Information and Computation 217 (2012) 12–30

Contents lists available at SciVerse ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

Deciding regularity of hairpin completions of regular languages in polynomial time [☆]

Volker Diekert [a,*], Steffen Kopecki [a,b], Victor Mitrana [c]

[a] University of Stuttgart, Institute for Formal Methods in Computer Science, Universitätsstraße 38, 70569 Stuttgart, Germany
[b] The University of Western Ontario, Department of Computer Science, Middlesex College, London, Ontario N6A 5B7, Canada
[c] University of Bucharest, Faculty of Mathematics and Computer Science, Str. Academiei 14, 010014, Bucharest, Romania

ARTICLE INFO

Article history:
Received 10 August 2011
Revised 3 April 2012
Available online 3 May 2012

Keywords:
Hairpin completion
Regular language
Finite automaton
Unambiguous linear language
Rational growth

ABSTRACT

Hairpin completion is an operation on formal languages that has been inspired by hairpin formation in DNA biochemistry and by DNA computing. In this paper we investigate the one- and two-sided hairpin completion of regular languages. We solve an open problem from the literature by showing that the *regularity problem* for hairpin completions is decidable. Actually, we show that the problem is decidable in polynomial time if the input is specified by DFAs.

Furthermore, we prove that the hairpin completion of regular languages is an unambiguous linear context-free language. Beforehand, it was known only that it is linear context-free. Unambiguity is a strong additional property because it allows to compute the growth function or the topological entropy. In particular, we can compare the growth of the hairpin completion with the growth of the defining regular languages. We show that the growth of the hairpin completion is exponential if and only if the growth of the underlying languages is exponential. Even if both growth functions are exponential, they can be as far apart as $2^{\Theta(\sqrt{n})}$ for the hairpin completion and $2^{\Theta(n)}$ for the defining regular languages. However, if the hairpin completion is still regular, then the hairpin completion and its underlying language have essentially the same growth and same topological entropy.

1. Introduction

A *DNA strand* can be seen as a word over the four-letter alphabet $\{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{T}\}$ where the letters represent the nucleobases Adenine, Cytosine, Guanine, and Thymine, respectively. By a *Watson–Crick base pairing* two strands may bond to each other if they have opposite orientation and their bases are pairwise complementary, where A is complementary to T and C to G; see Fig. 1 for a picture. Throughout, we use a bar-notation for the Watson–Crick complement and its language theoretic pendant; thus $\overline{\mathsf{A}} = \mathsf{T}$ and $\overline{\mathsf{C}} = \mathsf{G}$. For base sequences we extend the bar-notation to strings by $\overline{a_1 \cdots a_m} = \overline{a_m} \cdots \overline{a_1}$. This yields an involution which is an anti-morphism.

The *polymerase chain reaction* (PCR) is a technique which is used in DNA computing to amplify a *template strand* or a fragment of the template strand. Short DNA sequences, so-called *primers*, bond to parts of the template. After bonding, the primer is *extended* to a strand which is complementary to a fragment of the template. A particular case of this process is

☆ The revision of this paper was completed during the BIRS workshop *Outstanding Challenges in Combinatorics on Words* and an additional research stay at The Banff Centre. V.D. and S.K. greatly acknowledge the inspiring creativity of The Banff Centre and its hospitality. The research of S.K. was partially supported by the UWO Faculty of Science grant to Lila Kari.

* Corresponding author.
E-mail addresses: diekert@fmi.uni-stuttgart.de (V. Diekert), steffen@csd.uwo.ca (S. Kopecki), mitrana@fmi.unibuc.ro (V. Mitrana).

0890-5401/$ – see front matter
http://dx.doi.org/10.1016/j.ic.2012.04.003

$$\begin{array}{c} 5'-\mathrm{C}-\mathrm{G}-\mathrm{G}-\mathrm{T}-\mathrm{T}-\mathrm{T}-\mathrm{C}-\mathrm{A}-\mathrm{T}-\mathrm{C}-\mathrm{A}-3' \\ 3'-\mathrm{G}-\mathrm{C}-\mathrm{C}-\mathrm{A}-\mathrm{A}-\mathrm{A}-\mathrm{G}-\mathrm{T}-\mathrm{A}-\mathrm{G}-\mathrm{T}-5' \end{array}$$

Fig. 1. Bonding leads to base-wise complementary strands. Reading both strands in $5'$-to-$3'$ orientation we can write them as σ and $\overline{\sigma}$.

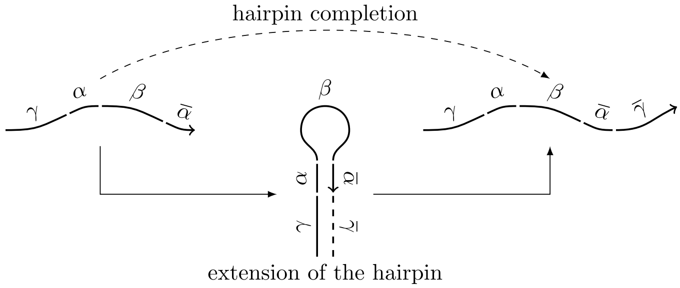

Fig. 2. Hairpin completion of a strand.

the *hairpin completion* of a strand, which may develop during the PCR. Suppose a strand σ can be written as $\sigma = \gamma\alpha\beta\overline{\alpha}$. Its suffix $\overline{\alpha}$ can act as a primer to the strand and form an intramolecular base-pairing which is called *hairpin* or *hairpin formation*. During the extension process we obtain strands of the form $\gamma\alpha\beta\overline{\alpha}\overline{\gamma'}$, where γ' is a prefix of γ. In this paper we are interested in the situation where the extension is completed; this means $\gamma' = \gamma$. Thus, $\gamma\alpha\beta\overline{\alpha}$ may form a hairpin which leads to its *hairpin completion* $\gamma\alpha\beta\overline{\alpha}\overline{\gamma}$, see Fig. 2. The primer α should consist of several bases, otherwise the bond between α and $\overline{\alpha}$ is too weak. The length of the primer depends on the reaction conditions, like temperature, e.g., [1] suggests that α should have 9 bases. The actual value of $|\alpha|$ is not very important for the following. However, in our algorithms we must pay attention to the fact that α and $\overline{\alpha}$ may overlap.

Hairpin completions are often seen as undesirable byproducts that occur during DNA computations and, therefore, sets of DNA strands have been investigated that do not tend to form hairpins or other undesired hybridizations, see e.g., [2–6] and the references within. On the other hand, DNA algorithms have been designed that make positive use of hairpins and hairpin completions. For example, the *whiplash PCR* is a technique where a single DNA strand computes one run of a non-deterministic GOTO-machine by repetitive hairpin completions, where the length of the extended part is controlled by *stopper sequences*. Starting with a huge set of strands, all runs of such a machine can be computed in parallel. The whiplash PCR was proposed as a computational model which, in theory, is able solve NP-complete problems like the Hamiltonian path problem [7,8]. The practical implementation of whiplash PCR is discussed in [9].

Motivated by the hairpin formation in biochemistry, the hairpin completion of formal languages has been introduced in 2006 by Cheptea, Martín-Vide, and Mitrana [10]. Our paper continues the investigation of hairpin formation from a purely formal language theoretical viewpoint. The hairpin completion of languages L_1 and L_2 contains all right hairpin completions (as in Fig. 2) of all words in L_1 and all left hairpin completions (a word $\alpha\beta\overline{\alpha}\overline{\gamma}$ is extended to the left by γ) of all words in L_2. Thus, L_1 and L_2 can be viewed as generators for the hairpin completion. It turns out to be best to work with two languages simultaneously, whereas the original definitions used only one language. A formal definition of the hairpin completion is given in Section 2.1. The hairpin completion and some related operations have been investigated in a series of papers, see e.g. [11–18]. It is known from [10] that hairpin completions of regular languages are linear context-free, and it has been asked whether the regularity problem for hairpin completions. In the original setting the input to the problem is given by some regular language L. The question is whether the one-sided (tow-sided respectively) hairpin completion of L is regular.

The problem is far from being trivial, because the regularity problem is undecidable for linear context-free languages. The regularity problem for hairpin completions has been shown to be decidable in a preliminary conference abstract of the present paper in [19]. Actually, we proved that the problem is decidable in polynomial time if the input is given by DFAs. In this first approach we did not spend efforts in optimization; the degree of the polynomial for the time bound was about 20. In a second approach, which was presented as conference abstract in [20], we improved the decision algorithm and provided the time bounds which are presented here, i.e., the regularity problem can be decided in time $\mathcal{O}(n^8)$, where n denotes an upper bound for the sizes of the two input DFAs accepting L_1 and $\overline{L_2}$, respectively. Furthermore, for $L_2 = \emptyset$ we provided a time complexity of $\mathcal{O}(n^2)$ and for $L_1 = \overline{L_2}$ we provided $\mathcal{O}(n^6)$. In the second abstract we also showed that the problem is NL-complete. NL is the class of problems that are solvable by a non-deterministic Turing machine using logarithmic space, only. Since NL belongs to *Nick's Class* NC, the regularity problem is efficiently solvable in parallel. Since NL $\subseteq$ NC (see e.g. [21]), the regularity problem is efficiently solvable in parallel. Moreover, we proved that the hairpin completion of regular languages is an unambiguous linear context-free language. Thus, its generating function is an effectively computable rational function.

A full proof for the NL-completeness result can be found in [22]. The present paper gives full proofs for the time bounds mentioned above and deals with our results about unambiguity.

This paper is organized as follows. In Section 2 we formally define the hairpin completion operation, we lay down our notation, and we briefly recall the some concepts of formal language theory. In Section 3 we start our investigation of hairpin completions of regular languages by providing an unambiguous linear context-free grammar generating the hairpin completion of two given regular languages. Section 4 is devoted to explain the algorithm which solves the regularity problem for hairpin completions of regular languages in time $\mathcal{O}(n^8)$ for the general case and in time $\mathcal{O}(n^2)$ for the one-sided case. In Section 5 we discuss the relation of the growths of hairpin completions with the growths of their underlying regular languages.

The paper is a journal version for results which have been presented at ICTAC 2009 and at CIAA 2010. It contains some additional results, too.

2. Preliminaries and notation

We assume the reader to be familiar with the fundamental concepts of formal language theory and automata theory as it can be found e.g. in the textbook [23]. By Σ we denote a finite alphabet with at least two letters which is equipped with an *involution* $\overline{}: \Sigma \to \Sigma$. An involution is a bijection such that $\bar{\bar{a}} = a$ for all $a \in \Sigma$. (In a biological setting we may think of $\Sigma = \{A, C, G, T\}$ with $\overline{A} = T$ and $\overline{C} = G$.) We extend this 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. Note that the involution on words is an anti-morphism since $\overline{uv} = \bar{v}\bar{u}$. For languages, $\overline{L}$ denotes the set $\{\overline{w} \mid w \in L\}$. The set of words over Σ is denoted by Σ^*; and the *empty word* is denoted by 1. By $\Sigma^{\leqslant m}$ we mean the set of all words with length of at most m.

Given a word w, we denote by $|w|$ its length, by $w[i] \in \Sigma$ its i-th letter, and by $w[i, j]$ the factor $w[i] \cdots w[j]$. A factor $w[1, j]$ is called a *prefix*, and a factor $w[i, |w|]$ is called a *suffix*. A prefix or suffix x of w is said to be *proper* if $x \neq w$. The prefix relation (respectively proper prefix relation) between words x and w is denoted by $x \leqslant w$ (respectively $x < w$).

2.1. Hairpin completion

Let L_1 and L_2 be languages in Σ^*. By κ we denote a (small) constant that gives the minimal length of primers. We define the *hairpin completion* $\mathcal{H}_\kappa(L_1, L_2)$ as an abstract operation by

$$\mathcal{H}_\kappa(L_1, L_2) = \left\{\gamma\alpha\beta\bar{\alpha}\bar{\gamma} \;\middle|\; (\gamma\alpha\beta\bar{\alpha} \in L_1 \vee \alpha\beta\bar{\alpha}\bar{\gamma} \in L_2) \wedge |\alpha| = \kappa\right\}.$$

We can think that L_1 and L_2 are generators of the hairpin completion $\mathcal{H}_\kappa(L_1, L_2)$. The language $\mathcal{H}_\kappa(L_1, L_2)$ clearly depends on L_1, L_2, but also on κ. Indeed, it might happen that $\mathcal{H}_{\kappa+1}(L_1, \emptyset) = \emptyset$ whereas $\mathcal{H}_\kappa(L_1, \emptyset)$ is not regular, even if L_1 is regular; e.g., consider $L_1 = a^*b\bar{b}$, then $\mathcal{H}_1(L_1, \emptyset) = \{a^i b\bar{b}\bar{a}^i \mid i \in \mathbb{N}\}$, but $\mathcal{H}_2(L_1, \emptyset) = \emptyset$. In our examples we will always choose $\kappa = 1$ for simplicity. However, e.g., due to possible overlaps between α and $\bar{\alpha}$, the general case is more complicated. Therefore we allow $\kappa \geqslant 1$. The definition of $\mathcal{H}_\kappa(L_1, L_2)$ does not change if we replace $|\alpha| = \kappa$ by $|\alpha| \geqslant \kappa$. This reflects the fact that the primer should have at least length κ.

Three cases are of main interest:

1. $L_2 = L_1$,
2. $L_2 = \overline{L_1}$,
3. $L_1 = \emptyset$ or $L_2 = \emptyset$.

Compared to the definition of the hairpin completion in [10,17] Case 1 corresponds to the two-sided hairpin completion and Case 3 to the one-sided hairpin completion. In biochemistry the hairpin completion is obtained in $5'$*-to-*$3'$ *direction* only, which corresponds to the one-sided case $\mathcal{H}_\kappa(L_1, \emptyset)$. So, strictly speaking, the general study of $\mathcal{H}_\kappa(L_1, L_2)$ is less supported by the biochemical motivation.

However, due to the complementary structure in DNA double strands, it is natural to assume that a strand and its complement co-occur. For a single language L we consider $\mathcal{H}_\kappa(L, \overline{L})$ in Case 2. Hence we obtain an operator $\overline{\mathcal{H}}_\kappa$ by $\overline{\mathcal{H}}_\kappa(L) = \mathcal{H}_\kappa(L, \overline{L})$. The construction $\overline{\mathcal{H}}_\kappa(L)$ is closer to the biochemistry setting than the traditional two-sided hairpin completion $\mathcal{H}_\kappa(L) = \mathcal{H}_\kappa(L, L)$ which has been investigated in earlier papers. Indeed, we always have $\overline{\mathcal{H}}_\kappa(L) = \mathcal{H}_\kappa(L, \emptyset) \cup \mathcal{H}_\kappa(\emptyset, \overline{L}) = \mathcal{H}_\kappa(L, \emptyset) \cup \overline{\mathcal{H}_\kappa(L, \emptyset)}$. Hence $\overline{\mathcal{H}}_\kappa(L)$ is the "Watson–Crick-complementary closure" (or closure by involution) of the one-sided hairpin completion $\mathcal{H}_\kappa(L, \emptyset)$. Interesting enough, it may happen that $L = \overline{L}$ and hairpin completion $\mathcal{H}_\kappa(L, \emptyset)$ is neither closed by involution nor regular, but $\overline{\mathcal{H}}_\kappa(L)$ is both: it is regular and clearly closed by involution. For example, let $L = \{a^n b\bar{b}\bar{a}, \; ab\bar{b}\bar{a}^n \mid n \geqslant 1\}$; obviously, $L = \overline{L}$. We obtain $\mathcal{H}_1(L, \emptyset) = \{a^n b\bar{b}\bar{a}^m \mid 1 \leqslant m \leqslant n \vee n = 1\}$, which is not regular, but $\overline{\mathcal{H}}_1(L) = a^+ b\bar{b}\bar{a}^+$ is regular.

In any case, to work with two languages L_1 and L_2 simultaneously is the convenient framework to study various special cases in a unified way. It gives the best results, solves an open problem from the literature, and from the formal language viewpoint the two-sided hairpin completion is the most challenging and interesting one. It also leads in a *natural* way the language operator $L \mapsto \overline{\mathcal{H}}_\kappa(L)$ which yields the desirable closure under involution.

2.2. Linear context-free grammars and unambiguity

A grammar G is a tuple $G=(V,\Sigma,P,\mathcal{S})$ where V is the finite set of *non-terminals*, Σ is the *terminal alphabet*, P is the finite set of *production rules*, and $\mathcal{S}\subseteq V$ is the set of *axioms*. We allow a set of axioms rather than the more usual restriction to have exactly one axiom S. The *size* of a grammar G is defined to be the sum $\sum_{\ell\to r\in P}(|\ell|+|r|)$. A grammar is called *context-free*, if every rule in P is of the form $A\to w$ where $A\in V$ and $w\in(V\cup\Sigma)^*$; a grammar is called *linear context-free*, or simply *linear*, if, in addition, w contains at most one non-terminal. For a context-free grammar G, a *derivation step* is denoted by $uAv\underset{G}{\Longrightarrow}uwv$, where $A\to w$ is a production rule in P and $u,v\in(V\cup\Sigma)^*$. By $\overset{*}{\underset{G}{\Longrightarrow}}$, we denote the reflexive and transitive closure of $\underset{G}{\Longrightarrow}$ and we call $u\overset{*}{\underset{G}{\Longrightarrow}}v$ (with $u,v\in(V\cup\Sigma)^*$) a *derivation*. The language generated by G is the set of terminal words

$$L(G)=\left\{w\in\Sigma^*\;\middle|\;\exists A\in\mathcal{S}\colon\; A\overset{*}{\underset{G}{\Longrightarrow}}w\right\}.$$

A linear grammar G is said to be *unambiguous* if for every word $w\in L(G)$, there is exactly one derivation $A\overset{*}{\underset{G}{\Longrightarrow}}w$ where $A\in\mathcal{S}$; in particular, there is only one axiom A that yields w. For general context-free grammars we would require that there is exactly one *left-most derivation* $A\overset{*}{\underset{G}{\Longrightarrow}}w$; but for linear grammars all derivations are left-most. A language L is called *(unambiguous) linear* if it is generated by an (unambiguous) linear grammar.

2.3. Generating functions, growth, and topological entropy

For a profound discussion of formal power series and how the growth of regular and unambiguous linear languages can be calculated we refer to [24–27]. We content ourselves with a few basic facts. The *growth* or *generating function* g_L of a formal language L is defined as

$$g_L(z)=\sum_{m\geqslant 0}\left|L\cap\Sigma^m\right|z^m.$$

We can view g_L as a formal power series or as an analytic function in one complex variable where the radius of convergence is strictly positive. The radius of convergence is at least $1/|\Sigma|$.

It is well known that the growth of a regular language L is effectively rational, i.e., it can be written as a quotient of two polynomials, and this holds more generally for unambiguous linear languages, see e.g., [24]. In particular, the growth is either polynomial or exponential. If the growth is exponential, then there exists an algebraic number $\lambda_L\in\mathbb{R}^{\geqslant 0}$, its growth indicator, such that $|L\cap\Sigma^m|$ behaves essentially like λ_L^m. More precisely, for a language L, its *growth indicator* is defined as the non-negative real number λ_L where

$$\lambda_L=\inf\left\{\lambda\in\mathbb{R}^{\geqslant 0}\;\middle|\;\exists c>0,\ \forall m\in\mathbb{N}\colon\ \left|L\cap\Sigma^m\right|\leqslant c\lambda^m\right\}.$$

The *topological entropy* of a language $L\subseteq\Sigma^*$ has been defined in [28] as

$$\lim_{m\to\infty}\sup\frac{\log_2|L\cap\Sigma^m|}{m}.$$

Thus, the topological entropy is equal to $\log_2(\lambda_L)$.

The growth of a language L is

1. exponential if $1<\lambda_L\leqslant|\Sigma|$, i.e., the topological entropy is positive,
2. sub-exponential but infinite if $\lambda_L=1$, i.e., the topological entropy is zero,
3. finite if $\lambda_L=0$, i.e., the topological entropy is equal to $-\infty$.

Note that other values for λ_L do not occur and that λ_L is the inverse of the convergence radius of $g_L(z)$. As we discussed above, the growth of an unambiguous linear language L is either polynomial or exponential; thus, if $\lambda_L=1$, the growth of L can be considered to be polynomial. Regular languages of polynomial growth have a very restricted form. Indeed, a regular language has polynomial growth if and only if it can be written as a finite union of languages of the form $u_0u_1^*u_2\cdots u_{2k-1}^*u_{2k}$ where u_i are words, see e.g., [29]. Thus, the more interesting situation occurs when a language has exponential growth. It is then when the growth indicator and the topological entropy become significant.

2.4. Regular languages and finite automata

Regular languages can be specified by non-deterministic finite automata (NFA) $\mathcal{A}=(\mathcal{Q},\Sigma,E,\mathcal{I},\mathcal{F})$, where $\mathcal{Q}$ is the finite set of *states*, $\mathcal{I}\subseteq\mathcal{Q}$ is the set of *initial states*, and $\mathcal{F}\subseteq\mathcal{Q}$ is the set of *final states*. The set E contains labeled *transitions* (or *arcs*), it is a subset of $\mathcal{Q} \times \Sigma \times \mathcal{Q}$. For a word $w \in \Sigma^*$ we write $p \xrightarrow{w} q$, if there is a path from state p to q which is labeled by w. Thus, the accepted language becomes

$$L(\mathcal{A}) = \left\{ w \in \Sigma^* \,\middle|\, \exists p \in \mathcal{I},\ \exists q \in \mathcal{F}\colon\ p \xrightarrow{w} q \right\}.$$

Later it will be crucial to use also paths which avoid final states. Let us write $p \xRightarrow{w} q$, if there is a path from state p to q which is labeled by the word w and which never enters a final state. Note that for such a path $p \xRightarrow{w} q$ we allow $p \in \mathcal{F}$.

An NFA is called a deterministic finite automaton (DFA), if it has one initial state and for every state $p \in \mathcal{Q}$ and every letter $a \in \Sigma$ there is exactly one arc $(p, a, q) \in E$. In particular, a DFA in this paper is always complete, thus we can read every word to its end. We also write $p \cdot w = q$, if $p \xrightarrow{w} q$. This yields a (totally defined) function $\mathcal{Q} \times \Sigma^* \to \mathcal{Q}$, which defines an action of Σ^* on $\mathcal{Q}$ on the right.

2.5. Specification of the input

Throughout the paper, L_1 and L_2 denote regular languages inside Σ^*. We use a DFA accepting L_1 as well as a DFA accepting $\overline{L_2}$. The latter automaton has the same number of states as (and is structurally isomorphic to) a DFA accepting the *reversal language* of L_2. Our input is therefore given by two DFAs $\mathcal{A}_i = (\mathcal{Q}_i, \Sigma, E_i, \{q_{0i}\}, \mathcal{F}_i)$ for $i = 1, 2$ which accept the languages L_1 and $\overline{L_2}$, respectively. We let $n_1 = |\mathcal{Q}_1|$, $n_2 = |\mathcal{Q}_2|$, and we let $n = \max\{n_1, n_2\}$ be the input size.

We do not require that $\mathcal{A}_1$ and $\mathcal{A}_2$ are *minimal*, however for convenience, we assume that every state is *essential*. This means that for $i = 1, 2$ every state in $\mathcal{Q}_i$ is reachable from the initial state q_{0i} and there is at most one *dead state* $t_i \in \mathcal{Q}_i$ from which no final state can be reached.

In terms of complexity, a situation $L_2 = \overline{L_1}$ behaves better than $L_2 = L_1$. For $L_2 = \overline{L_1}$ we can work with a single DFA because we may choose $\mathcal{A}_1 = \mathcal{A}_2$. If we have $L_1 = L_2$, then there might be an exponential gap between the number of states in $\mathcal{A}_1$ and $\mathcal{A}_2$, since $\mathcal{A}_1$ and $\mathcal{A}_2$ accept mutual "reversal languages". A standard example for this phenomenon is $L = \Sigma^n a \Sigma^*$ and $\bar{L} = \Sigma^* \bar{a} \Sigma^n$. A minimal complete DFA for L has $n + 2$ states whereas a minimal DFA for $\bar{L}$ needs more than 2^n states. As a matter of fact, the preference to $\mathcal{H}_\kappa(L, \bar{L})$ suits well to the biochemical background saying that a strand should co-occur with its Watson–Crick complement. It is the two-sided hairpin completion $\mathcal{H}_\kappa(L, \bar{L})$ which is closed under involution, whereas $\mathcal{H}_\kappa(L, L)$ is not closed under involution, in general. For example, let L be the singleton $\{ab\bar{a}\}$, then $\mathcal{H}_1(L, L) = L$ and $\mathcal{H}_1(L, \bar{L}) = \{ab\bar{a}, a\bar{b}\bar{a}\}$. Of course, if we start with a language L such that $L = \bar{L}$, then such a discussion is vacuous.

3. Unambiguity of $\mathcal{H}_\kappa(L_1, L_2)$

In this section we introduce the crucial concept of a *basic bridge*. The actual result in this section shows that the hairpin completion $\mathcal{H}_\kappa(L_1, L_2)$ is always an unambiguous linear context-free language. This result itself is not needed for deciding regularity of $\mathcal{H}_\kappa(L_1, L_2)$, but it came out as a byproduct of the decision procedure. Moreover, this fact turned out to be rather fundamental for the understanding of hairpin completions of regular languages. For example, it allows to compute the growth of $\mathcal{H}_\kappa(L_1, L_2)$ and to compare it with the growths of the languages L_1 and L_2. This will be shown in Section 5.

Theorem 3.1. *The hairpin completion $\mathcal{H}_\kappa(L_1, L_2)$ is an unambiguous linear context-free language. Moreover, there is an effective construction of a generating unambiguous linear grammar $G_\kappa(L_1 L_2) = (V, \Sigma, P, \mathcal{S})$ for $\mathcal{H}_\kappa(L_1, L_2)$ such that the size of the grammar $G_\kappa(L_1 L_2)$ is bounded by $(2\kappa + 10)|\Sigma|^\kappa n_1^2 n_2^2 \in \mathcal{O}(n_1^2 n_2^2) \subseteq \mathcal{O}(n^4)$.*

Let us give a brief informal outline how we prove Theorem 3.1. Consider first all words $\pi = \gamma\alpha\beta\overline{\alpha}\overline{\gamma} \in \Sigma^*$ with $|\alpha| = \kappa$. Clearly, these words are generated by some linear context-free grammar. Now we add the constraints that either $\gamma\alpha\beta\overline{\alpha} \in L_1$ or $\alpha\beta\overline{\alpha}\overline{\gamma} \in L_2$ or both. This yields $\mathcal{H}_\kappa(L_1, L_2)$ and the regular constraint $\gamma\alpha\beta\overline{\alpha} \in L_1$ or $\alpha\beta\overline{\alpha}\overline{\gamma} \in L_2$ can be put into a "finite control" associated to non-terminals. Hence, it is straightforward that $\mathcal{H}_\kappa(L_1, L_2)$ is linear context-free. For unambiguity we need to do more. We add the constraints that if a prefix of π belongs to L_1, then it is a prefix of $\gamma\alpha\beta\overline{\alpha}$, and if a suffix of π belongs to L_2, then it is a suffix of $\alpha\beta\overline{\alpha}\overline{\gamma}$. Now, this is still a constraint of regular type, so we have a larger, but still linear context-free grammar taking care of that. The most technical part of the proof below is to show that these constraints are strong enough to yield unambiguity.

Proof of Theorem 3.1. The crucial observation is that every word $\pi \in \mathcal{H}_\kappa(L_1, L_2)$ has a unique factorization $\pi = \gamma\alpha\beta\overline{\alpha}\overline{\gamma}$ such that

1. $\gamma\alpha\beta\overline{\alpha} \in L_1$ or $\alpha\beta\overline{\alpha}\overline{\gamma} \in L_2$,
2. $|\alpha| = \kappa$,
3. if a prefix of π belongs to L_1, then it is a prefix of $\gamma\alpha\beta\overline{\alpha}$, and
4. if a suffix of π belongs to L_2, then it is a suffix of $\alpha\beta\overline{\alpha}\overline{\gamma}$.

$$\mathcal{A}_1: \quad q_{01} \xrightarrow{\gamma} c_1 \xrightarrow{\alpha} d_1 \xrightarrow{\beta} e_1 \xrightarrow{\overline{\alpha}} f_1 \xRightarrow{\overline{\gamma}} q_1'$$
$$\mathcal{A}_2: \quad q_{02} \xrightarrow{\gamma} c_2 \xrightarrow{\alpha} d_2 \xrightarrow{\overline{\beta}} e_2 \xrightarrow{\overline{\alpha}} f_2 \xRightarrow{\overline{\gamma}} q_2'$$

Fig. 3. The runs defined by $\pi \in \mathcal{H}_\kappa(L_1, L_2)$ where $\gamma\alpha$ is the minimal gamma-alpha-prefix and, therefore, $f_1 \in \mathcal{F}_1$ or $f_2 \in \mathcal{F}_2$.

In other words, among all factorizations of π which satisfy the first and second condition, we choose the factorization where the length of γ is minimal. In such a factorization we call $\gamma\alpha \leqslant \pi$ the *minimal gamma-alpha-prefix* of π. This factorization yields runs in the DFAs $\mathcal{A}_1$ and $\mathcal{A}_2$ as in Fig. 3. (Recall that $\mathcal{A}_2$ accepts $\overline{L_2}$ and $\overline{\pi} = \gamma\alpha\overline{\beta}\overline{\alpha}\overline{\gamma}$.) As π determines the factors γ and α, the states c_i, d_i, e_i, f_i, and q_i' (for $i = 1, 2$) are determined by π as well.

Vice versa, paths of the form as shown in Fig. 3 where $|\alpha| = \kappa$ define one word $\pi = \gamma\alpha\beta\overline{\alpha}\overline{\gamma}$ from the hairpin completion $\mathcal{H}_\kappa(L_1, L_2)$ such that $\gamma\alpha$ is its minimal gamma-alpha-prefix.

By this observation, we can use quadruples of states in order to define the unambiguous linear grammar $G = G_\kappa(L_1 L_2)$ that generates the hairpin completion $\mathcal{H}_\kappa(L_1, L_2)$. The set of terminal symbols is Σ. For each $(p_1, p_2, q_1, q_2) \in \mathcal{Q}_1 \times \mathcal{Q}_2 \times \mathcal{Q}_1 \times \mathcal{Q}_2$ we define two non-terminal symbols of G. The first one is denoted by $B(p_1, p_2, q_1, q_2)$. It is defined as the regular language

$$B(p_1, p_2, q_1, q_2) = \left\{ w \in \Sigma^* \;\middle|\; p_1 \cdot w = q_1 \wedge p_2 \cdot \overline{w} = q_2 \right\}.$$

The second one is denoted by $R(p_1, p_2, q_1, q_2)$, which is just a formal symbol. The role of the non-terminal $B(p_1, p_2, q_1, q_2)$ is to derive all words of the language $B(p_1, p_2, q_1, q_2)$. For example, in Fig. 3 we have $\pi \in B(q_{01}, q_{02}, q_1', q_2')$, $\alpha\beta\overline{\alpha} \in B(c_1, c_2, f_1, f_2)$, and $\beta \in B(d_1, d_2, e_1, e_2)$.

A tuple (p_1, p_2, q_1, q_2) with $B(p_1, p_2, q_1, q_2) \neq \emptyset$ is called a *basic bridge*. This notation will be used in Section 4.1, too. Compared to Fig. 3, we intend that $B(d_1, d_2, e_1, e_2) \overset{*}{\underset{G}{\Longrightarrow}} \beta$. In order to achieve this, for all $p_1, q_1 \in \mathcal{Q}_1$, $p_2, q_2 \in \mathcal{Q}_2$, and $a \in \Sigma$ we define a first set of productions by the following right-linear rules, which we call B-rules:

$$B(p_1, p_2, q_1, q_2 \cdot \bar{a}) \to aB(p_1 \cdot a, p_2, q_1, q_2),$$
$$B(p_1, p_2, p_1, p_2) \to 1 \quad (= \text{empty word}).$$

Every derivation from $B(p_1, p_2, q_1, q_2)$ to a terminal word must use the rule $B(q_1, p_2, q_1, p_2) \to 1$ as the final step. Thus, $B(p_1, p_2, q_1, q_2) \overset{*}{\underset{G}{\Longrightarrow}} w$ implies $p_1 \cdot w = q_1$ and $p_2 \cdot \overline{w} = q_2$ as desired. Furthermore, for all words $w \in B(p_1, p_2, q_1, q_2)$ and all factorizations $w = uv$ (i.e., $(p_1 \cdot u) \cdot v = q_1$ and $(p_2 \cdot \bar{v}) \cdot \bar{u} = q_2$) there is a derivation

$$B(p_1, p_2, q_1, q_2) \overset{*}{\underset{G}{\Longrightarrow}} uB(p_1 \cdot u, p_2, q_2, p_2 \cdot \bar{v}) \overset{*}{\underset{G}{\Longrightarrow}} uvB(q_1, p_2, q_1, p_2) \underset{G}{\Longrightarrow} uv$$

where the non-terminal reached after $|u|$ steps is uniquely determined. We conclude, the non-terminal $B(p_1, p_2, q_1, q_2)$ derives all words from the language $B(p_1, p_2, q_1, q_2)$ and the derivation of each word is unambiguous.

The other rules of the grammar G are containing a symbol of the form $R(p_1, p_2, q_1, q_2)$. For each $(p_1, p_2, q_1, q_2) \in \mathcal{Q}_1 \times \mathcal{Q}_2 \times \mathcal{Q}_1 \times \mathcal{Q}_2$ and letter $a \in \Sigma$ such that $q_1 \cdot \bar{a} \notin \mathcal{F}_1$ and $q_2 \cdot \bar{a} \notin \mathcal{F}_2$ we define an RR-rule:

$$R(p_1, p_2, q_1 \cdot \bar{a}, q_2 \cdot \bar{a}) \to aR(p_1 \cdot a, p_2 \cdot a, q_1, q_2)\bar{a}.$$

For each $p_1, q_1 \in \mathcal{Q}_1$, $p_2, q_2 \in \mathcal{Q}_2$ and $\alpha \in \Sigma^\kappa$ such that $q_1 \cdot \overline{\alpha} \in \mathcal{F}_1$ or $q_2 \cdot \overline{\alpha} \in \mathcal{F}_2$ we define an RB-rule:

$$R(p_1, p_2, q_1 \cdot \overline{\alpha}, q_2 \cdot \overline{\alpha}) \to \alpha B(p_1 \cdot \alpha, p_2 \cdot \alpha, q_1, q_2)\overline{\alpha}.$$

Every symbol $R(p_1, p_2, q_1', q_2')$ which appears on the left-hand side of an RR-rule does not appear as a left-hand side of an RB-rule; and vice versa. Moreover the derivations using RR- and RB-rules are again unambiguous. To see this, consider a derivation

$$R\big(q_{01}, q_{02}, q_1', q_2'\big) \overset{*}{\underset{G}{\Longrightarrow}} uR(p_1, p_2, q_1, q_2)\bar{u} \overset{*}{\underset{G}{\Longrightarrow}} uvR(c_1, c_2, f_1, f_2)\bar{v}\bar{u}.$$

The non-terminal $R(p_1, p_2, q_1, q_2)$ is determined by $p_i = q_{0i} \cdot u$ and $q_i = f_i \cdot \bar{v}$ (for $i = 1, 2$). Furthermore, the states q_1', q_2', q_1, and q_2 cannot be final states unless $v = 1$. If $f_1 \in \mathcal{F}_1$ or $f_2 \in \mathcal{F}_2$, then we have to use an RB-rule in the next derivation step. This is possible if and only if $f_1 = e_1 \cdot \overline{\alpha}$ and $f_2 = e_2 \cdot \overline{\alpha}$ for some $\alpha \in \Sigma^\kappa$. Then we can apply the RB-rule $R(c_1, c_2, f_1, f_2) \to \alpha B(c_1 \cdot \alpha, c_2 \cdot \alpha, e_1, e_2)\overline{\alpha}$.

We conclude that we have factorizations as in Fig. 3 if and only if

$$R\big(q_{01}, q_{02}, q_1', q_2'\big) \overset{*}{\underset{G}{\Longrightarrow}} \gamma R(c_1, c_2, f_1, f_2)\overline{\gamma} \underset{G}{\Longrightarrow} \gamma\alpha B(d_1, d_2, e_1, e_2)\overline{\alpha}\overline{\gamma} \overset{*}{\underset{G}{\Longrightarrow}} \gamma\alpha\beta\overline{\alpha}\overline{\gamma} = \pi.$$

Moreover, the derivation from $R(q_{01}, q_{02}, q_1', q_2')$ to π is unambiguous.

We now let $R(q_{01}, q_{02}, q'_1, q'_2)$ be the axioms in the grammar G for all $q'_1 \in \mathcal{Q}_1$ and $q'_2 \in \mathcal{Q}_2$. Thus, we let

$$\mathcal{S} = \{R(q_{01}, q_{02}, q'_1, q'_2) \mid q'_1 \in \mathcal{Q}_1 \wedge q'_2 \in \mathcal{Q}_2\}.$$

This concludes the definition of the grammar G. We still have to show that different axioms generate disjoint languages. Consider a word π. There exists at most one axiom with $R(q_{01}, q_{02}, q'_1, q'_2) \xRightarrow[G]{*} \pi$, as we have $q'_1 = q_{01} \cdot \pi$ and $q'_2 = q_{02} \cdot \overline{\pi}$, hence G is unambiguous linear.

According to our convention the size of the grammar is at most

$$(2\kappa + 10)|\Sigma|^\kappa n_1^2 n_2^2 \in \mathcal{O}(n_1^2 n_2^2) \subseteq \mathcal{O}(n^4). \qquad \square$$

4. Main result

The purpose of this section is to prove the following theorem, where Σ and κ are viewed as constants; therefore, they are not part of the input.

Theorem 4.1. *Consider the following decision problem whether the hairpin completion $\mathcal{H}_\kappa(L_1, L_2)$ is regular:*

Input. *DFAs $\mathcal{A}_1$ and $\mathcal{A}_2$ with state sets $\mathcal{Q}_1$ and $\mathcal{Q}_2$ accepting the languages L_1 and $\overline{L_2}$, respectively. The input size is $n = \max\{|\mathcal{Q}_1|, |\mathcal{Q}_2|\}$.*
Question. *Is the hairpin completion $\mathcal{H}_\kappa(L_1, L_2)$ regular?*

The problem is decidable in time:

1. *$\mathcal{O}(n^2)$, if L_1 or L_2 is finite.*
2. *$\mathcal{O}(n^6)$, if $L_1 = \overline{L_2}$.*
3. *$\mathcal{O}(n^8)$, in general.*

The proof of Theorem 4.1 covers the rest of this section. Proposition 4.11 in Subsection 4.2 yields the theorem, if L_1 or L_2 is finite. In particular, it covers the case that $L_1 = \emptyset$ or $L_2 = \emptyset$. The demanding situation is to show the theorem when L_1 and L_2 are infinite. The proof of the theorem is then based on Tests 1, 2, and 3. All tests check properties of an automaton $\mathcal{A}$, which accepts the minimal gamma-alpha-prefixes, introduced in Section 3. Therefore, we start with the construction of $\mathcal{A}$.

4.1. The Automaton $\mathcal{A}$

Our goal is to construct a non-deterministic automaton $\mathcal{A}$ which accepts the minimal gamma-alpha-prefix of certain words $\pi = \gamma\alpha\beta\overline{\alpha}\overline{\gamma}$. The construction is analogous to the definition of rules for the non-terminals $R(p_1, p_2, q_1, q_2)$ in Section 3. In particular, the final states of the automaton will determine a basic bridge (d_1, d_2, e_1, e_2) such that $\beta \in B(d_1, d_2, e_1, e_2)$.

We use the product automaton of $\mathcal{A}_1$ and $\mathcal{A}_2$ where all states are reachable. Thus, the state set is:

$$\mathcal{Q}_{12} = \{(p_1, p_2) \in \mathcal{Q}_1 \times \mathcal{Q}_2 \mid \exists w \in \Sigma^*\colon q_{01} \cdot w = p_1 \wedge q_{02} \cdot w = p_2\}.$$

The transition function is given by $(p_1, p_2) \cdot w = (p_1 \cdot w, p_2 \cdot w)$ for $(p_1, p_2) \in \mathcal{Q}_{12}$ and $w \in \Sigma^*$. Furthermore, we let $n_{12} = |\mathcal{Q}_{12}|$. If $L_2 = \emptyset$ or $L_1 = \overline{L_2}$, then $n_{12} = n_1 = n$ and in general $n \leqslant n_{12} \leqslant n^2$.

The state set of $\mathcal{A}$ will use several copies of subsets of $\mathcal{Q}_{12} \times \mathcal{Q}_1 \times \mathcal{Q}_2$ which in turn is a subset of $\mathcal{Q}_1 \times \mathcal{Q}_2 \times \mathcal{Q}_1 \times \mathcal{Q}_2$. According to Section 3 we call (p_1, p_2, q_1, q_2) a *basic bridge* if $B(p_1, p_2, q_1, q_2) \neq \emptyset$. If (p_1, p_2, q_1, q_2) is a basic bridge, then there is some word β such that $p_1 \cdot \beta = q_1$ and $p_2 \cdot \overline{\beta} = q_2$. In order to accept the α-part, we use $\kappa + 1$ *levels*. By $[\kappa]$ we denote the set $[\kappa] = \{0, \ldots, \kappa\}$. We define the state space of $\mathcal{A}$ by

$$\{((p_1, p_2), q_1, q_2, \ell) \in \mathcal{Q}_{12} \times \mathcal{Q}_1 \times \mathcal{Q}_2 \times [\kappa] \mid (p_1, p_2, q_1, q_2) \text{ is a basic bridge}\}.$$

In the following N denotes the number $N = n_{12} n_1 n_2 \leqslant n^4$. The number of states of $\mathcal{A}$ is therefore bounded by $N \cdot (\kappa + 1) \in \mathcal{O}(N) \subseteq \mathcal{O}(n^4)$. For $L_1 = \emptyset$ or $L_2 = \emptyset$ we have $N \leqslant n^2$; for $L_2 = \overline{L_1}$ we have $N \leqslant n^3$.

Generalizing the notion of basic bridges we call a state $((p_1, p_2), q_1, q_2, \ell)$ a *bridge*. Bridges are frequently denoted by (P, q_1, q_2, ℓ) with $P = (p_1, p_2) \in \mathcal{Q}_{12}$, $q_1 \in \mathcal{Q}_1$, $q_2 \in \mathcal{Q}_2$, and $\ell \in [\kappa]$. Bridges are a central concept in the following.

The a-transitions in the NFA for $a \in \Sigma$ are given by the following arcs:

$$\begin{aligned}
&(P, q_1 \cdot \bar{a}, q_2 \cdot \bar{a}, 0) \xrightarrow{a} (P \cdot a, q_1, q_2, 0) && \text{for } q_i \cdot \bar{a} \notin \mathcal{F}_i,\ i = 1, 2,\\
&(P, q_1 \cdot \bar{a}, q_2 \cdot \bar{a}, 0) \xrightarrow{a} (P \cdot a, q_1, q_2, 1) && \text{for } q_1 \cdot \bar{a} \in \mathcal{F}_1 \text{or } q_2 \cdot \bar{a} \in \mathcal{F}_2,\\
&(P, q_1 \cdot \bar{a}, q_2 \cdot \bar{a}, \ell) \xrightarrow{a} (P \cdot a, q_1, q_2, \ell + 1) && \text{for } 1 \leqslant \ell < \kappa.
\end{aligned}$$

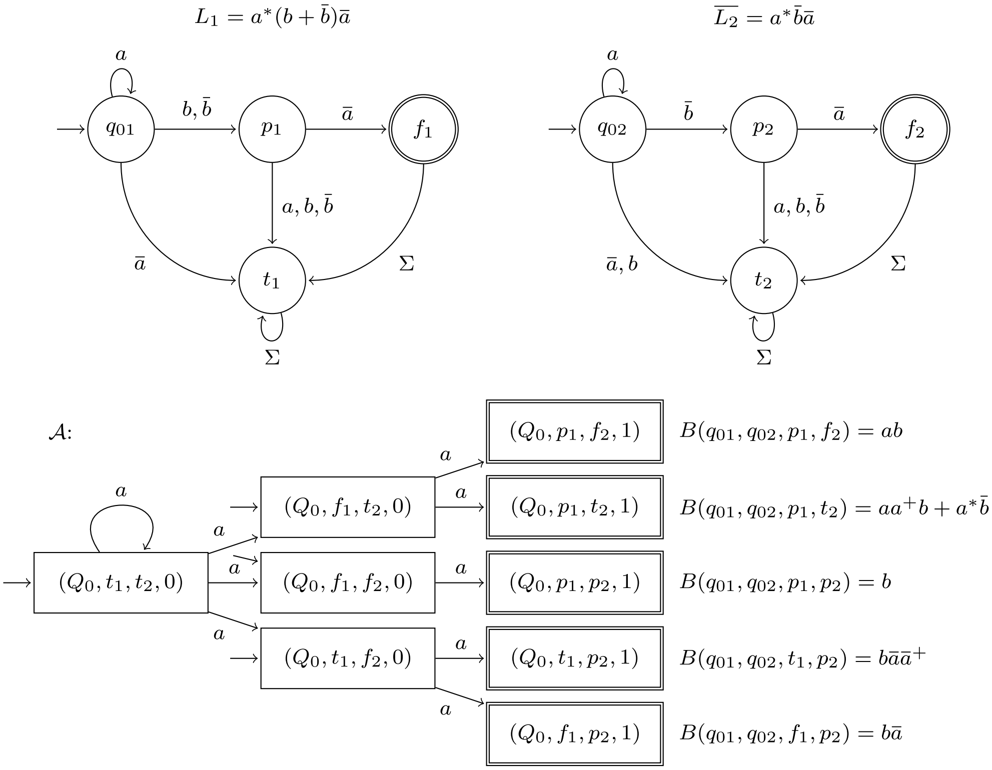

Fig. 4. DFAs for L_1 and $\overline{L_2}$ and the resulting NFA $\mathcal{A}$ associated to the hairpin completion $\mathcal{H}_\kappa(L_1, L_2)$ as defined in Example 4.3. Note that the NFA $\mathcal{A}$ is reduced such that every state is reachable from an initial state and leads to some final state.

Observe that no state of the form $(P, q_1, q_2, 0)$ with $q_1 \in \mathcal{F}_1$ or $q_2 \in \mathcal{F}_2$ has an outgoing arc to level zero; we must switch to level one. There are no outgoing arcs on level κ and for each $(a, P, q_1, q_2, \ell) \in \Sigma \times \mathcal{Q}_{12} \times \mathcal{Q}_1 \times \mathcal{Q}_2 \times [\kappa - 1]$ there exists at most one arc $(P, q_1', q_2', \ell) \xrightarrow{a} (P \cdot a, q_1, q_2, \ell')$. Indeed, the triple (q_1', q_2', ℓ') is determined by (q_1, q_2, ℓ) and the letter a. More precisely, $q_i' = q_i \cdot \bar{a}$ for $i = 1, 2$. Moreover, $\ell' = 0$ if $\ell = 0$ and $q_1' \notin \mathcal{F}_1$ and $q_2' \notin \mathcal{F}_2$; otherwise, $\ell' = \ell + 1$. Not all arcs exist because it is possible that (P, q_1', q_2', ℓ) is a bridge whereas $(P \cdot a, q_1, q_2, \ell')$ is not. Thus, there are at most $|\Sigma| \cdot N \cdot \kappa \in \mathcal{O}(N)$ arcs in the NFA. For later reference, let us state a remark:

Remark 4.2. The number of states plus the number of arcs in the automaton $\mathcal{A}$ is bounded by $\mathcal{O}(N)$.

The set of initial states $\mathcal{I}$ contains all bridges of the form $(Q_0, q_1', q_2', 0)$ where $Q_0 = (q_{01}, q_{02})$. The set of final states $\mathcal{F}$ is given by all bridges (P, q_1, q_2, κ) on level κ.

The underlying graph of the automaton restricted to levels $\ell \geqslant 1$ is a directed acyclic graph, since in each step we have to climb by one level, but it doesn't necessarily have to be a forest.

Example 4.3. Let $L_1 = a^*(b + \bar{b})\bar{a}$ and $L_2 = ab\bar{a}^*$. For $\kappa = 1$ we obtain

$$\mathcal{H}_\kappa(L_1, L_2) = a^+b\bar{a}^+ \cup \left\{a^i\bar{b}\bar{a}^j \;\middle|\; i \geqslant j \geqslant 1\right\}.$$

The hairpin completion is linear context-free, but not regular. For a graphical presentation of the resulting NFA $\mathcal{A}$ with 4 initial states and 5 final states, see Fig. 4.

Next, we show that the automaton $\mathcal{A}$ encodes the minimal gamma-alpha-prefixes and that we obtain the hairpin completion $\mathcal{H}_\kappa(L_1, L_2)$ in a natural way from $\mathcal{A}$. For languages B and R we denote by B^R the language

$$B^R = \{v\beta\bar{v} \mid \beta \in B \wedge v \in R\}.$$

The notation B^R is adopted from group theory where exponentiation denotes conjugation and the involution refers to taking inverses. Clearly, if B and R are regular, then B^R is linear context-free, but not regular in general, e.g., consider $\{1\}^{\Sigma^*}$. On the other hand, if B is regular and R is finite, then B^R is regular.

Lemma 4.4. *Let $M = \mathcal{I} \times \mathcal{F}$. For each pair $\mu = (I, F) \in M$ with $F = ((d_1, d_2), e_1, e_2, \kappa)$ let R_μ be the (regular) set of words which label a path from the initial state I to the final state F, and let $B_\mu = B(d_1, d_2, e_1, e_2)$. Then, the following assertions hold.*

The hairpin completion $\mathcal{H}_\kappa(L_1, L_2)$ is the disjoint union

$$\mathcal{H}_\kappa(L_1, L_2) = \bigcup_{\mu \in M} B_\mu^{R_\mu}.$$

Moreover, for $\mu \in \mathcal{I} \times \mathcal{F}$ and for all words $\beta \in B_\mu$ and $v \in R_\mu$, the minimal gamma-alpha-prefix of $v\beta\bar{v}$ is v.

Proof. Let $\pi \in \mathcal{H}_\kappa(L_1, L_2)$. Let $\gamma\alpha$ be the minimal gamma-alpha-prefix of π with $|\alpha| = \kappa$ and factorize $\pi = \gamma\alpha\beta\bar{\alpha}\bar{\gamma}$. There are runs in the DFAs

$$\mathcal{A}_1: \quad q_{01} \xrightarrow{\gamma} c_1 \xrightarrow{\alpha} d_1 \xrightarrow{\beta} e_1 \xrightarrow{\bar{\alpha}} f_1 \xRightarrow{\bar{\gamma}} q_1',$$

$$\mathcal{A}_2: \quad q_{02} \xrightarrow{\gamma} c_2 \xrightarrow{\alpha} d_2 \xrightarrow{\bar{\beta}} e_2 \xrightarrow{\bar{\alpha}} f_2 \xRightarrow{\bar{\gamma}} q_2'$$

with $f_1 \in \mathcal{F}_1$ or $f_2 \in \mathcal{F}_2$; just as in Fig. 3. Recall that all states on these paths are determined by π. By the definition of the NFA $\mathcal{A}$, we find a path $I \xrightarrow{\gamma} A \xrightarrow{\alpha} F$ where $I = (Q_0, q_1', q_2', 0)$, $A = ((c_1, c_2), f_1, f_2, 0)$, and $F = ((d_1, d_2), e_1, e_2, \kappa)$. As $\beta \in B(d_1, d_2, e_1, e_2)$, there is a unique $\mu = (I, F) \in \mathcal{I} \times \mathcal{F}$ with $\pi \in B_\mu^{R_\mu}$.

Conversely, let $\mu = (I, F) \in \mathcal{I} \times \mathcal{F}$, let $\beta \in B_\mu$, and let $I \xrightarrow{\gamma} A \xrightarrow{\alpha} F$ with $|\alpha| = \kappa$ be a path in $\mathcal{A}$. As F is a final state it is on level κ and $A = ((c_1, c_2), f_1, f_2, 0)$ is the last state on level zero, whence $f_1 \in \mathcal{F}_1$ or $f_2 \in \mathcal{F}_2$. Therefore, we find runs in the DFAs just like above where $I = (Q_0, q_1', q_2', 0)$ and $F = ((d_1, d_2), e_1, e_2, \kappa)$. We conclude $\gamma\alpha$ is the minimal gamma-prefix of $\gamma\alpha\beta\bar{\alpha}\bar{\gamma}$ and $\gamma\alpha\beta\bar{\alpha}\bar{\gamma} \in \mathcal{H}_\kappa(L_1, L_2)$. □

The next lemma tells us that the paths in the automaton are unambiguous. The arguments are essentially the same as used in Section 3. The unambiguity of paths will become crucial later.

Lemma 4.5. *Let $w \in \Sigma^*$ be the label of a path in $\mathcal{A}$ from a bridge $A = (P, p_1, p_2, \ell)$ to $A' = (P', p_1', p_2', \ell')$, then the path is unique. This means that $B = B'$ whenever $w = uv$ and*

$$A \xrightarrow{u} B \xrightarrow{v} A', \qquad A \xrightarrow{u} B' \xrightarrow{v} A'.$$

Proof. It is enough to consider $u = a \in \Sigma$. Let $B = (Q, q_1, q_2, m)$. Then we have $Q = P \cdot a$ and $q_i = p_i' \cdot \bar{v}$. If $\ell = 0$ and $p_i \notin \mathcal{F}_i$ for $i = 1, 2$, then $m = 0$, too; otherwise $m = \ell + 1$. Thus, B is determined by A, A', and u, v. We conclude $B = B'$. □

Next, we show that the automaton $\mathcal{A}$ can be constructed in time $\mathcal{O}(n_1^2 n_2^2)$. By Remark 4.2 the number of states plus the number of transitions in $\mathcal{A}$ is in $\mathcal{O}(N) \subseteq \mathcal{O}(n_1^2 n_2^2)$. Thus, it suffices to prove that we can compute the set of basic bridges in time $\mathcal{O}(n_1^2 n_2^2)$. This is shown in the next lemma. Lemma 4.6 is actually more general for later use, where we need to control the first letters of words in $B(d_1, d_2, e_1, e_2)$. A basic bridge (d_1, d_2, e_1, e_2) is called an *a-bridge*, if $B(d_1, d_2, e_1, e_2) \cap a\Sigma^* \neq \emptyset$.

Lemma 4.6. *The set containing all basic bridges and the sets containing all a-bridges for $a \in \Sigma$, respectively, can be computed in time $\mathcal{O}(n_1^2 n_2^2)$.*

Proof. All tuples $(p_1, p_2, p_1, p_2) \in \mathcal{Q}_1 \times \mathcal{Q}_2 \times \mathcal{Q}_1 \times \mathcal{Q}_2$ are marked as basic bridges since the empty word is a witness for $B(p_1, p_2, p_1, p_2) \neq \emptyset$. It is therefore enough to compute the set of a-bridges in time $\mathcal{O}(n_1^2 n_2^2)$. Consider the transition system with state set $\mathcal{Q}_1 \times \mathcal{Q}_2$ and transitions $(p_1, q_2) \xrightarrow{a} (q_1, p_2)$ for all $p_1 \cdot a = q_1$ and $p_2 \cdot \bar{a} = q_2$. Hence, we use forward edges from $\mathcal{A}_1$ and backward edges from $\mathcal{A}_2$. There are $n_1 n_2 \cdot |\Sigma|$ transitions and the transition system can be constructed in $\mathcal{O}(n_1 n_2)$.

There is a path in the transition system $(p_1, q_2) \xrightarrow{w} (q_1, p_2)$ with $w \in \Sigma^*$ if and only if we have $p_1 \cdot w = q_1$ and $p_2 \cdot \overline{w} = q_2$. Thus, a tuple $(p_1, p_2, q_1, q_2) \in \mathcal{Q}_1 \times \mathcal{Q}_2 \times \mathcal{Q}_1 \times \mathcal{Q}_2$ is an a-bridge if and only if there exists such a path that starts with an a-transition.

We now run a depth-first reachability search for all triples $(p_1, q_2, a) \in \mathcal{Q}_1 \times \mathcal{Q}_2 \times \Sigma$. If during the depth-first search algorithm we meet a pair $(q_1, p_2) \in \mathcal{Q}_1 \times \mathcal{Q}_2$ which is reachable from (p_1, q_2) by a path in the transition system starting with an a-transition, then we mark (p_1, p_2, q_1, q_2) as a basic bridge and as an a-bridge. Since each depth-first search can be performed in $\mathcal{O}(n_1 n_2)$, the whole computation can be done in $\mathcal{O}(n_1^2 n_2^2)$. □

Remark 4.7. For convenience, we will henceforth assume that all states in the automaton $\mathcal{A}$ are reachable from an initial state and lead to some final state. Such a reachability test can easily be performed in $\mathcal{O}(n_1^2 n_2^2)$; thus, this will not breach the time bounds.

4.2. Test 0

In the case where L_1 or L_2 is finite, we provide a simple necessary and sufficient condition for the regularity of $\mathcal{H}_\kappa(L_1, L_2)$. First, we make the following observation.

Lemma 4.8. *Let L_1 be finite, then the hairpin completion $\mathcal{H}_\kappa(L_1, L_2)$ is regular if and only if the hairpin completion $\mathcal{H}_\kappa(\emptyset, L_2)$ is regular.*

Proof. We have $\mathcal{H}_\kappa(L_1, L_2) = \mathcal{H}_\kappa(L_1, \emptyset) \cup \mathcal{H}_\kappa(\emptyset, L_2)$. Since L_1 is finite, $\mathcal{H}_\kappa(L_1, \emptyset)$ is finite, too. The result follows since regular languages are closed under finite variation. $\square$

Proposition 4.9. *Let $\mathcal{A}$ be the automaton constructed above.*

1. *If the language $L(\mathcal{A})$ is finite, then $\mathcal{H}_\kappa(L_1, L_2)$ is regular.*
2. *If the language $L(\mathcal{A})$ is infinite and either L_1 is finite or L_2 is finite, then $\mathcal{H}_\kappa(L_1, L_2)$ is not regular.*

Proof. Statement 1 follows directly from Lemma 4.4. For Statement 2 let $L(\mathcal{A})$ be infinite. There is a path

$$I \xrightarrow{u} A \xrightarrow{v} A \xrightarrow{w} F$$

in $\mathcal{A}$ where I is an initial bridge, $F = ((d_1, d_2), e_1, e_2)$ is a final bridge, and $A \xrightarrow{v} A$ is a loop with $v \neq 1$, i.e., the loop is non-trivial. Note that A is on level 0 and hence $|w| \geqslant \kappa$. Let α be the suffix of w of length κ and let β be a word from the language $B(d_1, d_2, e_1, e_2)$. We have $\pi_i = uv^i w\beta\overline{w}\overline{v}^i\overline{u} \in \mathcal{H}_\kappa(L_1, L_2)$ for all $i \geqslant 0$. Moreover, if a prefix of π_i belongs to L_1, then it is a prefix of $uv^i w\beta\overline{\alpha}$; and if a suffix of π_i belongs to L_2, then it is a suffix of $\alpha\beta\overline{w}\overline{v}^i\overline{u}$.

To achieve a contradiction, assume $\mathcal{H}_\kappa(L_1, L_2)$ is regular and either L_1 or L_2 is finite. By symmetry and by Lemma 4.8, it is enough to consider the case when L_1 is empty. Let $j \geqslant 1$ such that the power v^j is idempotent in the syntactic monoid of $\mathcal{H}_\kappa(\emptyset, L_2)$. This means that for all words x, $y \in \Sigma^*$ we have $xv^j y \in \mathcal{H}_\kappa(\emptyset, L_2)$ if and only if $xv^{2j} y \in \mathcal{H}_\kappa(\emptyset, L_2)$. As a consequence for all $k \geqslant 1$ we obtain

$$uv^{jk} w\beta\overline{w}\overline{v}^j\overline{u} \in \mathcal{H}_\kappa(L_1, L_2).$$

We now consider k to be large enough such that v^{jk} is longer than half of the length of $\pi = uv^{jk} w\beta\overline{w}\overline{v}^j\overline{u}$. Since $\pi \in \mathcal{H}_\kappa(\emptyset, L_2)$, we must find $\pi = \gamma'\alpha'\beta'\overline{\alpha'}\overline{\gamma'}$ with $\alpha'\beta'\overline{\alpha'}\overline{\gamma'} \in L_2$. However, the longest suffix of π that belongs to L_2 is still a suffix of $\alpha\beta\overline{w}\overline{v}^j\overline{u}$, because the word v yields a loop around the state A, so we are free to pump with v. This is a subtle point and refers to the translation of Fig. 3 into the runs of $\mathcal{A}$. Now, we have the following contradiction: $1/2 \cdot |\pi| < |\alpha'\beta'\overline{\alpha'}\overline{\gamma'}| \leqslant |\alpha\beta\overline{w}\overline{v}^j\overline{u}| \leqslant 1/2 \cdot |\pi|$. $\square$

Test 0. Decide whether or not $L(\mathcal{A})$ is finite. If it is finite, then stop with the output that $\mathcal{H}_\kappa(L_1, L_2)$ is regular. If it is not finite but L_1 or L_2 is finite, then stop with the output that $\mathcal{H}_\kappa(L_1, L_2)$ is not regular.

Strictly speaking, Test 0 is redundant for the general case, but it yields the desired time complexity for $L_1 = \emptyset$ or $L_2 = \emptyset$, because in these cases we have $n_1 = 1$ or $n_2 = 1$, respectively.

Lemma 4.10. *Test 0 can be performed in time $\mathcal{O}(n_1^2 n_2^2)$.*

Proof. Recall that every state in $\mathcal{A}$ is reachable and co-reachable, by Remark 4.7. The language $L(\mathcal{A})$ is infinite if and only if $\mathcal{A}$ contains at least one non-trivial loop $A \xrightarrow{v} A$. By Tarjan's algorithm [30] we can decompose a directed graph (as well as a finite automaton) into its strongly connected components in linear time with respect to the number of transitions. As the automaton $\mathcal{A}$ has $\mathcal{O}(n_1^2 n_2^2)$ transitions, this yields the time complexity. $\square$

We are now ready to prove Theorem 4.1, 1 by the following proposition.

Proposition 4.11. *If L_1 or L_2 is finite, then the regularity of $\mathcal{H}_\kappa(L_1, L_2)$ can be decided in time $\mathcal{O}(n^2)$.*

Proof. Given an NFA we can decide emptiness of the accepted language in linear time by Tarjan's algorithm [30]. This is a better time complexity than needed here. Say L_2 is finite, then we replace the automaton $\mathcal{A}_2$ by a single state automaton and we can start our procedure with the value $n_2 = 1$. The result follows by Lemma 4.10. $\square$

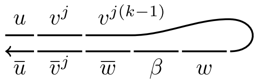

Fig. 5. The hairpin of π (read the upper part from left to right and the lower part from right to left).

4.3. Test 1

By Test 0, we may assume in the following that $\mathcal{A}$ accepts an infinite language and that the set S of non-trivial strongly connected components of the automaton $\mathcal{A}$ has been computed. Every non-trivial strongly connected component is on level 0 and, moreover, as $\mathcal{A}$ accepts an infinite language, there is at least one. For $s \in S$ let N_s be the number of states in the component s. Note that $\sum_{s\in S} N_s \leqslant N$. By putting some linear order on the set of bridges, we assign to each $s \in S$ the least bridge A_s and some shortest, non-empty word v_s such that $A_s \xrightarrow{v_s} A_s$.

The next lemma tells us that for a regular hairpin completion $\mathcal{H}_\kappa(L_1, L_2)$ every strongly connected component $s \in S$ is a simple cycle, and hence, the word v_s is uniquely defined.

Lemma 4.12. *Let the hairpin completion $\mathcal{H}_\kappa(L_1, L_2)$ be regular, $s \in S$ be a strongly connected component, and $A_s \xrightarrow{w} F$ be a path from A_s to a final bridge F. Then the word w is a prefix of some word in v_s^+.*

In addition, the word v_s is uniquely defined and the loop $A_s \xrightarrow{v_s} A_s$ visits every other bridge $B \in s \setminus \{A\}$ exactly once. Thus, it forms a Hamiltonian cycle of s and $|v_s| = N_s$.

Proof. Let $A = A_s$ and $v = v_s$. Consider a path labeled by w from A to a final bridge $F = ((d_1, d_2), e_1, e_2, k)$. As all bridges are reachable, we find a word u and an initial bridge I such that

$$I \xrightarrow{u} A \xrightarrow{v} A \xrightarrow{w} F.$$

As the automaton $\mathcal{A}$ accepts uv^iw for all $i \geqslant 0$, we see that $uv^iw\beta\overline{w}\overline{v}^i\overline{u} \in \mathcal{H}_\kappa(L_1, L_2)$ for all $i \geqslant 0$ and all $\beta \in B(d_1, d_2, e_1, e_2)$. As $\mathcal{H}_\kappa(L_1, L_2)$ is regular, there are $j \geqslant 1$ and $k > |w\beta|$ such that $uv^{jk}w\beta\overline{w}\overline{v}^j\overline{u} \in \mathcal{H}_\kappa(L_1, L_2)$, by pumping. Due to the definition of $\mathcal{A}$, the longest suffix of π belonging to L_2 is a suffix of $\alpha\beta\overline{w}\overline{v}^j\overline{u}$, where α is the suffix of w of length κ, and this suffix is too short to create the hairpin completion. This means that the hairpin completion is forced to use a prefix in L_1 which has to be a prefix of $uv^{jk}w\beta\overline{\alpha}$. Therefore, the suffix $\overline{w}\overline{v}^j\overline{u}$ is complementary to a prefix of uv^{jk}, whence w must be a prefix of $v^{j(k-1)}$; see Fig. 5. We conclude the first statement of our lemma.

Recall that $A \xrightarrow{v} A$ is a shortest, non-trivial loop around A; hence $|v| \leqslant N_s$ is obvious. Let $B \in s \setminus \{A\}$ and $x = x_1x_2$ such that $A \xrightarrow{x_1} B \xrightarrow{x_2} A$. For some $i, j \geqslant 1$ we have $|v^i| = |x^j|$. Thus, $v^i = x^j$ by the first statement. By the unique-path-property stated in Lemma 4.5 we obtain that the loop $A \xrightarrow{x^j} A$ just uses the shortest loop $A \xrightarrow{v} A$ several times. In particular, B is on the shortest loop around A. This yields $|v| \geqslant N_s$ and hence the second statement. $\square$

Example 4.13. In the above example (Example 4.3 and Fig. 4) the state $(Q_0, t_1, t_2, 0)$ forms the only strongly connected component and the corresponding path is labeled with a. As one can easily observe, the automaton $\mathcal{A}$ satisfies the properties stated in Lemma 4.12, even though the hairpin completion is not regular.

The next test tries to falsify the property of Lemma 4.12. Hence it gives a sufficient condition that $\mathcal{H}_\kappa(L_1, L_2)$ is not regular.

Test 1. Decide whether there is $s \in S$ and a path $A_s \xrightarrow{w} F$ such that w is not a prefix of a word in v_s^+. If there is such a path, then stop with the output that $\mathcal{H}_\kappa(L_1, L_2)$ is not regular.

Lemma 4.14. *Test 1 can be performed in time $\mathcal{O}(N^2)$.*

Proof. For $s \in S$, let $A = A_s$ and compute a shortest non-empty word v such that $A \xrightarrow{v} A$. If $|v| \neq N_s$, stop with the output that $\mathcal{H}_\kappa(L_1, L_2)$ is not regular. Otherwise, assign to each bridge that is reachable from A a subset of marks from $\{0, \ldots, N_s - 1\}$. A mark i is assigned to a bridge B if B is reachable from A with a word from $v^*v[1, i]$. Test 1 yields that $\mathcal{H}_\kappa(L_1, L_2)$ is not regular if and only if there is a bridge that is marked by i and that has an outgoing a-transition where $a \neq v[i+1]$. The marking algorithm can be performed by a depth-first search that runs in time $\mathcal{O}(N \cdot N_s)$. Summing over all strongly connected components we deduce a time complexity in $\mathcal{O}(\sum_{s\in S} N \cdot N_s) \subseteq \mathcal{O}(N^2)$. $\square$

$$\begin{array}{ll} L_1: & q_{01} \xrightarrow{u} p_1 \xrightarrow{v^k} p_1 \xrightarrow{xy} c_1 \xrightarrow{z} d_1 \xrightarrow{\bar{x}} e_1 \stackrel{\bar{v}^{n_1}}{\Longrightarrow} q_1 \stackrel{\bar{v}^*}{\Longrightarrow} q_1 \stackrel{\bar{u}}{\Longrightarrow} q'_1 \\ \overline{L_2}: & q_{02} \xrightarrow{u} p_2 \xrightarrow{v^\ell} p_2 \xrightarrow{x} c_2 \xrightarrow{\bar{z}} d_2 \xrightarrow{\bar{y}\bar{x}} e_2 \stackrel{\bar{v}^{n_2}}{\Longrightarrow} q_2 \stackrel{\bar{v}^*}{\Longrightarrow} q_2 \stackrel{\bar{u}}{\Longrightarrow} q'_2 \end{array}$$

Fig. 6. Runs through $\mathcal{A}_1$ and $\mathcal{A}_2$ based on the loop $A \xrightarrow{v} A$.

4.4. Tests 2 and 3

Henceforth, we assume that Test 1 was successful (i.e., Test 1 did not yield that $\mathcal{H}_\kappa(L_1, L_2)$ is not regular). We fix a strongly connected component $s \in S$ of $\mathcal{A}$. We let $A = A_s = ((p_1, p_2), q_1, q_2, 0)$, we let $v = v_s$, and we assume $A \xrightarrow{v} A$ forms a Hamiltonian cycle in s. By u we denote some word leading from an initial bridge $((q_{01}, q_{02}), q'_1, q'_2, 0)$ to A. For the following test we do not need to know u we just need to know it exists. The main idea is to investigate runs through the DFAs $\mathcal{A}_1$ and $\mathcal{A}_2$ where $k, \ell \geqslant n$ according to Fig. 6.

We investigate the case when $uv^kxyz\bar{x}\bar{v}^\ell\bar{u} \in \mathcal{H}_\kappa(L_1, L_2)$ for all $k \geqslant \ell$ and where (by symmetry) this property is due to the longest prefix belonging to L_1.

The following lemma is rather technical. However, the notations are chosen to fit exactly to Fig. 6.

Lemma 4.15. *Let $x, y, z \in \Sigma^*$ be words and $(d_1, d_2) \in \mathcal{Q}_1 \times \mathcal{Q}_2$ with the following properties:*

1. *$\kappa \leqslant |x| < |v| + \kappa$ and x is a prefix of some word in v^+.*
2. *$0 \leqslant |y| < |v|$ and xy is the longest common prefix of xyz and some word in v^+.*
3. *$z \in B(c_1, c_2, d_1, d_2)$, where $c_1 = p_1 \cdot xy$ and $c_2 = p_2 \cdot x$.*
4. *$q_1 = d_1 \cdot \bar{x}\bar{v}^{n_1}$ and during the computation of $d_1 \cdot \bar{x}\bar{v}^{n_1}$ we see after exactly κ steps a final state in $\mathcal{F}_1$ and then never again.*
5. *$q_2 = d_2 \cdot \bar{y}\bar{x}\bar{v}^{n_2}$ and, let $e_2 = d_2 \cdot \bar{y}\bar{x}$, during the computation of $e_2 \cdot \bar{v}^{n_2}$ we do not see a final state in $\mathcal{F}_2$.*

If $\mathcal{H}_\kappa(L_1, L_2)$ is regular, then there exists a factorization $xyz\bar{x}\bar{v} = \mu\delta\beta\bar{\delta}\overline{\mu}$ where $|\delta| = \kappa$ and $p_2 \cdot \mu\delta\overline{\beta}\bar{\delta} \in \mathcal{F}_2$ (which implies $\delta\beta\bar{\delta}\overline{\mu}\bar{v}^*\bar{u} \subseteq L_2$).

Proof. The conditions say that $uv^kxyz\bar{x}\bar{v}^\ell\bar{u} \in \mathcal{H}_\kappa(L_1, L_2)$ for all $k \geqslant \ell \geqslant n$. Moreover, by condition 4, the hairpin completion can be achieved with a prefix in L_1, and the longest prefix of $uv^kxyz\bar{x}\bar{v}^\ell\bar{u}$ belonging to L_1 is the prefix $uv^kxyz\overline{\alpha}$ where $\overline{\alpha}$ is the prefix of $\bar{x}$ of length κ.

If $\mathcal{H}_\kappa(L_1, L_2)$ is regular, then we have $uv^kxyz\bar{x}\bar{v}^{k+1}\bar{u} \in \mathcal{H}_\kappa(L_1, L_2)$, too, as soon as k is large enough, by a simple pumping argument. For this hairpin completion we must use a suffix belonging to L_2. For $z = 1$, this follows from $|y| < |v|$. For $z \neq 1$ we use $|y| < |v|$ and, in addition, that xya with $a = z[1]$ is not a prefix of vx by condition 2.

By 5 the longest suffix of $uv^kxyz\bar{x}\bar{v}^{k+1}\bar{u}$ belonging to L_2 is a suffix of $xyz\bar{x}\bar{v}^{k+1}\bar{u}$. Thus, we can write

$$uv^kxyz\bar{x}\bar{v}^{k+1}\bar{u} = uv^kxyz\bar{x}\bar{v}\bar{v}^k\bar{u} = uv^k\mu\delta\beta\bar{\delta}\overline{\mu}\bar{v}^k\bar{u}$$

where $\delta\beta\bar{\delta}\overline{\mu}\bar{v}^k\bar{u} \in L_2$ and $|\delta| = \kappa$. We obtain $xyz\bar{x}\bar{v} = \mu\delta\beta\bar{\delta}\overline{\mu}$. As $p_2 = q_{02} \cdot u$ and $p_2 = p_2 \cdot v$, we conclude $p_2 \cdot \mu\delta\overline{\beta}\bar{\delta} \in \mathcal{F}_2$ as desired. (Recall that our second DFA $\mathcal{A}_2$ accepts $\overline{L_2}$.) $\square$

Example 4.16. Let us take a look at Fig. 4 again. Let $A = (Q_0, t_1, t_2, 0)$, $v = a$ and $u = 1$. If we choose $x = a$, $y = 1$, $z = \bar{b}$, and $(d_1, d_2) = (p_1, p_2)$ we can see that conditions 1 to 5 of Lemma 4.15 are satisfied but there is no factorization $a\bar{b}\bar{a}\bar{a} = \mu\delta\beta\bar{\delta}\overline{\mu}$ with $|\delta| = \kappa = 1$ such that $q_{02} \cdot \mu\delta\overline{\beta}\bar{\delta} \notin \mathcal{F}_2$. Hence, the hairpin completion is not regular.

We perform Tests 2 and 3 which, again, try to falsify the property given by Lemma 4.15 for a regular hairpin completion. The tests distinguish whether the word z is empty or non-empty.

Test 2. Decide the existence of words $x, y \in \Sigma^*$ and states $(d_1, d_2) \in \mathcal{Q}_1 \times \mathcal{Q}_2$ satisfying conditions 1 to 5 of Lemma 4.15 with $z = 1$, but where for all factorizations $xy\bar{x}\bar{v} = \mu\delta\beta\bar{\delta}\overline{\mu}$ with $|\delta| = \kappa$ we have $p_2 \cdot \mu\delta\overline{\beta}\bar{\delta} \notin \mathcal{F}_2$. If we find such a situation, then stop with the output that $\mathcal{H}_\kappa(L_1, L_2)$ is not regular.

Test 3. Decide the existence of words $x, y, z \in \Sigma^*$ with $z \neq 1$ and states $(d_1, d_2) \in \mathcal{Q}_1 \times \mathcal{Q}_2$ satisfying conditions 1 to 5 of Lemma 4.15, but where for all factorizations $xyz\bar{x}\bar{v} = \mu\delta\beta\bar{\delta}\overline{\mu}$ with $|\delta| = \kappa$ we have $p_2 \cdot \mu\delta\overline{\beta}\bar{\delta} \notin \mathcal{F}_2$. If we find such a situation, then stop with the output that $\mathcal{H}_\kappa(L_1, L_2)$ is not regular.

Before we analyze the time complexity of Test 2 and Test 3 we will prove that if languages L_1 and L_2 pass the tests we described so far, then the hairpin completion $\mathcal{H}_\kappa(L_1, L_2)$ is regular. Thus, the properties given by Lemma 4.12 and Lemma 4.15 together are sufficient for the regularity of $\mathcal{H}_\kappa(L_1, L_2)$. The time complexity analysis of Test 2 and Test 3 can be found in Section 4.5.

$$L_1: \quad q_{01} \xrightarrow{u} p_1 \xrightarrow{v} p_1 \xrightarrow{w\alpha\beta\bar{\alpha}} f_1 \xRightarrow{\overline{w}} q_1 \xRightarrow{\overline{v}} q_1 \xRightarrow{\overline{u}} q_1'$$
$$\overline{L_2}: \quad q_{02} \xrightarrow{u} p_2 \xrightarrow{v} p_2 \xrightarrow{w\alpha\overline{\beta}\bar{\alpha}} f_2 \xRightarrow{\overline{w}} q_2 \xRightarrow{\overline{v}} q_2 \xRightarrow{\overline{u}} q_2'$$

Fig. 7. Runs through $\mathcal{A}_1$ and $\mathcal{A}_2$ for the word π.

Lemma 4.17. *Suppose we passed all of Tests 1, 2 and 3 without obtaining the result "$\mathcal{H}_\kappa(L_1, L_2)$ is not regular". Then the hairpin completion $\mathcal{H}_\kappa(L_1, L_2)$ is regular.*

Proof. Let $\pi \in \mathcal{H}_\kappa(L_1, L_2)$. Write $\pi = \gamma\alpha\beta\bar{\alpha}\bar{\gamma}$ such that $\gamma\alpha$ is the minimal gamma-alpha-prefix of π and $|\alpha| = \kappa$. Therefore, either $\gamma\alpha\beta\bar{\alpha} \in L_1$ or $\alpha\beta\bar{\alpha}\bar{\gamma} \in L_2$; we assume $\gamma\alpha\beta\bar{\alpha} \in L_1$, by symmetry. In addition, we may assume that $|\gamma| > n^4$ (cf. Proposition 4.9 and Test 0). We can factorize $\gamma = uvw$ with $|uv| \leqslant n^4$ and $|v| \geqslant 1$ such that there are runs as in Fig. 7 where $f_1 \in \mathcal{F}_1$.

We infer from Test 1 that $w\alpha$ is a prefix of some word in v^+. Hence, we can write $w\alpha\beta = v^i xyz$ with $i \geqslant 0$ such that $v^i xy$ is the maximal common prefix of $w\alpha\beta$ and some word in v^+, $w\alpha \in v^*x$ with $\kappa \leqslant |x| < |v| + \kappa$, and $|y| < |v|$.

We see that for some $k_\pi \geqslant \ell_\pi \geqslant 0$ we can write

$$\pi = uv^{k_\pi} xyz\bar{x}\bar{v}^{\ell_\pi}\bar{u}.$$

Moreover, $uv^k xyz\bar{x}\bar{v}^\ell\bar{u} \in \mathcal{H}_\kappa(L_1, L_2)$ for all $k \geqslant \ell \geqslant 0$. There are only finitely many choices for u, v, x, y (due to the length bounds) and for each of them there is a regular set R_z associated to the finite collection of bridges such that

$$\pi \in \left\{uv^k xyR_z\bar{x}\bar{v}^\ell\bar{u} \;\middle|\; k \geqslant \ell \geqslant 0\right\} \subseteq \mathcal{H}_\kappa(L_1, L_2).$$

More precisely, we can choose $R_z = \{1\}$ for $z = 1$ and otherwise we can choose

$$R_z \in \left\{B(c_1, c_2, d_1, d_2) \cap a\Sigma^* \;\middle|\; (c_1, c_2, d_1, d_2) \text{ is a bridge and } a \in \Sigma\right\}.$$

Note that the sets $\{uv^k xyR_z\bar{x}\bar{v}^\ell\bar{u} \mid k \geqslant \ell \geqslant 0\}$ are not regular in general. However, if we bound ℓ by n, then the finite union

$$\bigcup_{0 \leqslant \ell \leqslant n} \left\{uv^k xyR_z\bar{x}\bar{v}^\ell\bar{u} \;\middle|\; k \geqslant \ell\right\}$$

is regular. Thus, we may assume that $\ell > n$. Let $e_2 = p_2 \cdot x\bar{z}\bar{y}\bar{x}$. We have $e_2 \cdot \bar{v}^n = q_2$ and if we see a final state during the computation of $e_2 \cdot \bar{v}^n$, then for all $\ell > k \geqslant n$ and $z \in R_z$ we see that $uv^k xyz\bar{x}\bar{v}^\ell\bar{u} \in \mathcal{H}_\kappa(L_1, L_2)$, due to a suffix in L_2 and

$$uv^n v^+ xyR_z\bar{x}\bar{v}^+\bar{v}^n\bar{u} \subseteq \mathcal{H}_\kappa(L_1, L_2).$$

Otherwise, Test 2 or Test 3 tells us that for all $z \in R_z$ the word $xyz\bar{x}\bar{v}$ has a factorization $\mu\delta\nu\bar{\delta}\bar{\mu}$ such that $|\delta| = \kappa$ and $p_2 \cdot \mu\delta\bar{\nu}\bar{\delta} \in \mathcal{F}_2$. The paths $q_{02} \cdot u = p_2$ and $p_2 \cdot v = p_2$ yield $\delta\nu\bar{\delta}\bar{\mu}\bar{v}^*\bar{u} \subseteq L_2$ and, again,

$$uv^n v^+ xyR_z\bar{x}\bar{v}^+\bar{v}^n\bar{u} \subseteq \mathcal{H}_\kappa(L_1, L_2).$$

Hence, the hairpin completion $\mathcal{H}_\kappa(L_1, L_2)$ is a finite union of regular languages and, therefore, regular itself. $\square$

4.5. Time complexities of Test 2 and Test 3

In this section we provide the final step of the proof of Theorem 4.1. We show that Test 2 can be performed in time $\mathcal{O}(N^2)$ and that Test 3 can be performed in time $\mathcal{O}(n_{12}n_1^2n_2^2n)$. Thus, in case $L_1 = \overline{L_2}$ both tests run in $\mathcal{O}(n^6)$, and in general Test 2 runs in $\mathcal{O}(n^8)$ and Test 3 runs in $\mathcal{O}(n^7)$.

Test 2. Decide the existence of words $x, y \in \Sigma^*$ and states $(d_1, d_2) \in \mathcal{Q}_1 \times \mathcal{Q}_2$ satisfying conditions 1 to 5 of Lemma 4.15 with $z = 1$, but where for all factorizations $xy\bar{x}\bar{v} = \mu\delta\beta\bar{\delta}\bar{\mu}$ with $|\delta| = \kappa$ we have $p_2 \cdot \mu\delta\overline{\beta}\bar{\delta} \notin \mathcal{F}_2$. If we find such a situation, then stop with the output that $\mathcal{H}_\kappa(L_1, L_2)$ is not regular.

Lemma 4.18. *Test 2 can be performed in time $\mathcal{O}(N^2)$.*

Proof. For a strongly connected component $s \in S$ with $A_s = ((p_1, p_2), q_1, q_2)$ and $v_s = v$, we have to compute all words x and y such that there are runs

$$p_1 \xrightarrow{xy} d_1 \xrightarrow{\bar{x}\bar{v}^{n_1}} q_1, \qquad p_2 \xrightarrow{x} d_2 \xrightarrow{\bar{y}\bar{x}\bar{v}^{n_2}} q_2$$

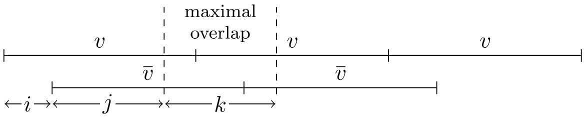

Fig. 8. Matching positions of v_i^2 with $\bar{v}^2$.

and the conditions 1 to 5 are satisfied. In addition, we demand that during the computation of $d_2 \cdot \bar{y}\bar{x}\bar{v}^{n_2}$ we do not meet any final state in $\mathcal{F}_2$ after more than $\kappa - 1$ steps. (In case such a final state exists, either condition 5 is breached or a factorization $xy\bar{x}\bar{v} = \mu\delta\beta\bar{\delta}\bar{\mu}$ with $|\delta| = \kappa$ and $p_2 \cdot \mu\delta\overline{\beta}\overline{\delta} \in \mathcal{F}_2$ exists.) By backwards searches in $\mathcal{A}_1$ and $\mathcal{A}_2$ starting at states q_1 and q_2, respectively, and searching for paths labeled by suffixes of $\bar{v}^+$, we compute all pairs (x, xy) satisfying these conditions in time $\mathcal{O}(N \cdot N_s)$.

At this stage we also compute the position $\ell(x, xy)$ of the last final state during the run $p_2 \cdot vx\bar{y}\bar{x}$ and we let $\ell(x, xy) = 0$ if no such state exists. Note that $0 \leqslant \ell(x, xy) < N_s + |x| + \kappa$. If a factorization $xy\bar{x}\bar{v} = \mu\delta\beta\bar{\delta}\bar{\mu}$ with $|\delta| = \kappa$ and $p_2 \cdot \mu\delta\overline{\beta}\overline{\delta} \in \mathcal{F}_2$ exists, then $|xy\bar{x}\bar{v}| - \ell(x, xy)$ gives us a lower bound for the length of μ.

Let $m(x, xy)$ be the length of the longest μ such that a factorization $xy\bar{x}\bar{v} = \mu\delta\beta\bar{\delta}\bar{\mu}$ with $|\delta| = \kappa$ exists (without the condition $p_2 \cdot \mu\delta\overline{\beta}\overline{\delta} \in \mathcal{F}_2$).

There is a factorization $xy\bar{x}\bar{v} = \mu\delta\beta\bar{\delta}\bar{\mu}$ with $|\delta| = \kappa$ and $p_2 \cdot \mu\delta\overline{\beta}\overline{\delta} \in \mathcal{F}_2$ if and only if $m(x, xy) \geqslant |xy\bar{x}\bar{v}| - \ell(x, xy)$ and $\ell(x, xy) - \kappa \geqslant |xy\bar{x}\bar{v}|/2$.

We need to compute the values $m(x, xy)$ efficiently, which turns out to be a little tricky. For $0 \leqslant i < N_s$ we let $v_i = v[i+1, N_s]v[1, i]$ be the conjugate of v starting at the $(i+1)$-st letter. We wish to match positions in v_i^2 with positions in $\bar{v}^2$. For each $0 \leqslant j < N_s$ we store the maximal $k \leqslant N_s$ such that $v_i^2[j, j+k] = \bar{v}^2[j, j+k]$ in a table entry $M(i, j)$, see Fig. 8. For each i one run (from right to left) over the words v_i^2 and $\bar{v}^2$ is enough. It takes $\mathcal{O}(N_s^2)$ time to build the table M. Now, if we know the length m' of the longest common prefix of $v_{|xy|}$ and $\bar{x}\bar{v}$, then $m(x, xy) = |xy| + m' - \kappa$ (yet at most $|xy\bar{x}\bar{v}|/2 - \kappa$). The length of m' is stored in $M(|xy\bar{x}| \bmod N_s, (-|\bar{x}|) \bmod N_s)$, hence we have access to $m(x, xy)$ in constant time.

Summing up, Test 2 can be performed in $\mathcal{O}(\sum_{s \in S} N \cdot N_s) \subseteq \mathcal{O}(N^2)$. $\square$

Test 3. Decide the existence of words $x, y, z \in \Sigma^*$ with $z \neq 1$ and states $(d_1, d_2) \in \mathcal{Q}_1 \times \mathcal{Q}_2$ satisfying conditions 1 to 5 of Lemma 4.15, but where for all factorizations $xyz\bar{x}\bar{v} = \mu\delta\beta\bar{\delta}\bar{\mu}$ with $|\delta| = \kappa$ we have $p_2 \cdot \mu\delta\overline{\beta}\overline{\delta} \notin \mathcal{F}_2$. If we find such a situation, then stop with the output that $\mathcal{H}_\kappa(L_1, L_2)$ is not regular.

Lemma 4.19. *Test 3 can be performed in time $\mathcal{O}(n_{12}n_1^2n_2^2n)$.*

Proof. For $s \in S$ with $A_s = ((p_1, p_2), q_1, q_2)$ and $v_s = v$, we create two tables T_1 and T_2. Table T_1 holds all pairs $(c_2, d_1) \in \mathcal{Q}_2 \times \mathcal{Q}_1$ such that a word x exists with

1. $\kappa \leqslant |x| < |v| + \kappa$ and x is a prefix of a word in v^+,
2. $p_2 \cdot x = c_2$,
3. $d_1 \cdot \bar{x}\bar{v}^{n_1} = q_1$, and during the computation of $d_1 \cdot \bar{x}\bar{v}^{n_1}$ we see a final state after exactly κ steps and then never again.

We call x a witness for $(c_2, d_1) \in T_1$. Table T_2 holds all triples $(c_1, d_2, a) \in \mathcal{Q}_1 \times \mathcal{Q}_2 \times \Sigma$ such that a proper prefix $y' < v$ exists with

1. $y'a$ is no prefix of v,
2. $p_1 \cdot y' = c_1$,
3. $d_2 \cdot \overline{y'}\bar{v}^{n_2} = q_2$, and during the computation of $d_2 \cdot \overline{y'}\bar{v}^{n_2}$ we do not see a final state after κ or more steps.

We call y' a witness for $(c_1, d_2, a) \in T_2$. By backwards computing in the second component, tables T_1 and T_2 can be created in $\mathcal{O}(N_s n_1)$ and $\mathcal{O}(N_s n_2)$, respectively.

We claim Test 3 to yield that $\mathcal{H}_\kappa(L_1, L_2)$ is not regular if and only if there exists a pair $(c_2, d_1) \in T_1$ and a triple $(c_1, d_2, a) \in T_2$ such that (c_1, c_2, d_1, d_2) is an a-bridge. Recall that the list of a-bridges is precomputed.

First, assume $(c_2, d_1) \in T_1$, $(c_1, d_2, a) \in T_2$, and (c_1, c_2, d_1, d_2) is indeed an a-bridge. Let x and y' be the witnesses for $(c_2, d_1) \in T_1$ and $(c_1, d_2, a) \in T_2$, respectively. Choose $z \in B(c_1, c_2, d_1, d_2) \cap a\Sigma^*$ and y such that xy is a prefix of some word in v^+, $|xy| \equiv |y'| \pmod{|v|}$, and $|y| < |v|$. Verify that x, y, z and (d_1, d_2) satisfy the conditions 1 to 5 of Test 3. However, for any factorization $xyz\bar{x}\bar{v} = \mu\delta\beta\bar{\delta}\bar{\mu}$ with $|\delta| = \kappa$, the word $\mu\delta$ has to be a prefix of xy, since xya is no prefix of vx. During the computation of $d_2 \cdot \overline{y'}\bar{v}^{n_2}$ we did not see a final state after more than $\kappa - 1$ steps. The same holds for the computation of $d_2 \cdot \bar{y}\bar{x}\bar{v}^{n_2}$ and, therefore, we have $p_2 \cdot \mu\delta\overline{\beta}\overline{\delta} \notin \mathcal{F}_2$.

Now assume that $x, y, z \in \Sigma^*$, $z \neq 1$, and $(d_1, d_2) \in \mathcal{Q}_1 \times \mathcal{Q}_2$ exist, which satisfy the conditions 1 to 5 of Test 3 but where for all factorizations $xyz\bar{x}\bar{v} = \mu\delta\beta\bar{\delta}\bar{\mu}$ with $|\delta| = \kappa$ we have $p_2 \cdot \mu\delta\overline{\beta}\overline{\delta} \notin \mathcal{F}_2$. Choose $y' < v$ such that $|xy| \equiv |y'| \pmod{|v|}$.

Let $c_2 = p_2 \cdot x$, $c_1 = p_1 \cdot y'$ and $a \in \Sigma$ be the first letter of z. Obviously, (c_1, c_2, d_1, d_2) is an a-bridge and x is a witness for $(c_2, d_1) \in T_1$. If we saw a final state after more than $\kappa - 1$ steps during the computation of $d_2 \cdot \overline{y'}\,\overline{v}^{n_2}$, then a factorization $xyz\bar{x}\bar{v} = \mu\delta\beta\bar{\delta}\overline{\mu}$ where $|\delta| = \kappa$ and $p_2 \cdot \mu\delta\overline{\beta}\bar{\delta} \in \mathcal{F}_2$ would exist. Thus, y' is a witness for $(c_1, d_2, a) \in T_2$.

Since the table of a-bridges is precomputed (see Lemma 4.6), this test can be performed in time $\mathcal{O}(|T_1| \cdot |T_2|)$. The set of all first components of T_1 (respectively T_2) is bounded by both, the size N_s and n_2 (respectively n_1). Therefore, we have $|T_1| \in \mathcal{O}(n_1 \cdot \min(N_s, n_2))$ and $|T_2| \in \mathcal{O}(n_2 \cdot \min(N_s, n_1))$. By symmetry, assume $n_2 \leqslant n_1$.

Test 3 can be performed in time

$$\mathcal{O}\Bigl(\sum_{s\in S}\bigl(N_s n_1 + N_s n_2 + n_1 n_2 \cdot \min(N_s, n_1) \cdot \min(N_s, n_2)\bigr)\Bigr)$$
$$\subseteq \mathcal{O}\Bigl(n_{12} n_1^2 n_2 + n_{12} n_1 n_2^2 + \sum_{s\in S, N_s \geqslant n_2} n_1^2 n_2^2 + \sum_{s\in S, N_s < n_2} N_s^2 n_1 n_2\Bigr).$$

Recall that $n_1 \leqslant n \leqslant n_{12} \leqslant n_1 n_2 \leqslant n^2$ and $\sum_{s\in S} N_s \leqslant N = n_{12} n_1 n_2$. Since there are at most $n_{12} n_1$ strongly connected components with a size of n_2 or more states,

$$\sum_{s\in S, N_s \geqslant n_2} n_1^2 n_2^2 \leqslant n_{12} n_1^3 n_2^2.$$

For the last term we can use the approximation

$$\sum_{s\in S, N_s < n_2} N_s^2 n_1 n_2 \leqslant \sum_{s\in S, N_s < n_2} N_s n_1 n_2^2 \leqslant n_{12} n_1^2 n_2^3.$$

We conclude, Test 3 can be performed in time $\mathcal{O}(n_{12} n_1^2 n_2^2 n)$. □

5. Growth and topological entropy

In this section we compare the growth function of $\mathcal{H}_\kappa(L_1, L_2)$ with the growth functions of L_1 and L_2. For this it is enough to consider the growth functions of L_1 and L_2 restricted to those words which can be used for a hairpin completion. Let $L_1' = L_1 \cap \bigcup_{\alpha\in\Sigma^\kappa} \Sigma^*\alpha\Sigma^*\overline{\alpha}$ and $L_2' = L_2 \cap \bigcup_{\alpha\in\Sigma^\kappa} \alpha\Sigma^*\overline{\alpha}\Sigma^*$; clearly, $\mathcal{H}_\kappa(L_1', L_2') = \mathcal{H}_\kappa(L_1, L_2)$. Thus, the growth function of $\mathcal{H}_\kappa(L_1, L_2)$ should be compared with the growth functions of L_1' and L_2' rather than with those of L_1 and L_2. The languages L_1' and L_2' are still regular and we can compute their growth functions as part of a preprocessing. In order to simplify notation, we assume from the very beginning that L_1 and L_2 contain only words that can form hairpins. Thus, we assume $L_1' = L_1$ and $L_2' = L_2$.

The growth indicator λ_L of a language L has been defined in Section 2.3. Recall that the growth of $|L \cap \Sigma^m|$ behaves essentially as λ_L^m.

Theorem 5.1. *Let $\lambda = \max\{\lambda_{L_1}, \lambda_{L_2}\}$ be the maximum growth indicator of L_1 and L_2, and let η be the growth indicator of $\mathcal{H}_\kappa(L_1, L_2)$. Then we have the following assertions.*

1. *The values λ and η satisfy*:

$$\sqrt{\lambda} \leqslant \eta \leqslant \lambda.$$

   In particular, the growth of $\mathcal{H}_\kappa(L_1, L_2)$ is exponential (*respectively polynomial, respectively finite*) *if and only if the growth of $L_1 \cup L_2$ is exponential* (*respectively polynomial, respectively finite*)*. In other terms this means that the topological entropy of $\mathcal{H}_\kappa(L_1, L_2)$ is positive* (*respectively zero, respectively equal to $-\infty$*) *if and only if the topological entropy of $L_1 \cup L_2$ is positive* (*respectively zero, respectively equal to $-\infty$*)*.*
2. *If $\mathcal{H}_\kappa(L_1, L_2)$ is regular, then we have $\eta = \lambda$. Thus, the growth indicators* (*respectively topological entropies*) *of $\mathcal{H}_\kappa(L_1, L_2)$ and of $L_1 \cup L_2$ are equal.*

The theorem will follow by Lemma 5.4 and Lemma 5.5 in Section 5.1 which compare the growth indicators λ and η with the growth indicators of the languages B_μ and R_μ for $\mu \in M$. In preparation for the proof we recall some well-known on growth indicators of (regular) languages.

Consider two languages K_1 and K_2. The growth indicator of their union is $\lambda_{K_1\cup K_2} = \max\{\lambda_{K_1}, \lambda_{K_2}\}$. Furthermore, if $K_1 \neq \emptyset \neq K_2$ the growth indicator of their concatenation is $\lambda_{K_1 K_2} = \max\{\lambda_{K_1}, \lambda_{K_2}\}$, too.

Now, let K be a regular language. The *prefix closure* of K is defined as

$$\mathrm{Pref}(K) = \bigl\{u \in \Sigma^* \bigm| \exists v \in \Sigma^*\colon uv \in K\bigr\}.$$

The next lemma is folklore. It holds more generally for *factorial closures* of regular languages. Instead of working with growths we might also use the notion of topological entropy as done e.g., in [31]. We content ourselves with the lemma for prefix closures. Actually, the general case is an immediate consequence by applying first a prefix closure and then a suffix closure to the prefix closed language. For sake of completeness, we give a short proof.

Lemma 5.2. *Let K be a regular language, then we have $\lambda_K = \lambda_{\mathrm{Pref}(K)}$.*

Proof. The inequality $\lambda_K \leqslant \lambda_{\mathrm{Pref}(K)}$ is obvious because $K \subseteq \mathrm{Pref}(K)$. Conversely, let k be the number of states of some DFA accepting the language K. For a word $u \in \mathrm{Pref}(K) \cap \Sigma^m$, there is some word v such that $uv \in K$ and, moreover, we may assume $|v| \leqslant k$. For $m \in \mathbb{N}$ let $h_m : \mathrm{Pref}(K) \cap \Sigma^m \to \Sigma^*$ be a mapping such that both, $h_m(u) = uv \in K$ and $|v| \leqslant k$. Note that h_m is injective. Thus, for all $m \in \mathbb{N}$ we obtain

$$\left|\mathrm{Pref}(K) \cap \Sigma^m\right| \leqslant \sum_{i=m}^{m+k} \left|K \cap \Sigma^i\right|.$$

For all $\nu > \lambda_K$ there exists some $c > 0$ such that $|K \cap \Sigma^i| \leqslant c\nu^i$ for all $i \in \mathbb{N}$. Therefore,

$$\left|\mathrm{Pref}(K) \cap \Sigma^m\right| \leqslant \sum_{i=m}^{m+k} c\nu^i \leqslant c(k+1)\nu^{k+m}.$$

We conclude $\lambda_{\mathrm{Pref}(K)} \leqslant \nu$ and hence $\lambda_K = \lambda_{\mathrm{Pref}(K)}$. $\square$

Remark 5.3. The conclusion $\lambda_K = \lambda_{\mathrm{Pref}(K)}$ in Lemma 5.2 does not hold for unambiguous linear languages K, in general. For example, let K be the language of *pseudo-palindromes* $K = \{w \in \Sigma^* \mid w = \overline{w}\}$. Then, we have $\lambda_K = \sqrt{|\Sigma|}$ and $\lambda_{\mathrm{Pref}(K)} = |\Sigma|$.

Actually, there are uncountably many languages with $\lambda_K = 1$, but $\mathrm{Pref}(K) = \Sigma^*$ and hence $\lambda_{\mathrm{Pref}(K)} = |\Sigma|$. Define K by choosing for every length exactly one random word; then $|K \cap \Sigma^m| = 1$ for all m, but $\mathrm{Pref}(K) = \Sigma^*$ with probability 1.

5.1. Proof of Theorem 5.1

Recall from Lemma 4.4 that the hairpin completion is the disjoint union

$$\mathcal{H}_\kappa(L_1, L_2) = \bigcup_{\mu \in M} B_\mu^{R_\mu}.$$

We let σ_μ and ρ_μ be the growth indicators of B_μ and R_μ, respectively. By $\sigma = \max\{\sigma_\mu \mid \mu \in M\}$ and $\rho = \max\{\rho_\mu \mid \mu \in M\}$ we denote the maximum growth indicators of all B_μ and all R_μ, respectively. The next lemma compares the growth indicator λ with the growth indicators σ and ρ.

Lemma 5.4. *We have $\lambda = \max\{\sigma, \rho\}$.*

Proof. We start by proving $\lambda \geqslant \max\{\sigma, \rho\}$. Let $\mu \in M$ be fixed. For $\gamma\alpha \in R_\mu$ with $|\alpha| = \kappa$ and $\beta \in B_\mu$, we have $\gamma\alpha\beta\overline{\alpha} \in L_1$ or $\alpha\beta\overline{\alpha}\overline{\gamma} \in L_2$. Thus, we may define a mapping $h : (R_\mu \times B_\mu) \to L_1 \cup L_2$ such that

$$h(\gamma\alpha, \beta) = \begin{cases} \gamma\alpha\beta\overline{\alpha} & \text{if } \gamma\alpha\beta\overline{\alpha} \in L_1, \\ \alpha\beta\overline{\alpha}\overline{\gamma} & \text{otherwise}. \end{cases}$$

Obviously, $|\gamma\alpha| + |\beta| = |h(\gamma\alpha, \beta)| - \kappa$. Since a word $w \in L_1 \cup L_2$ of length m can form less than $2m$ hairpin completions, the cardinality of the inverse image h of w is $|h^{-1}(w)| < 2m$. Using the mapping h, we can compare the growth $r_m = |R_\mu B_\mu \cap \Sigma^m|$ with the growth $\ell_m = |(L_1 \cup L_2) \cap \Sigma^m|$; that is $r_m \leqslant 2(m+\kappa) \cdot \ell_{m+\kappa}$ for $m \in \mathbb{N}$.

For $\nu > \lambda = \lambda_{L_1 \cup L_2}$ we choose ν' from the open interval (λ, ν). There exists $c' > 0$ such that $r_m \leqslant 2(m+\kappa)c'\nu'^\kappa \nu'^m$ for all $m \in \mathbb{N}$ and, as the function ν^m is growing faster than ν'^m, there is some $c > 0$ such that $r_m \leqslant c\nu^m$ for all $m \in \mathbb{N}$. Therefore, $\max\{\sigma_\mu, \rho_\mu\} \leqslant \nu$ for all $\nu > \lambda$, whence $\max\{\sigma_\mu, \rho_\mu\} \leqslant \lambda$. As this inequality holds for all $\mu \in M$, we deduce $\lambda \geqslant \max\{\sigma, \rho\}$.

Conversely, we will prove that L_1 is included in a language K whose growth indicator is $\max\{\sigma, \rho\}$. As there is a symmetric language that includes L_2, this yields $\lambda \leqslant \max\{\sigma, \rho\}$. Let $B = \bigcup_{\mu \in M} B_\mu$, let $R = \bigcup_{\mu \in M} R_\mu$, and let K be the prefix closure $K = \mathrm{Pref}(RB\Sigma^\kappa)$. As the growth indicator of $RB\Sigma^\kappa$ is $\lambda_{RB\Sigma^\kappa} = \max\{\sigma, \rho\}$ and by Lemma 5.2, we conclude $\lambda_K = \max\{\sigma, \rho\}$.

Now, consider $w \in L_1$. By assumption, w can form a hairpin on its right side. We let $\pi \in \mathcal{H}_\kappa(\{w\}, \emptyset)$ be a hairpin completion of w. Let $\gamma\alpha$ be the minimal gamma-alpha-prefix of π with $|\alpha| = \kappa$ and β such that $\pi = \gamma\alpha\beta\overline{\alpha}\overline{\gamma}$. The word w has to be a prefix of $\gamma\alpha\beta\overline{\alpha} \in RB\Sigma^\kappa$, by the minimality of $|\gamma|$. Thus, $L_1 \subseteq K$ as desired. $\square$

Next, let us compare the growth indicator η with the growth indicators σ and ρ.

Fig. 9. Growth indicators λ and η in dependency of σ and ρ.

Lemma 5.5. *We have* $\eta = \max\{\sigma, \sqrt{\rho}\}$.

Proof. Let τ_μ be the growth indicator of $B_\mu^{R_\mu}$ for $\mu \in M$. Since $\mathcal{H}_\kappa(L_1, L_2) = \bigcup_{\mu \in M} B_\mu^{R_\mu}$, we see that $\eta = \max\{\tau_\mu \mid \mu \in M\}$. Thus, in order to prove the claim, it suffices to show that $\tau_\mu = \max\{\sigma_\mu, \sqrt{\rho_\mu}\}$ for $\mu \in M$. Let $\mu \in M$ be fixed from here on and recall that B_μ and R_μ are non-empty. We let

$$g_{B_\mu}(z) = \sum_{m \geqslant 0} b_m z^m \quad \text{with } b_m = |B_\mu \cap \Sigma^m|,$$

$$g_{R_\mu}(z) = \sum_{m \geqslant 0} r_m z^m \quad \text{with } r_m = |R_\mu \cap \Sigma^m|.$$

It will be convenient to let $r_{i+1/2} = 0$ for $i \in \mathbb{N}$.

First, let us prove $\tau_\mu \geqslant \sigma_\mu$. Let $v \in R_\mu$ and consider $K = vB_\mu\bar{v}$. Obviously, $K \subseteq B_\mu^{R_\mu}$ and hence $\tau_\mu \geqslant \lambda_K = \sigma_\mu$.

Next, we prove $\tau_\mu \geqslant \sqrt{\rho_\mu}$. Let $K = \{\beta\}^{R_\mu} \subseteq B_\mu^{R_\mu}$ for some $\beta \in B_\mu$. The generating function of K is given as $g_K(z) = \sum_{m \geqslant 0} r_{(m-|\beta|)/2} z^m$. For all $\nu > \lambda_K$ there exists $c > 0$ such that

$$\forall m \in \mathbb{N}\colon r_{(m-|\beta|)/2} \leqslant c\nu^m \quad \iff \quad \forall m \in \mathbb{N}\colon r_m \leqslant c\nu^{|\beta|}(\nu^2)^m$$

and, therefore, $\nu^2 \geqslant \rho_\mu$. We conclude $\tau_\mu \geqslant \lambda_K \geqslant \sqrt{\rho_\mu}$.

Finally, we need to prove that $\tau_\mu \leqslant \max\{\sigma_\mu, \sqrt{\rho_\mu}\}$. As $B_\mu^{R_\mu}$ is unambiguous and by Lemma 4.4,

$$g_{B_\mu^{R_\mu}}(z) = \sum_{m \geqslant 0} d_m z^m \quad \text{with } d_m = \sum_{k+\ell=m} b_k r_{\ell/2}.$$

For $\nu > \max\{\sigma_\mu, \sqrt{\rho_\mu}\}$ we choose ν' from the open interval $(\max\{\sigma_\mu, \sqrt{\rho_\mu}\}, \nu)$. By that choice, ν^m grows faster than ν'^m and there is $c' > 0$ such that for all $m \in \mathbb{N}$ and $k + \ell = m$, we have $b_k r_{\ell/2} \leqslant c'\nu'^m$. Thus, there is $c > 0$ such that for all $m \in \mathbb{N}$, the inequality $d_m \leqslant mc'\nu'^m \leqslant c\nu^m$ holds. This deduces the last step in the proof, $\tau_\mu \leqslant \max\{\sigma_\mu, \sqrt{\rho_\mu}\}$. $\square$

Lemma 5.4 and Lemma 5.5 yield a development of the growth indicators λ and η as shown in Fig. 9. The growth indicator η is at least $\sqrt{\lambda}$ and at most λ; therefore, we deduce the first statement of Theorem 5.1. The second statement of Theorem 5.1 claims that if the hairpin completion is regular, then $\lambda = \eta$. We infer from Lemma 4.12 that if $\mathcal{H}_\kappa(L_1, L_2)$ is regular, then the growth of all R_μ is at most polynomial, more precisely, it is linear or finite, i.e., $\rho = 1$ or $\rho = 0$. We conclude $\lambda = \max\{\sigma, \rho\} = \max\{\sigma, \sqrt{\rho}\} = \eta$.

6. Conclusion and open problems

We proved that the regularity problem for hairpin completions of regular languages is decidable in time $\mathcal{O}(n^8)$ (respectively $\mathcal{O}(n^6)$ in case when $L_1 = \overline{L_2}$). The question is whether we could have a lower degree polynomial for the time bound. The first step of the algorithm is the construction of an automaton $\mathcal{A}$, which has size $\mathcal{O}(n^4)$ (respectively $\mathcal{O}(n^3)$). Thus, when speaking of time complexity with respect to the size of $\mathcal{A}$, the algorithm uses quadratic time, only. Furthermore, it seems that all pairs of states of $\mathcal{A}$ must be considered in the decision procedure, hence the time bound might be optimal for our approach. Further improvement of the time complexity would probably call for a completely different approach. For the one-sided hairpin completion of a regular language, we provide a faster algorithm which runs in quadratic time. Assuming that the one-sided hairpin completion is the closest to the biochemical setting, our quadratic time bound is quite reasonable. The regularity problem can be considered as easy for the one-sided hairpin completion.

The polynomial time bounds are due to the fact that we use DFAs for the specification of L_1 and $\overline{L_2}$. One might argue that a natural specification of a regular language uses an NFA, and we are hiding a first exponential blow-up. Actually, it is open what happens if L_1 and L_2 are given by NFAs. The problem becomes much more difficult and asks for further investigation. We suspect that deciding regularity of $\mathcal{H}_\kappa(L_1, L_2)$ might become PSPACE-complete.

Under the natural assumption that every word from the underlying languages can form a hairpin we have shown in Section 2.3 that the hairpin completion has an exponential growth if and only if one of the underlying languages has an exponential growth. On the other hand, the structure of regular languages with polynomial growths is well understood by [29]. It might be worth to investigate the structure of hairpin completions for regular languages with polynomial growths.

The situation for iterated hairpin completions is very interesting, but far from understood. The iterated hairpin completion has been defined as $\mathcal{H}_\kappa^*(L) = \bigcup_{i \geqslant 0} \mathcal{H}_\kappa^i(L)$ where $\mathcal{H}_\kappa(L) = \mathcal{H}_\kappa(L, L)$. The iterated hairpin completion of a singleton language is not context-free, in general, but still in NL, hence context-sensitive, see [13]. Whether or not regularity of the iterated hairpin completion of a singleton or, more generally, a regular language is decidable, remains a challenging open problem. A partial result for so-called *non-crossing words* has been shown in [12]. We also suggest to investigate $\overline{\mathcal{H}}_\kappa^*(L)$ with $\overline{\mathcal{H}}_\kappa(L) = \mathcal{H}_\kappa(L, \bar{L})$. Actually, this operation is closer to the spirit of the paper and seems to be closer to DNA-computing.

Another interesting problem concerns the hairpin lengthening of regular languages, which is a similar operation as the hairpin completion. We call $\gamma_1\alpha\beta\overline{\alpha}\overline{\gamma_2}$ a (right) hairpin lengthening of $\gamma_1\alpha\beta\overline{\alpha}$ if γ_2 is a suffix of γ_1 and we call it a (left) hairpin lengthening of $\alpha\beta\overline{\alpha}\overline{\gamma_2}$ if $\overline{\gamma_1}$ is a prefix of $\overline{\gamma_2}$. The hairpin lengthening $\mathcal{HL}_\kappa(L_1, L_2)$ of languages L_1 and L_2 is introduced analogously to the hairpin completion. It is known that the hairpin lengthening $\mathcal{HL}_\kappa(L_1, L_2)$ of two regular languages is linear context-free. However, $\mathcal{HL}_\kappa(L_1, L_2)$ is not unambiguous, in general, see [32]. This might indicate that deciding regularity of the hairpin lengthening $\mathcal{HL}_\kappa(L_1, L_2)$ is more difficult than for the hairpin completion. It is open whether the regularity problem for hairpin lengthening is decidable. The iterated hairpin lengthening of a regular language remains regular, see [33,34] in contrast to the hairpin completion.

Acknowledgments

The authors thank the anonymous referees for many valuable comments and suggestions which improved the presentation of the results. The first two authors also thank James Currie and Jeffrey Shallit for the invitation to the BIRS workshop *Outstanding Challenges in Combinatorics on Words* which made it possible to complete the revision at The Banff Centre in February 2012.

References

[1] J.G. Williams, A.R. Kubelik, K.J. Livak, J.A. Rafalski, S.V. Tingey, DNA polymorphisms amplified by arbitrary primers are useful as genetic markers, Nucleic Acids Research 18 (1990) 6531–6535.
[2] R. Deaton, R. Murphy, M. Garzon, D. Franceschetti, S. Stevens, Good encodings for DNA-based solutions to combinatorial problems, Proc. of DNA-based computers DIMACS Series 44 (1998) 247–258.
[3] M. Garzon, R. Deaton, P. Neathery, R. Murphy, D. Franceschetti, E. Stevens, On the encoding problem for DNA computing, in: The Third DIMACS Workshop on DNA-Based Computing, 1997, pp. 230–237.
[4] M. Garzon, R. Deaton, L.F. Nino, E. Stevens, M. Wittner, Encoding genomes for DNA computing, in: J.R. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D.B. Fogel, M.H. Garzon, D.E. Goldberg, H. Iba, R. Riolo (Eds.), Genetic Programming 1998: Proceedings of the Third Annual Conference, Morgan Kaufmann, University of Wisconsin, Madison, Wisconsin, USA, 1998, pp. 684–690.
[5] L. Kari, S. Konstantinidis, E. Losseva, P. Sosík, G. Thierrin, Hairpin structures in DNA words, in: A. Carbone, N.A. Pierce (Eds.), DNA, in: Lecture Notes in Computer Science, vol. 3892, Springer, 2005, pp. 158–170.
[6] L. Kari, K. Mahalingam, G. Thierrin, The syntactic monoid of hairpin-free languages, Acta Inf. 44 (2007) 153–166.
[7] M. Hagiya, M. Arita, D. Kiga, K. Sakamoto, S. Yokoyama, Towards parallel evaluation and learning of boolean μ-formulas with molecules, in: Second Annual Genetic Programming Conf., pp. 105–114.
[8] E. Winfree, Whiplash PCR for O(1) computing, University of Pennsylvania, pp. 175–188.
[9] K. Sakamoto, D. Kiga, K. Komiya, H. Gouzu, S. Yokoyama, S. Ikeda, H. Sugiyama, M. Hagiya, State transitions by molecules, Biosystems 52 (1999) 81–91.
[10] D. Cheptea, C. Martín-Vide, V. Mitrana, A new operation on words suggested by DNA biochemistry: Hairpin completion, Transgressive Computing (2006) 216–228.
[11] M. Ito, P. Leupold, F. Manea, V. Mitrana, Bounded hairpin completion, Inf. Comput. 209 (2011) 471–485.
[12] L. Kari, S. Kopecki, S. Seki, Iterated hairpin completions of non-crossing words, in: M. Bieliková, G. Friedrich, G. Gottlob, S. Katzenbeisser, G. Turán (Eds.), SOFSEM, in: Lecture Notes in Computer Science, vol. 7147, Springer, 2012, pp. 337–348.
[13] S. Kopecki, On iterated hairpin completion, Theoretical Computer Science 412 (2011) 3629–3638.
[14] F. Manea, C. Martín-Vide, V. Mitrana, On some algorithmic problems regarding the hairpin completion, Discrete Applied Mathematics 157 (2009) 2143–2152.
[15] F. Manea, C. Martín-Vide, V. Mitrana, Hairpin lengthening, in: F. Ferreira, B. Löwe, E. Mayordomo, L.M. Gomes (Eds.), CiE, in: Lecture Notes in Computer Science, vol. 6158, Springer, 2010, pp. 296–306.
[16] F. Manea, V. Mitrana, Hairpin completion versus hairpin reduction, in: S.B. Cooper, B. Löwe, A. Sorbi (Eds.), CiE, in: Lecture Notes in Computer Science, vol. 4497, Springer, 2007, pp. 532–541.
[17] F. Manea, V. Mitrana, T. Yokomori, Two complementary operations inspired by the DNA hairpin formation: Completion and reduction, Theoretical Computer Science 410 (2009) 417–425.
[18] F. Manea, V. Mitrana, T. Yokomori, Some remarks on the hairpin completion, Int. J. Found. Comput. Sci. 21 (2010) 859–872.
[19] V. Diekert, S. Kopecki, V. Mitrana, On the hairpin completion of regular languages, in: M. Leucker, C. Morgan (Eds.), ICTAC, in: Lecture Notes in Computer Science, vol. 5684, Springer, 2009, pp. 170–184.
[20] V. Diekert, S. Kopecki, Complexity results and the growths of hairpin completions of regular languages (extended abstract), in: M. Domaratzki, K. Salomaa (Eds.), CIAA, in: Lecture Notes in Computer Science, vol. 6482, Springer, 2010, pp. 105–114.
[21] Ch.H. Papadimitriou, Computatational Complexity, Addison–Wesley, 1994.
[22] V. Diekert, S. Kopecki, It is NL-complete to decide whether a hairpin completion of regular languages is regular, Int. J. Found. Comput. Sci. 22 (2011) 1813–1828.
[23] J.E. Hopcroft, J.D. Ulman, Introduction to Automata Theory, Languages and Computation, Addison–Wesley, 1979.
[24] J. Berstel, C. Reutenauer, Noncommutative Rational Series with Applications, Encyclopedia of Mathematics and Its Applications, Cambridge University Press, 2010.

[25] T. Ceccherini-Silberstein, On the growth of linear languages, Advances in Applied Mathematics 35 (2005) 243–253.
[26] P. Gawrychowski, D. Krieger, N. Rampersad, J. Shallit, Finding the growth rate of a regular of context-free language in polynomial time, in: M. Ito, M. Toyama (Eds.), Developments in Language Theory, in: Lecture Notes in Computer Science, vol. 5257, Springer, 2008, pp. 339–358.
[27] W. Kuich, On the entropy of context-free languages, Information and Control 16 (1970) 173–200.
[28] G. Hansel, D. Perrin, I. Simon, Compression and entropy, in: A. Finkel, M. Jantzen (Eds.), STACS, in: Lecture Notes in Computer Science, vol. 577, Springer, 1992, pp. 515–528.
[29] A. Szilard, S. Yu, K. Zhang, J. Shallit, Characterizing regular languages with polynomial densities, in: I. Havel, V. Koubek (Eds.), Mathematical Foundations of Computer Science, in: Lecture Notes in Computer Science, vol. 629, Springer, Berlin/Heidelberg, 1992, pp. 494–503, http://dx.doi.org/10.1007/3-540-55808-X_48.
[30] R.E. Tarjan, Depth-first search and linear graph algorithms, SIAM J. Comput. 1 (1972) 146–160.
[31] D.A. Lind, B. Marcus, An Introduction to Symbolic Dynamics and Coding, Cambridge University Press, New York, NY, USA, 1995.
[32] S. Kopecki, Formal language theory of hairpin formations, PhD thesis, University of Stuttgart, 2011.
[33] V. Diekert, S. Kopecki, Language theoretical properties of hairpin formations, Theoretical Computer Science 429 (2012) 65–73 (Magic in Science, Special issue dedicated to the 70th birthday of Grzegorz Rozenberg).
[34] F. Manea, R. Mercas, V. Mitrana, Hairpin lengthening and shortening of regular languages, in: Lecture Notes in Computer Science, vol. 7300, 2012, in press.