads to a series of decidable conditions for the regularity of the hairpin completion which are either sufficient or necessary. The last test in this series yields the result.

3.1 An NFA for L and R

Regular languages can be specified by deterministic finite automata (DFA). A DFA is essentially a finite set Q together with a monoid action of Σ^* on the right. The action is written as a product $q \cdot u$ with the usual laws $q \cdot uv = (q \cdot u) \cdot v$ and $q \cdot 1 = q$, where $q \in \mathcal{Q}$ and $u, v \in \Sigma^*$. By 1 we denote the empty word and the neutral element in other monoids. The action is defined by a function $\mathcal{Q} \times \Sigma^* \to \mathcal{Q}$. In the following we assume that the regular language L is specified by a DFA with state set $\mathcal{Q}_L$, $q_{0,L} \in \mathcal{Q}_L$ as initial state, and $\mathcal{F}_L \subseteq \mathcal{Q}_L$ as final states. We fix $n_L = |\mathcal{Q}_L|$ to be the number of states. For R we need however a DFA reading R from right-to-left. Such an automaton is essentially equivalent to a DFA accepting the *reversal language* of R.

We start with a finite set Q_R and a *left-action* of Σ^*. For simplicity we use a product sign again, but we write it on the left: $u \cdot q$ satisfying $uv \cdot q = u \cdot (v \cdot q)$ and $1 \cdot q = q$. We choose $\mathcal{Q}_R$, $q_{0,R} \in \mathcal{Q}_R$ and $\mathcal{F}_R \subseteq \mathcal{Q}_R$ such that

$$R = \{u \in \Sigma^* \mid u \cdot q_{0,R} \in \mathcal{F}_R\}.$$

Let $n_R = |\mathcal{Q}_R|$. For the rest of the paper we fix $n = n_L + n_R$. We view n as input size for our decidability problem (stated in Theorem 2) to test whether the hairpin completion $\mathcal{H}(L, R, k)$ is regular.

What we are really interested in is the product automaton with state space

$$\mathcal{Q} = \mathcal{Q}_L \times \mathcal{Q}_R.$$

Although we started with deterministic automata, we content to read $\mathcal{Q}$ as the state space of a non-deterministic automaton which accepts L reading words from left-to-right and accepts R reading words from right-to-left. Since this construction is crucial, we make it precise: Let $P = (p_1, p_2)$, $Q = (q_1, q_2)$ be states of $\mathcal{Q}$ and $a \in \Sigma$ be a letter. We define an *arc* (P, a, Q), if $p_1 \cdot a = q_1$ and $p_2 = a \cdot q_2$. Note that P may have several outgoing arcs labeled by a because for each p_2 and each a there might be several q_2 with $p_2 = a \cdot q_2$.

Let $u \in \Sigma^*$ be a word. Then for each pair (p, q) there is a unique pair $(r, s) \in \mathcal{Q}_R \times \mathcal{Q}_L$ such that there is u-labeled path in the NFA from (p, r) to (s, q). Moreover the path is uniquely defined. This is easily seen by induction on the length of u.

In particular, u is in L if and only if there is such a path from $(q_{0,L}, r)$ to $(s, q_{0,R})$ with $s \in \mathcal{F}_L$. By symmetry, u is in R if and only if $r \in \mathcal{F}_R$ for that path.

Now for each pair $(P, Q) \in \mathcal{Q} \times \mathcal{Q}$ we define a regular language $\mathcal{R}[P, Q]$ by

$$\mathcal{R}[P, Q] = \{u \in \Sigma^* \mid \text{There is a } u\text{-labeled path from } P \text{ to } Q\}.$$

There are at most n^4 such regular languages and for each of them we can test emptiness in polynomial time. For $P = (p, r)$ and $Q = (s, q)$ we obtain

$$\mathcal{R}[P, Q] = \{u \in \Sigma^* \mid p \cdot u = s \wedge r = u \cdot q\}.$$

3.2 A First Linear Context-Free Grammar

We continue with the same notations. In addition we view each symbol $[P, Q]$ with $(P, Q) \in \mathcal{Q} \times \mathcal{Q}$ as a variable of a context-free grammar. First we define productions of the form

$$[P, Q] \longrightarrow a[R, S]\overline{a}$$

with $a \in \Sigma$. We do so for all $[P, Q], [R, S]$ and a, where (P, a, R) and $(S, \overline{a}\,, Q)$ are arcs in the NFA above. For example, let $P = (p_1, p_2)$ and $R = (r_1, r_2)$, then we must have $p_1 \cdot a = r_1$ and $p_2 = a \cdot r_2$.

Moreover, we introduce chain rules

$$[P, Q] \longrightarrow \mathcal{R}_0[P, Q],$$

where $\mathcal{R}_i[P, Q]$ denotes a variable for $0 \leq i < k$; and $\mathcal{R}_k[P, Q]$ denotes a new terminal symbol. Of course, the idea is that we are free to substitute $\mathcal{R}_k[P, Q]$ by the regular language $\mathcal{R}[P, Q]$.

The index i can be viewed as a level where we produce the words α and $\overline{\alpha}$ used in the hairpin. This idea leads us to the third type of productions. These productions are of the form

$$\mathcal{R}_{i-1}[P, Q] \longrightarrow a\mathcal{R}_i[R, S]\overline{a}$$

where $1 \leq i \leq k$ and again $a \in \Sigma$. In order to have rules of the third type we impose again that (P, a, R) and $(S, \overline{a}\,, Q)$ are arcs in the NFA above.

We obtain a linear grammar with variables $[P, Q]$, $\mathcal{R}_i[P, Q]$, $0 \leq i < k$, and terminal symbols a, $\overline{a}\,$, and $\mathcal{R}_k[P, Q]$ with $a \in \Sigma$, and $\mathcal{R}_i[P, Q]$ as above. Note that the symbols $\mathcal{R}_0[P, Q]$ produce finite languages of the form $\alpha\mathcal{R}_k[R, S]\overline{\alpha}$ with $|\alpha| = k$. In particular, replacing the symbol $\mathcal{R}_k[R, S]$ by the language $\mathcal{R}[R, S]$, the symbol $\mathcal{R}_0[P, Q]$ produces a regular language, too.

Consider next a derivation

$$[P, Q] \overset{*}{\Longrightarrow} \gamma\mathcal{R}_i[R, S]\overline{\gamma}\,.$$

Let $P = (p_1, p_2)$, $Q = (q_1, q_2)$, $R = (r_1, r_2)$, $S = (s_1, s_2)$ be states in the NFA and $w \in \mathcal{R}_i[R, S]$ be a word.

This implies:

$$\begin{aligned} p_1 \cdot \gamma &= r_1, & p_2 &= \gamma \cdot r_2, \\ r_1 \cdot w &= s_1, & r_2 &= w \cdot s_2, \\ s_1 \cdot \overline{\gamma} &= q_1, & s_2 &= \overline{\gamma} \cdot q_2. \end{aligned}$$

In particular, we have

$$p_1 \cdot \gamma w\overline{\gamma} = q_1, \quad p_2 = \gamma w\overline{\gamma} \cdot q_2.$$

For the other direction, assume we have $p_1 \cdot \gamma w\overline{\gamma} = q_1$ and $p_2 = \gamma w\overline{\gamma} \cdot q_2$ with $|\gamma| \geq k$. Then, for each $1 \leq i \leq k$, there are uniquely defined symbols

$[P,Q], \mathcal{R}_i[R,S]$ with $P=(p_1,p_2)$, $Q=(q_1,q_2)$, $R=(r_1,r_2)$, $S=(s_1,s_2)$ and a word $w \in \mathcal{R}_i[R,S]$ such that we find a derivation:

$$[P,Q] \stackrel{*}{\Longrightarrow} \gamma \mathcal{R}_i[R,S]\overline{\gamma}\,.$$

In the next step we fix six states $P_0=(p_1,p_2)$, $Q_0=(q_1,q_2)$, $R_0=(r_1,r_2)$, $S_0=(s_1,s_2)$, $I_0=(i_1,i_2)$, and $J_0=(j_1,j_2)$, with the following properties:

1.) $p_1 = q_{0,L}$ is the initial state in the DFA above accepting L.
2.) $q_2 = q_{0,R}$ is the initial state in the right-to-left DFA above accepting R.
3.) Either $s_1 \in \mathcal{F}_L$ or $r_2 \in \mathcal{F}_R$ or both.
4.) There is a k-step derivation $\mathcal{R}_0[R_0,S_0] \stackrel{k}{\Longrightarrow} \alpha\mathcal{R}_k[I_0,J_0]\overline{\alpha}\,.$

The number of possible ways to choose these six states is bounded by $n_L^5 \cdot n_R^5$, hence at most n^{10}. By symmetry we assume in addition that we have $s_1 \in \mathcal{F}_L$, thus whenever $[P_0,Q_0] \stackrel{*}{\Longrightarrow} \gamma\mathcal{R}_0[R_0,S_0]\overline{\gamma}$ and $w \in \mathcal{R}[R_0,S_0]$, then we know $\gamma w \in L$.

We continue as follows: We choose the variable $[P_0,Q_0]$ to be the single axiom of the linear grammar G_0 we are going to define. We restrict the terminal alphabet to be the set $\Sigma \cup \{\mathcal{R}_k[I_0,J_0]\}$.

Next, we remove more productions and variables. On level 0 we only keep one single variable, namely $\mathcal{R}_0[R_0,S_0]$. Thus, all terminal derivations admit the form:

$$[P_0,Q_0] \stackrel{*}{\Longrightarrow} \gamma\mathcal{R}_0[R_0,S_0]\overline{\gamma} \stackrel{k}{\Longrightarrow} \gamma\alpha\mathcal{R}_k[I_0,J_0]\overline{\alpha}\;\overline{\gamma}\;.$$

So far, the productions can be assumed to be of three types:

$$\begin{aligned}[P,Q] &\longrightarrow a[R,S]\overline{a}\,,\\ [R_0,S_0] &\longrightarrow \mathcal{R}_0[R_0,S_0],\\ \mathcal{R}_{i-1}[P,Q] &\longrightarrow a\mathcal{R}_i[R,S]\overline{a}\end{aligned}$$

Now we remove all productions $[P,Q] \longrightarrow a[R,S]\overline{a}$ where $P=(p_1,p_2)$ and $Q=(q_1,q_2)$ with either $q_1 \in \mathcal{F}_L$ or $p_2 \in \mathcal{F}_R$ or both. Let us call this new linear grammar G_0. Derivation in the grammar G_0 look as follows.:

$$[P_0,Q_0] \stackrel{*}{\Longrightarrow} \gamma_1[P,Q]\overline{\gamma_1} \stackrel{*}{\Longrightarrow} \gamma\mathcal{R}_0[R_0,S_0]\overline{\gamma} \stackrel{k}{\Longrightarrow} \gamma\alpha\mathcal{R}_k[I_0,J_0]\overline{\alpha}\;\overline{\gamma}\;.$$

Now let $\beta \in \mathcal{R}[I_0,J_0]$ and $w=\alpha\beta\overline{\alpha}$, then we know that either $\gamma w = \gamma\alpha\beta\overline{\alpha} \in L$ or $w\overline{\gamma} = \alpha\beta\overline{\alpha}\;\overline{\gamma} \in R$ or both, but every prefix of $\gamma w\overline{\gamma}$ belonging to L is a prefix of γw and every suffix belonging to R is a suffix of $w\gamma$.

As usual, the generated language is called $L(G_0)$. By $\mathcal{H}(G_0)$ we mean the language where we substitute the terminal symbol $\mathcal{R}_k[I_0,J_0]$ by the (non-empty) regular language $\mathcal{R}[I_0,J_0]$. Thus,

$$\mathcal{H}(G_0) = \left\{ \gamma\alpha\beta\overline{\alpha}\;\overline{\gamma} \;\middle|\; [P_0,Q_0] \underset{G_0}{\stackrel{*}{\Longrightarrow}} \gamma\alpha\mathcal{R}_k[I_0,J_0]\overline{\alpha}\;\overline{\gamma} \wedge \beta \in \mathcal{R}[I_0,J_0] \right\}.$$

By the very construction $\mathcal{H}(G_0) \subseteq \mathcal{H}(L, R, k)$. Moreover, every word in the hairpin completion $\mathcal{H}(L, R, k)$ belongs to one of these $\mathcal{H}(G_0)$. Thus, $\mathcal{H}(L, R, k)$ is regular if and only if for all these $\mathcal{H}(G_0)$ we find regular languages $\mathcal{R}(G_0)$ such that $\mathcal{H}(G_0) \subseteq \mathcal{R}(G_0) \subseteq \mathcal{H}(L, R, k)$.

Thus, it is enough to show that we can decide in polynomial time whether there is such a regular language $\mathcal{R}(G_0)$ for a given grammar G_0 as above.

Note that we can test in polynomial time whether $L(G_0) \subseteq \Sigma^* \mathcal{R}_k[I_0, J_0]\Sigma^*$ is finite. In the case that $L(G_0)$ is finite, we are done, because $\mathcal{H}(G_0)$ is obtained by substituting $\mathcal{R}_k[I_0, J_0]$ by a regular language. So we can choose $\mathcal{R}(G_0) = \mathcal{H}(G_0)$.

In the spirit of an algorithm we could also say:

Test 1. Check whether $L(G_0)$ is finite. If yes, we construct the next grammar of this type.

We continue with the linear grammar G_0 under the assumption that $L(G_0)$ is infinite and that the grammar is reduced. This means all symbols are reachable and productive. Since $L(G_0)$ is infinite there must be variables of the form $[P, Q]$ and non-trivial derivations:

$$[P, Q] \overset{+}{\underset{G_0}{\Longrightarrow}} [P, Q].$$

There are at most n^4 such symbols. They are called *self-reproducing* symbols in the following. Let us fix one self-reproducing symbol and denote it by $[P', Q']$. We define a linear context-free grammar G_1 and a language $L(G_1)$ given as the following set:

$$\left\{ \pi\gamma\alpha\mathcal{R}_k[I_0, J_0]\overline{\alpha}\;\overline{\gamma}\,\overline{\pi} \;\middle|\; [P_0, Q_0] \overset{\leq n^4}{\underset{G_0}{\Longrightarrow}} \pi[P', Q']\overline{\pi} \overset{*}{\underset{G_0}{\Longrightarrow}} \pi\gamma\alpha\mathcal{R}_k[I_0, J_0]\overline{\alpha}\;\overline{\gamma}\,\overline{\pi} \right\}.$$

This gives us at most n^4 grammars G_1 of polynomial size such that $L(G_0)$ is, up to finitely many elements, the union of languages $L(G_1)$. Note also that each language $L(G_1)$ is infinite by construction.

As above, we also have a linear context-free language $\mathcal{H}(G_1)$ by defining:

$$\mathcal{H}(G_1) = \{\pi\gamma\alpha\beta\overline{\alpha}\;\overline{\gamma}\;\overline{\pi} \mid \pi\gamma\alpha\mathcal{R}_k[I_0, J_0]\overline{\alpha}\;\overline{\gamma}\,\overline{\pi} \in L(G_1) \wedge \beta \in \mathcal{R}[I_0, J_0]\}\,.$$

This reduces the proof of Theorem 2 to the following statement: We can decide in polynomial time whether there is a regular language $\mathcal{R}$ such that $\mathcal{H}(G_1) \subseteq \mathcal{R} \subseteq \mathcal{H}(L, R, k)$.

For $[P', Q']$ we compute two words π and p with length $0 < |\pi|, |p| \leq n^4$ such that we have:

$$[P_0, Q_0] \overset{+}{\underset{G_1}{\Longrightarrow}} \pi[P', Q']\overline{\pi} \overset{+}{\underset{G_1}{\Longrightarrow}} \pi p[P', Q']\overline{p}\,\overline{\pi}\,.$$

N.B., there are perhaps many choices for π and p, but we content to fix one pair (π, p) for each $[P', Q']$. As we will see below, the solution to Problems 1 and 2 can be based on these fixed pairs!

The main idea is from now to investigate the effect of pumping the word p under the assumption that the hairpin completion is regular. This means we consider derivations $[P_0, Q_0] \underset{G_1}{\overset{+}{\Longrightarrow}} \pi p^s [P', Q'] \overline{p}^{\,s}\, \overline{\pi}$, where s is huge and $\mathcal{H}(L, R, k)$ is regular.

Consider some $\beta \in \mathcal{R}[I_0, J_0]$ and $\pi v \alpha \mathcal{R}_k[I_0, J_0] \overline{\alpha}\, \overline{v}\, \overline{\pi} \in L(G_1)$. The choice of the word p implies $[P', Q'] \underset{G_1}{\overset{+}{\Longrightarrow}} p[P', Q']\overline{p}$ and hence, for all $s \in \mathbb{N}$ we have

$$z_s = \pi p^s v \alpha \beta \overline{\alpha}\, \overline{v}\, \overline{p}^{\,s}\, \overline{\pi} \in \mathcal{H}(G_1)$$

and the word $\pi p^s v \alpha \beta \overline{\alpha}$ is the longest prefix of z_s in L; and moreover, if a suffix of z_s belongs to R, then it is a suffix of $\alpha \beta \overline{\alpha}\, \overline{v}\, \overline{p}^{\,s}\, \overline{\pi}$.

Assume for a moment that $\mathcal{H}(L, R, k)$ is regular, then we find $s > 0$ such that p^s is idempotent in the syntactic monoid of $\mathcal{H}(L, R, k)$. However, this means that $\pi p^{sy} v \alpha \beta \overline{\alpha}\, \overline{v}\, \overline{p}^{\,s}\, \overline{\pi} \in \mathcal{H}(L, R, k)$ where s is perhaps large, but y can be taken as huge as we need. Now, for the hairpin we do not have the option to build it on the right, because $\alpha \beta \overline{\alpha}\, \overline{v}\, \overline{p}^{\,s}\, \overline{\pi}$ is too short compared to length of the whole word (it must cover more than half of the length). Thus, we must use the longest prefix $\pi p^{sy} v \alpha \beta \overline{\alpha}$ in L for the hairpin. But this implies that $v\alpha$ is a prefix of some power of p.

This leads to the following lemma:

Lemma 1. *Let $\mathcal{H}(L, R, k)$ be regular. Then $v\alpha$ is a prefix of some power of the word p for all derivations $[P', Q'] \underset{G_1}{\overset{*}{\Longrightarrow}} v \alpha \mathcal{R}_k[I_0, J_0] \overline{\alpha}\, \overline{v}$.*

Proof. This is clear, choose some $\beta \in \mathcal{R}[I_0, J_0]$ and derivation $[P_0, Q_0] \underset{G_1}{\overset{*}{\Longrightarrow}} \pi v \alpha \beta \overline{\alpha}\, \overline{v}\, \overline{\pi}$; and argue as above.

We have also the following complexity result:

Lemma 2. *There is a polynomial time algorithm which checks whether for all derivations $[P', Q'] \underset{G_1}{\overset{*}{\Longrightarrow}} v \alpha \mathcal{R}_k[I_0, J_0] \overline{\alpha}\, \overline{v}$ if we have that $v\alpha$ is a prefix of some power of p.*

Proof. This follows from a standard construction. For the language

$$X = \left\{ w \mathcal{R}_k[I_0, J_0] w' \in \Sigma^* \mathcal{R}_k[I_0, J_0] \Sigma^* \;\middle|\; w \text{ is no prefix of a word in } p^+ \right\}$$

we find a DFA with $|p| + 3$ states. Therefore we can check in polynomial time whether the following intersection is empty:

$$X \cap \left\{ v \alpha \mathcal{R}_k[I_0, J_0] \overline{\alpha}\, \overline{v} \in \Sigma^* \mathcal{R}_k[I_0, J_0] \Sigma^* \;\middle|\; [P', Q'] \underset{G_1}{\overset{*}{\Longrightarrow}} v \alpha \mathcal{R}_k[I_0, J_0] \overline{\alpha}\, \overline{v} \right\}$$

The intersection is empty if and only if for all derivations

$$[P', Q'] \underset{G_1}{\overset{*}{\Longrightarrow}} v \alpha \mathcal{R}_k[I_0, J_0] \overline{\alpha}\, \overline{v}$$

we have that $v\alpha$ is a prefix of some power of p.

This gives a non-trivial necessary condition.

Test 2. We check for all self-reproducing symbols $[P', Q']$ the condition in Lemma 2.

If one of the test fails, we know that the hairpin completion $\mathcal{H}(L, R, k)$ is not regular. Thus, in the following we assume that all self-reproducing symbols $[P', Q']$ passed this test.

3.3 Candidates

Thus by Test 2, for the rest of the proof we assume that all self-reproducing symbols $[P', Q']$ produce only terminal words of the form $p^s p' \alpha \mathcal{R}_k[I_0, J_0] \overline{\alpha}\, \overline{p'}\, \overline{p}^{\,s}$ where $s \geq 0$ and $p' \leq p$ and $p'\alpha$ is a prefix of some power of p. This condition remains valid if we replace p by some fixed power, say p^k. In particular, we may assume henceforth that $|p| \geq k$ and therefore α becomes a prefix of some conjugated word $q = p''p'$ with $p = p'p''$.

We use all these (at most n^4) symbols $[P', Q']$ and we collect all words p and all their conjugates $q = p''p'$ in a list of *candidates* $\mathcal{C}$. This list contains at most n^8 words, and $q \in \mathcal{C}$ defines a word α of length k such that $\alpha \leq q$.

We now need the reference to specific states in the DFAs. We have $P_0 = (q_{0,L}, p_2)$ and $P' = (p_1', p_2')$ and hence $q_{0,L} \cdot \pi = p_1'$ and $p_1' \cdot p = p_1'$. Let $q = p''p'$ with $p = p'p''$ and $c \in \mathcal{Q}_L$ such that $c = p_1' \cdot p'$. Then we have $c \cdot q = c$, too.

Moreover, let $J_0 = (j_1, j_2)$ and $f = j_1 \cdot \overline{\alpha}$, then we know that $f \in \mathcal{F}_L$ and (starting in f) reading any non-empty prefix of a word in $\overline{q}^{\,+}\overline{p'}\,\overline{\pi}$ cannot take us back to a final state. For the symmetric consideration we content that if $d = \overline{p'}\,\overline{\pi} \cdot q_{0,R} \in \mathcal{Q}_R$, then $d = \overline{q} \cdot d$.

The next step is to create a list $\mathcal{L}$ of tuples

$$(c, d, e, f, g, h, q) \in \mathcal{Q}_L \times \mathcal{Q}_R \times \mathcal{Q}_L \times \mathcal{Q}_L \times \mathcal{Q}_R \times \mathcal{Q}_R \times \mathcal{C},$$

which satisfy the following additional conditions:

1.) $f \in \mathcal{F}_L$ and reading any non-empty prefix of a word in $\overline{q}^{\,+}$ cannot take us back from f to a final state.
2.) $c = c \cdot q$ and $d = \overline{q} \cdot d$.
3.) $e \cdot \overline{q} = e$ and $f \cdot \overline{q}^{\,n} = e$.
4.) $g = q \cdot g$ and $g = q^n \cdot h$.

There are at most n^{14} elements in $\mathcal{L}$. We consider (c, d, e, f, g, h, q) one after another. For each tuple we define $\alpha \leq q$ by $|\alpha| = k$. We define a finite (!) language Π by all words $\pi \in \Sigma^*$ satisfying the following conditions:

1.) $|\pi| \leq 2n^4 + k$.
2.) $q_{0,L} \cdot \pi = c$, and $d = \overline{\pi} \cdot q_{0,R}$,
3.) For all $\eta \leq \overline{\pi}$ we have $e \cdot \eta \notin \mathcal{F}_L$.
4.) For all suffixes σ of π we have $\sigma \cdot g \notin \mathcal{F}_R$.

Note that an NFA of polynomial size for Π can be constructed in polynomial time, but the size of Π can be exponential, $|\Pi| \leq |\Sigma|^{2n^4+k}$. We also define a (possibly infinite) regular language B by all words $\beta \in \Sigma^*$ satisfying $c\cdot\alpha\beta\overline{\alpha} = f$, $h = \alpha\beta\overline{\alpha}\cdot d$, and $q\alpha$ is not a prefix of $\alpha\beta$. Again, an NFA of polynomial size for B can be constructed in polynomial time.

The idea behind this definition is as follows. Assume $\pi q^t q^n \alpha\beta\overline{\alpha}\,\overline{q}^{\,n}\overline{q}^{\,s}\,\overline{\pi}$ is in the hairpin closure, then we see these states as follows:

$$q_{0,L} \xrightarrow{\pi} c \xrightarrow{q^t q^n} c \xrightarrow{\alpha\beta\overline{\alpha}} f \xrightarrow{\overline{q}^{\,n}} e \xrightarrow{\overline{q}^{\,s}} e \xrightarrow{\overline{\pi}}$$

$$\xleftarrow{\pi} g \xleftarrow{q^t} g \xleftarrow{q^n} h \xleftarrow{\alpha\beta\overline{\alpha}} d \xleftarrow{\overline{q}^{\,n}\overline{q}^{\,s}} d \xleftarrow{\overline{\pi}} q_{0,R}$$

Let

$$\mathcal{H}(c,d,e,f,g,h,q) = \left\{\pi q^t \alpha\beta\overline{\alpha}\,\overline{q}^{\,s}\,\overline{\pi} \;\middle|\; \pi \in \Pi \wedge \beta \in B \wedge 0 \le s \le t\right\}.$$

Then obviously, $\mathcal{H}(c,d,e,f,g,h,q) \subseteq \mathcal{H}(L,R,k)$ because $\pi q^t\alpha\beta\overline{\alpha} \in L$. We claim that for the grammar G_1 as above and all words $w \in \mathcal{H}(G_1)$ there exists at least one tuple $(c,d,e,f,g,h,q) \in \mathcal{L}$ such that $w \in \mathcal{H}(c,d,e,f,g,h,q)$.

The crucial observation here is that we have introduced the states h and g just for the following purpose: We can write a word $w = \alpha\beta'\overline{\alpha}$ as $w = q^j\alpha\beta\overline{\alpha}$ such that $q\alpha$ is not a prefix of $\alpha\beta$. Then let $h = \alpha\beta\overline{\alpha}\cdot d$. The words w which play a role for $\mathcal{H}(G_1)$ are of the type that if we are during the right-to-left run in state h after reading $\alpha\beta\overline{\alpha}$, then for some perhaps huge t we reach the state $g = q^t \cdot h$ with $g = q\cdot g$. Indeed, we can use $g = p''\cdot p_2'$ where $P' = (p_1', p_2')$. But this means $g = q^n\cdot h$, too. We obtain a symmetric statement for e and f.

Thus, $\mathcal{H}(L,R,k)$ is regular if and only if for all $(c,d,e,f,g,h,q) \in \mathcal{L}$ we find regular languages $\mathcal{R}$ such that $\mathcal{H}(c,d,e,f,g,h,q) \subseteq \mathcal{R} \subseteq \mathcal{H}(L,R,k)$.

Note that for $\pi q^t\alpha\beta\overline{\alpha}\,\overline{q}^{\,s}\overline{\pi}$ in $\mathcal{H}(c,d,e,f,g,h,q)$ the longest prefix in L is the word $\pi q^t\alpha\beta\overline{\alpha}$, but we lost the control over the suffixes which are in R.

Clearly,

$$\left\{\pi q^t\alpha\beta\overline{\alpha}\,\overline{q}^{\,s}\,\overline{\pi} \;\middle|\; \pi \in \Pi \wedge \beta \in B \wedge 0 \le s < n \wedge s \le t\right\} \subseteq \mathcal{H}(L,R,k)$$

is a regular language because Π is finite and B is regular. Thus all we will have to show is the following.

Proposition 1. *Let*

$$\mathcal{H} = \mathcal{H}(c,d,e,f,g,h,q,n) = \left\{\pi q^t\alpha\beta\overline{\alpha}\,\overline{q}^{\,s}\,\overline{\pi} \;\middle|\; \pi \in \Pi \wedge \beta \in B \wedge n \le s \le t\right\}.$$

Then we can decide in polynomial time whether there is a regular language $\mathcal{R}$ *such that* $\mathcal{H} \subseteq \mathcal{R} \subseteq \mathcal{H}(L,R,k)$.

For the proof of Proposition 1 we start with the following test.

Test 3. Check in polynomial time whether there exists a suffix σ of q^n such that $\sigma\cdot h \in \mathcal{F}_R$ is a final state for R.

If Test 3 yields *yes*, then we can put

$$\mathcal{R} = \left\{\pi q^t\alpha\beta\overline{\alpha}\,\overline{q}^{\,s}\,\overline{\pi} \;\middle|\; \pi \in \Pi \wedge \beta \in B \wedge n \le s \wedge n \le t\right\}.$$

The set $\mathcal{R}$ is regular and satisfies $\mathcal{H} \subseteq \mathcal{R} \subseteq \mathcal{H}(L, R, k)$.

Thus, for the rest we assume that Test 3 is negative. Then the language $\mathcal{H}$ has some additional special features.

For $z_{t,s} = \pi q^t \alpha\beta\overline{\alpha}\,\overline{q}^{\,s}\overline{\pi} \in \mathcal{H}$ with $\pi \in \Pi$ and $\beta \in B$ and $n \le s \le t$ we know that the prefix $\pi' q^t \alpha\beta\overline{\alpha}$ belongs to L and it is the longest prefix with this property. If a suffix of $z_{t,s}$ belongs to R, then it is a suffix of $\alpha\beta\overline{\alpha}\,\overline{q}^{\,s}\overline{\pi}$, due to Test 3. Moreover, $q\alpha$ is not a prefix of $\alpha\beta$ which was the main purpose of defining B in such a way.

Let us assume that $\mathcal{H}(L, R, k)$ is regular, then there exists some $x > n$ such that $\overline{q}^{\,x}$ is idempotent in the syntactic monoid of $\mathcal{H}(L, R, k)$. Consider $t + 1 = s = 2x$.

Consider $z_t = \pi q^t \alpha\beta\overline{\alpha}\,\overline{q}^{\,t+1}\overline{\pi}$ with $\pi \in \Pi$ and $\beta \in B$. As $\overline{q}^{\,x}$ is idempotent and $\pi' q^t \alpha\beta\overline{\alpha}\,\overline{q}^{\,t+1-x}\,\overline{\pi'} \in \mathcal{H}(L, R, k)$ we see that $z_t \in \mathcal{H}(L, R, k)$, too. Since $q\alpha$ is not a prefix of $\alpha\beta$ the longest prefix in L becomes too short to create a hairpin completion for $\pi q^t \alpha\beta\overline{\alpha}\,\overline{q}^{\,t+1}\overline{\pi}$; we must use a suffix in R for that purpose. The longest suffix in R has the form $\delta u \in R$ with $|\delta| = k$, and it is a suffix of $\alpha\beta\overline{\alpha}\,\overline{q}^{\,t+1}\overline{\pi}$. Moreover as $|\alpha| = |\delta|$ we see that $\pi q^t \alpha$ must be a prefix of $\overline{u}$.

Thus, we must be able to write

$$\alpha\beta\overline{\alpha}\,\overline{q} = v\delta w\overline{\delta}\,\overline{v}$$

such that $\delta w\overline{\delta}\,\overline{v}\,\overline{q}^{\,t}\overline{\pi} \in R$. Now consider some huge y, say $y > |z_t|$. Then $\pi q^t \alpha\beta\overline{\alpha}\,\overline{q}^{\,t+1+xy}\,\overline{\pi} \in \mathcal{H}(L, R, k)$, too. Similar to an earlier observation this says that we can write $v\delta = q^m q'\delta$ with $m \ge 0$ and $q'\delta$ is a proper prefix of $q\alpha$. But we cannot have $m > 0$, since, again, $q\alpha$ is not a prefix of $\alpha\beta$.

Thus, if $\mathcal{H}(L, R, k)$ is regular, then $v\delta < q\alpha$ and $\alpha\beta\overline{\alpha}\,\overline{q} = v\delta u\overline{\delta}\,\overline{v}$ such that $\delta u\overline{\delta}\,\overline{v} \cdot d \in \mathcal{F}_R$.

This leads finally to another necessary condition. If $\mathcal{H}(L, R, k)$ is regular, then it must pass the following test:

Test 4. Check in polynomial time whether for all $\beta \in B$ there exist v, δ with $|\delta| = k$, $v\delta \le q\alpha$ and $\alpha\beta\overline{\alpha}\,\overline{q} = v\delta w$ with $|w| \ge |v\delta|$ and $\delta w \cdot d \in F_R$.

In order to perform a test in polynomial time we start with any NFA accepting the language

$$\{\alpha\beta\overline{\alpha}\,\overline{q} \mid \beta \in B\}.$$

Then we may take e.g. the cross product with the NFA constructed in Section 3.1, which, in particular, *knows* the state in $\mathcal{Q}_R$. This means if, in the new automaton, state Q *knows* $r \in \mathcal{Q}_R$ and if we can reach via a word z a final state, then we may infer $r = z \cdot d$. (This is because we may assume that in the right-to-left DFA d is an initial state for the right quotient $R(\overline{q}^{\,n}\overline{\pi})^{-1}$.) Recall that whenever we investigate properties of NFA, we first do a clean-up. Thus, we assume that all states are reachable and co-reachable.

We continue to modify the new NFA as follows. We duplicate each state Q several times so that each state becomes the form $[i, Q, j]$ with $i \in \{0, \ldots, |q|, *\}$ and $j \in \{0, \ldots, |q| + 2k, *\}$, where $*$ is a special symbol standing for integers greater than $|q|$, respectively greater than $|q| + 2k$.

After a transformation we may assume that if the NFA accepts a word uz with $|u| = i$ and $|z| = j$, then we are sure that reading u we reach some state $[i, Q, j]$. Vice versa if we reach after reading u a state $[i, Q, j]$, then $|u| = i$ and $|z| = j$ for every word z which takes $[i, Q, j]$ to some final state.

We duplicate the states again, and we introduce upper and lower states. We start in the upper part, but as soon as we deviate from reading a prefix $q\alpha$ we switch to the lower part. We switch also to the lower part if $j < k$. Once we are in the lower part we remain there. Note that the last k states on an accepting path are lower.

On every accepting path there is exactly one upper state U where the next state is a lower state.

Remember that our NFA of Section 3.1 transfers the following property: If we accept now a word uz with $|u| = i$ and if after reading u we reach $[i, Q, j]$, then we know the state $z \cdot d$ of the right-to-left DFA for R. Let us mark all upper states $[i, Q, j]$ as *good*, if both $z \cdot d \in \mathcal{F}_R$ and $i + 2k \le j$.

It is clear that every accepting path must go through some good upper state, otherwise Test 4 fails. This can be decided via a reachability algorithm. Finally consider all accepting paths and compute the set of good upper states $[i, Q, j]$ which are seen first on such paths. For each such states all outgoing paths of length k must stay in the upper part, otherwise Test 4 fails. If no such $[i, Q, j]$ leads to a failure, Test 4 is positive.

Now, all tests have been performed; and we get our result due to the following conclusion: Assume Test 4 is positive. Then we have for all $s, t \ge n$ the following fact:

$$z_{t,s} = \pi q^t \alpha\beta\overline{\alpha}\,\overline{q}\,\overline{q}^{\,s}\overline{\pi} \in \mathcal{H}(L, R, k)$$

Indeed for $t > s$ this holds because $\pi q^t \alpha\beta\overline{\alpha} \in L$. For $n \le t \le s$ we use that there exist v, δ with $|\delta| = k$, $v\delta \le p\alpha$, and $\alpha\beta\overline{\alpha}\,\overline{q} = v\delta w$ with $|w| \ge |v\delta|$, and $\delta w\overline{\pi} \cdot q_{0,R} \in \mathcal{F}_R$. Thus $z_{t,s} = \pi q^t v\delta u\overline{\delta}\,\overline{v}\,\overline{q}^{\,s}\overline{\pi}$ and $z_{t,s} \in \mathcal{H}(L, R, k)$ because $\delta u\overline{\delta}\,\overline{v}\,\overline{q}^{\,s}\overline{\pi} \in R$.

Open problems

We conclude with four questions which might be interesting for future research.

Question 1. What is the complexity of our decision algorithm in terms of n, if we start with a finite monoid of size n recognizing both L and R?

Question 2. What is the *practical performance* of our decision algorithm?

Let us define the *partial hairpin completion* of L by the set of words $\gamma\alpha\beta\overline{\alpha}\,\overline{\gamma}'$ where γ' is a prefix γ and $\gamma\alpha\beta\overline{\alpha} \in L$ or γ is a prefix γ' and $\alpha\beta\overline{\alpha}\,\overline{\gamma}' \in L$. (In particular, L becomes a subset of its partial hairpin completion.)

Question 3. Is it decidable whether the partial hairpin completion applied to a regular language is regular again?

Given a language L we can iterate the (partial) hairpin completion and can define the *iterated (partial) hairpin completion* as the union over all iterations.

Question 4. Is it decidable whether the iterated (partial) hairpin completion applied to a regular language (finite language resp.) is regular again?

Acknowledgement

We thank the anonymous referees for many useful remarks and hints.

References

1. Cheptea, D., Martin-Vide, C., Mitrana, V.: A new operation on words suggested by DNA biochemistry: Hairpin completion. Transgressive Computing, 216–228 (2006)
2. Deaton, R., Murphy, R., Garzon, M., Franceschetti, D., Stevens, S.: Good encodings for DNA-based solutions to combinatorial problems. Proc. of DNA-based computers DIMACS Series 44, 247–258 (1998)
3. Garzon, M., Deaton, R., Neathery, P., Murphy, R., Franceschetti, D., Stevens, E.: On the encoding problem for DNA computing. In: The Third DIMACS Workshop on DNA-Based Computing, pp. 230–237 (1997)
4. Garzon, M., Deaton, R., Nino, L., Stevens Jr., S., Wittner, M.: Genome encoding for DNA computing. In: Proc. Third Genetic Programming Conference, pp. 684–690 (1998)
5. Hopcroft, J.E., Ulman, J.D.: Introduction to Automata Theory, Languages and Computation. Addison-Wesley, Reading (1979)
6. Ito, M., Leupold, P., Mitrana, V.: Bounded hairpin completion. In: LATA. LNCS, vol. 5457, pp. 434–445. Springer, Heidelberg (2009)
7. Kari, L., Konstantinidis, S., Losseva, E., Sosík, P., Thierrin, G.: Hairpin structures in DNA words. In: Carbone, A., Pierce, N.A. (eds.) DNA 2005. LNCS, vol. 3892, pp. 158–170. Springer, Heidelberg (2006)
8. Kari, L., Mahalingam, K., Thierrin, G.: The syntactic monoid of hairpin-free languages. Acta Inf. 44(3-4), 153–166 (2007)
9. Manea, F., Martín-Vide, C., Mitrana, V.: On some algorithmic problems regarding the hairpin completion. Discrete Applied Mathematics 27, 71–72 (2006)
10. Manea, F., Mitrana, V.: Hairpin completion versus hairpin reduction. In: Cooper, S.B., Löwe, B., Sorbi, A. (eds.) CiE 2007. LNCS, vol. 4497, pp. 532–541. Springer, Heidelberg (2007)
11. Manea, F., Mitrana, V., Yokomori, T.: Some remarks on the hairpin completion. In: 12th International Conference on Automata and Formal Languages, pp. 302–313 (2008)
12. Manea, F., Mitrana, V., Yokomori, T.: Two complementary operations inspired by the DNA hairpin formation: Completion and reduction. Theor. Comput. Sci. 410(4-5), 417–425 (2009)