Inner Palindromic Closure*

Jürgen Dassow[1], Florin Manea[2], Robert Mercaş[1], and Mike Müller[2]

[1] Otto-von-Guericke-Universität Magdeburg, Fakultät für Informatik, PSF 4120, D-39016 Magdeburg, Germany, {dassow,mercas}@iws.cs.uni-magdeburg.de
[2] Christian-Albrechts-Universität zu Kiel, Institut für Informatik, D-24098 Kiel, Germany, {flm,mimu}@informatik.uni-kiel.de

Abstract. We introduce the inner palindromic closure as a new operation ♠, which consists in expanding a factor u to the left or right by v such that vu or uv, respectively, is a palindrome of minimal length. We investigate several language theoretic properties of the iterated inner palindromic closure $\spadesuit^*(w) = \bigcup_{i\geq 0} \spadesuit^i(w)$ of a word w.

1 Introduction

The investigation of repetitions of factors in a word is a very old topic in formal language theory. For instance, already in 1906, Thue proved that there exists an infinite word over an alphabet with three letters which has no factor of the form ww. Since the eighties a lot of papers on combinatorial properties concerning repetitions of factors were published (see [17] and the references therein).

The duplication got further interest in connection with its importance in natural languages [16] and in DNA sequences and chromosomes [18]. Motivated by these applications, grammars with derivations consisting in "duplications" (more precisely, a word $xuwvy$ is derived to $xwuwvy$ or $xuwvwy$ under certain conditions for w, u, and v) were introduced. We refer to [5, 13].

Combining the combinatorial, linguistic and biological aspect, it is natural to introduce the duplication language $D(w)$ associated to a word $w \in \Sigma^+$, which is the language containing all words that double some factor of w, i. e., $D(w) = \{xuuy \mid w = xuy, x, y \in \Sigma^*, u \in \Sigma^+\}$ and its iterated version $D^*(w) = \bigcup_{i\geq 0} D^i(w)$. In the papers [1, 4, 6, 19], the regularity of $D^*(w)$ was discussed; for instance, it was shown that, for any word w over a binary alphabet, $D^*(w)$ is regular and that $D^*(abc)$ is not regular. Further results on iterated duplication languages can be found in [10]. Also the case of bounded duplication, i. e., the length of the duplicated word is bounded by a constant, was studied, [11].

It was noted that words w containing hairpins (i. e., $w = xuyh(u^R)z$) and words w with $w = xuy$ and $u = h(u^R)$, where u^R is the mirror image of u and h is a letter-to-letter isomorphism, are of interest in DNA structures (see [8, 9], where the Watson-Crick complementarity gives the isomorphism). Therefore, operations leading to words with hairpins as factors were studied (see [2, 3]).

* The work of Florin Manea and Mike Müller is supported by the DFG grant 582014. The work of Robert Mercaş is supported by Alexander von Humboldt Foundation.

In this paper, we consider the case where the operation leads to words which have palindromes (words with $w = w^R$) as factors (which is a restriction to the identity as the isomorphism). An easy step would be to obtain xuu^Ry from a word xuy in analogy to the duplication. But then all newly obtained palindromes are of even length. Thus it seems to be more interesting to consider the palindrome closure defined by DE LUCA [12]. Here a word is extended to a palindrome of minimal length. We allow this operation to be applied to factors and call it inner palindromic closure. We also study the case of iterated applications and a restriction bounding the increase of length.

The paper is organised as follows: After some preliminaries given in the following, we define the new operation, inner palindromic closure, and its versions in Section 2, where we also give some simple properties. In Sections 3 and 4, we discuss the regularity of the sets obtained by the inner palindromic closures. Finally, we present some language classes associated with the new operation.

Basic definitions. For more details on the concepts we define here see [17].

A set $M \subseteq \mathbb{N}^m$ of vectors is called linear, if it can be represented as

$$M = \{B + \sum_{i=1}^{n} \alpha_i A_i \mid \alpha_i \in \mathbb{N}, 1 \leq i \leq n\}$$

for some vectors B and A_i, $1 \leq i \leq n$. It is called semi-linear if it can be represented as a finite union of linear sets.

An alphabet Σ is a non-empty finite set with the cardinality denoted by $\|\Sigma\|$, and the elements called letters. A sequence of letters constitute a word $w \in \Sigma^*$, and we denote the *empty word* by ε. The set of all finite words over Σ is denoted by Σ^*, and any subset of it is called a language. Moreover, for a language L, by $\mathrm{alph}(L)$ we denote the set of all symbols occurring in words of L.

If $w = u_1v_1u_2v_2 \dots u_nv_n$ and $u = u_iu_{i+1} \dots u_j$ for $1 \leq i \leq j \leq n$, we say that u is a *scattered factor* of w, denoted as $u \preccurlyeq w$. Consider now $v_k = \varepsilon$ for all $1 \leq k \leq n$. We say that u is a *factor* of w, and, if $i = 1$ we call u a *prefix*. If $j = n$ we call u a *suffix*. Whenever $i > 1$ or $j < |w|$, the factor u is called *proper*.

The length of a finite word w is the number of not necessarily distinct symbols it consists of, and is denoted by $|w|$. The number of occurrences of a certain letter a in w is designated by $|w|_a$. The *Parikh vector* of a word $w \in \Sigma^*$, denoted by $\Psi(w)$, is defined as $\Psi(w) = \langle |w|_{a_1}, |w|_{a_2}, \dots, |w|_{a_{\|\Sigma\|}} \rangle$, where $\Sigma = \{a_1, a_2, \dots, a_{\|\Sigma\|}\}$. A language L is called linear or semi-linear, if its set of Parikh vectors is linear or semi-linear, respectively.

For $i \geq 0$, the i-fold catenation of a word w with itself is denoted by w^i and is called the ith power of w. When $i = 2$, we call the word $w^2 = ww$ a square.

We say that a language L is *dense*, if, for any word $w \in \Sigma^*$, $\Sigma^* w \Sigma^* \cap L$ is non-empty, i. e., each word occurs as a factor in L.

We recall Higman's Theorem.

Theorem 1 (Higman [7]). *If L is a language such that any two words in L are incomparable with respect to the scattered factors partial order, then L is finite.*

2 Definitions and preliminary results

We now look at a word operation due to DE LUCA [12], which considers extensions to the left and right of words such that the newly obtained words are palindromes.

Definition 1. *For a word u, the left (right) palindromic closure of u is a word vu (uv) which is a palindrome for some non-empty word v such that any other palindromic word having u as proper suffix (prefix) has length greater than $|uv|$.*

Here the newly obtained words have length greater than the original one, but minimal among all palindromes that have the original word as prefix or suffix.

As for duplication and reversal, we can now define a further operation.

Definition 2. *For a word w, the left (right) inner palindromic closure of w is the set of all words $xvuy$ ($xuvy$) for any factorisation $w = xuy$ with possibly empty x, y and non-empty u, v, such that vu (uv) is the left (right) palindromic closure of u. We denote these operations by $\spadesuit_\ell(w)$ and $\spadesuit_r(w)$, respectively, and define the inner palindromic closure $\spadesuit(w)$ as the union of $\spadesuit_\ell(w)$ and $\spadesuit_r(w)$.*

The operation is extended to languages and an iterated version is introduced.

Definition 3. *For a language L, let $\spadesuit(L) = \bigcup_{w\in L} \spadesuit(w)$. We set $\spadesuit^0(L) = L$, $\spadesuit^n(L) = \spadesuit(\spadesuit^{n-1}(L))$ for $n \geq 1$, $\spadesuit^*(L) = \bigcup_{n\geq 0} \spadesuit^n(L)$. Any set $\spadesuit^n(L)$ is called a finite inner palindromic closure of L, and we say that $\spadesuit^*(L)$ is the iterated inner palindromic closure of L.*

We start with a simple observation.

Lemma 1. *For every word w, if $u \in \spadesuit^*(w)$, then $w \preccurlyeq u$.*

Remark 1. Obviously, for any language L, $\spadesuit_r^*(L) \subseteq \spadesuit^*(L)$ and $\spadesuit_\ell^*(L) \subseteq \spadesuit^*(L)$. In general, allowing both left and right operations is stronger than allowing them only in one direction. To see this we consider $L = \{ax \mid x \notin a^*\}$. The language $\spadesuit_r^*(L)$ contains only words of the form ay with $x \preccurlyeq y$, while the language $\spadesuit_\ell^*(L^R)$ contains only words of the form $y'a$ with $x^R \preccurlyeq y'$. This is not the case of the languages obtained by the application of $\spadesuit$, since we can insert either before or after the letter a a letter $b \neq a$. Thus, $\spadesuit^*(L)$ and $\spadesuit^*(L^R)$ also contain words starting and ending with $b \neq a$, respectively. Hence $\spadesuit_r^*(L) \subsetneq \spadesuit^*(L)$ and $\spadesuit_\ell^*(L^R) \subsetneq \spadesuit^*(L^R)$. ◁

The next results are in tone with the ones from [10, Proposition 3.1.1]:

Proposition 1. *For any semi-linear (linear) language, its iterated inner palindromic closure is semi-linear (respectively, linear).*

Since each word is described by a linear set, as consequence of the above we get the following assertion.

Corollary 1. *For any word, its iterated inner palindromic closure is linear.*

Furthermore, we have the following result. We omit its proof as it follows similarly to Proposition 3.

Proposition 2. *For any word w, the language $\spadesuit^*(w)$ is dense with respect to the alphabet $alph(w)$.*

We mention that Proposition 2 does not hold for languages. This can be seen from $L = \{ab, ac\}$. Obviously, any word in $\spadesuit^*(L)$ contains only a and b or only a and c. Therefore $abc \in \Sigma^*$ is not a factor of any word in $\spadesuit^*(L)$.

Lemma 2. *Let $\Sigma = \{a_1, a_2, \ldots, a_k\}$ and define the recursive sequences*

$$w'_0 = \varepsilon \text{ and } w_0 = \varepsilon,$$
$$w'_i = w_{i-1}w'_{i-1} \text{ and } w_i = w'_i a_i \text{ for } 1 \leq i \leq k.$$

Then for $1 \leq i \leq k$, $alph(w_i)^* w_i \subseteq \spadesuit^*(w_i)$.

Proof. Note that, for $0 \leq j < i \leq k$, w'_i is a palindrome and w_j is a proper prefix of w_i. We want to generate $b_1 b_2 \ldots b_n w_i$ with $b_\ell \in \text{alph}(w_i)$ for $1 \leq \ell \leq n$. Let $b_1 = a_j$. Since w_j is a prefix of w_i, $w_i = w'_j a_j v$ for some v. Since w'_j is a palindrome, we obtain $a_j w'_j a_j v = b_1 w_i$ by an inner palindromic closure step. The conclusion follows after performing the procedure in succession for $b_2, \ldots, b_n$. □

We now define a variant of the inner palindromic closure, where we restrict the length of the words which are involved in the palindromic closure. First we introduce a parametrised version of the palindromic closure operation from [12].

Definition 4. *For a word u and $m, n \in \mathbb{N}$, we define the sets*

$$L_{m,n}(w) = \{u \mid u = u^R, u = xw \text{ for } x \neq \varepsilon, |x| \geq n,\ m \geq |w| - |x| \geq 0\},$$
$$R_{m,n}(w) = \{u \mid u = u^R, u = wx \text{ for } x \neq \varepsilon, |x| \geq n,\ m \geq |w| - |x| \geq 0\}.$$

The left (right) (m,n)-palindromic closure of w is the shortest word of $L_{m,n}(w)$ (resp., $R_{m,n}(w)$), or undefined if $L_{m,n}(w)$ (resp., $R_{m,n}(w)$) is empty.

The idea behind this new definition is that an element of $L_{m,n}(w)$ is a palindrome u obtained by extending the word w by adding a prefix x of length at least n such that the centre of the newly obtained palindrome u is inside the prefix of length $\lceil \frac{m}{2} \rceil$ of w. That is, $u = xvv^R x^R$ where $n \leq |x|$, $2|v| \leq m$, and $w = vv^R x^R$, or $u = xvav^R x^R$, where $n \leq |x|$, $2|v| + 1 \leq m$, and $w = vav^R x^R$. The left (m,n)-palindromic closure is the shortest such word u, obtained when the shortest v is chosen. The right (m,n)-palindromic closure is defined similarly.

We briefly describe the restrictions imposed by (m,n) on the left palindromic closure (similar explanations hold for the right variant). By (classical) left palindromic closure we added some letters to the left of a word that had a palindromic prefix to transform the entire initial word into a palindrome of minimal length. For the left (m,n)-palindromic closure we require that at least n letters should be added and that the palindromic prefix should not be longer than m.

We define now the parametrised version of the inner palindromic closure.

Definition 5. *For non-negative integers n, m with $n > 0$, we define the $\spadesuit_{(m,n)}$ one step inner palindromic closure of some word w as*

$$\spadesuit_{(m,n)}(w) = \{u \mid u = xy'z,\ w = xyz,\ \text{and } y' \text{ is obtained by left or right } (m,n)\text{−palindromic closure from } y\}.$$

This notion can be easily extended to languages, while its iterated version $\spadesuit^*_{(m,n)}$ is defined just as in the case of the inner palindromic closure.

Remark 2. Note that $L_{m,n}(w)$ and $R_{m,n}(w)$ are empty if and only if $|w| < n$; otherwise, $L_{m,n}(w)$ contains at least the word $w^R w$ and $R_{m,n}(w)$ contains the word ww^R. Therefore, $L_{m,n}(w)$ and $R_{m,n}(w)$ are either both empty or both non-empty; clearly, both sets are always finite.

Also, the length of the left (right) $(m, n+j)$-palindromic closure of w is greater or equal than both the length of the left (right) $(m+i, n+j)$-palindromic closure of w and the length of the left (right) (m, n)-palindromic closure of w for $i, j > 0$.

If $|w| < n$, then $\spadesuit_{(m,n)}(w) = \emptyset$. Further, if $|w| = n$ then $\spadesuit_{(m,n)}(w) = \{w^R w, ww^R\}$. Generally, for $|w| \geq n$, we have that $\spadesuit_{(m,n)}(w) \neq \emptyset$. Finally, it is not hard to see that $\spadesuit(w) = \spadesuit_{(|w|,1)}(w)$. ◁

A statement similar to Proposition 2 also holds for the bounded operation.

Proposition 3. *For any word w with $|w| \geq n$ and positive integer m, the language $\spadesuit^*_{(m,n)}(w)$ is dense with respect to the alphabet alph(w).*

Proof. We note that if u is a prefix of length at least n of w and u ends with a then there is a word w' starting with a in $\spadesuit_{(m,n)}(w)$. If the letter a appears only in the prefix of length $n-1$ of w, then we do as follows. Let $w_0 = w$ and let w_{i+1} the word obtained by left (m, n)-palindromic closure from w_i for $i \geq 0$. As w_i is a proper suffix of w_{i+1}, there exists i_a such that w_{i_a} has a prefix of length at least n that ends with a. Continuing this process, we derive a word w' that for each letter $s \in \text{alph}(w)$ has a prefix of length at least n ending with s.

Suppose we want to generate a word starting with $a_1 \cdots a_n$ by inner (m, n)-palindromic closure from w. First, we generate w' and let $v_0 = w'$. By the above, $v_0 = x_1 a_1 y_1$ for some $|x_1| \geq n - 1$. Then, applying a left (m, n)-palindromic closure to $x_1 a_1$ (which produces a word from the inner (m, n)-palindromic closure of v_0) we obtain from v_0 a palindrome $a_1 v_1$, where v_1 has v_0 as a proper suffix. Thus, v_1 also has prefixes of length greater than n that end with every letter in alph(w). Next, to generate the word $a_1 a_2 v_2$ from $a_1 v_1 = a_1 x_2 a_2 y_2$ we apply a left (m, n)-palindromic closure operation to $x_2 a_2$. The process is repeated until we generate the word $a_1 \cdots a_n v_n$. □

The next result is related to Proposition 3 and will be useful in the sequel.

Lemma 3. *Let Σ be an alphabet with $||\Sigma|| \geq 2$, $a \notin \Sigma$, and m and n positive integers. Let $w = a^m y_1 a \cdots y_{p-1} a y_p$ be a word such that alph$(w) = \Sigma \cup \{a\}$, $m, p > 0$, $y_i \in \Sigma^*$ for $1 \leq i \leq p$, $|y_1| > 0$, and such that there exists $1 \leq j \leq p$ with $|y_j| \geq n$. Then, for each $v \in \Sigma^*$ with $|v| \geq n$, there exists $w' \in \spadesuit^*_{(m,n)}(w)$ such that v is a prefix of w' and $|w'|_a = |w|_a$.*

Proof. As a first step, for a word $z = z_1 a z_2 a \cdots a z_k$, where $a \notin \bigcup_{1 \leq i \leq k} \mathrm{alph}(z_i)$ and $|z_1| \geq n$, we define $z'_1 = z_1$ and $z'_i = (z'_{i-1})^R z_i$ for $1 < i \leq k$. Let $z' = z'_1 a \cdots a z'_k$. It is immediate that $z' \in \spadesuit^*_{(m,n)}(z)$, as it is obtained by applying iteratively right (m,n)-palindromic closure to the factors $z'_i a$ to get $z'_i a (z'_i)^R$, for $i > 0$. Moreover, $\mathrm{alph}(z'_k) = \bigcup_{1 \leq i \leq k} \mathrm{alph}(z_i)$ and $|z'_i| \geq n$ for all $1 \leq i \leq k$.

As a second step, for a word $v = v_\ell a v_{\ell-1} a \cdots a v_1$, where $a \notin \bigcup_{1 \leq i \leq \ell} \mathrm{alph}(v_i)$ and $|v_1| \geq n$, we define $v'_1 = v_1$ and $v'_i = v_i (v'_{i-1})^R$ for $1 < i \leq \ell$. Let $v' = v'_\ell a \cdots a v'_1$. It is immediate that $v' \in \spadesuit^*_{(m,n)}(v)$, as it can be obtained by applying iteratively left (m,n)-palindromic closure to the factors $a v'_i$ to obtain $(v'_i)^R a {v'_i}^R$, for $i > 0$. Moreover, $\mathrm{alph}(y'_\ell) = \bigcup_{1 \leq i \leq k} \mathrm{alph}(y_i)$ and $|y'_i| \geq n$ for all $1 \leq i \leq \ell$.

Now we consider the word w from our hypothesis. We apply the first step described above to the factor $y_j a y_{j+1} \cdots a y_p$ to obtain $y'_j a \cdots a y'_p$, where $\mathrm{alph}(y'_p) = \bigcup_{j \leq i \leq p} \mathrm{alph}(y'_i)$ and $|y'_p| \geq n$. Afterwards, we apply the second step procedure to the factor $y_1 a y_2 a \cdots a y_{j-1} a y'_j a \cdots a y'_p$ to obtain $y''_1 a y''_2 a \cdots a y''_{j-1} a y''_j a \cdots a y''_p$, where $\mathrm{alph}(y''_1) = \bigcup_{1 \leq i \leq p} \mathrm{alph}(y_i) = \Sigma$ and $|y''_1| \geq n$. Accordingly, $w'' = a^m y''_1 a y''_2 a \cdots a y''_p \in \spadesuit^*_{(m,n)}(w)$.

Now, for a word $v \in \Sigma^*$ we obtain the word $w''_v = a^m v^R y_v a \cdots y''_p a$ from w'', for some $y_v \in \Sigma^*$, just like in the proof of Proposition 3. If $|v| \geq n$, we can obtain from w''_v the word $v w''_v$ by applying to $a^m v^R$ a left (m,n)-palindromic closure to get $v a^m v^R$. This concludes our proof. □

3 On the regularity of the inner palindromic closure

We start with some facts on words over a binary alphabet.

Lemma 4. *[Propagation rule] For a word $w = a^n b^m$ with positive integers n and m, the set $\spadesuit(w)$ contains all words of length $n + m + 1$ with a letter $x \in \{a, b\}$ inserted at any position i of w, where $0 \leq i < n + m$.*

Proof. To see this, assume we want to insert a letter a somewhere in w (the case of the insertion of a letter b is symmetric). To insert a between positions j and $j + 1$ with $j < n$ we just take the palindromic prefix a^{n-j} and perform a $\spadesuit_\ell$ step on it. This results in the word a^{n-j+1} which fulfils the conditions. When $n \leq j \leq m$, we perform a $\spadesuit_r$ step on the word $a b^{j-n}$, which produces the palindrome $a b^{j-n} a$. □

As a consequence of the Propagation Rule, we can show that the necessary condition given in Lemma 1 is also sufficient in the case of binary alphabets.

Corollary 2. *For any binary words w and u, $w \preccurlyeq u$ if and only if $u \in \spadesuit^*(w)$.*

Proof. By Lemma 1, we have that $w \preccurlyeq u$ for all $u \in \spadesuit^*(w)$. Using Lemma 4, all words u with $w \preccurlyeq u$ are in fact in $\spadesuit^*(w)$ since in each of them we can insert a's and b's at arbitrary positions. □

For the duplication operation, Bovet and Varricchio [1] showed that for any binary language, its iterated duplication completion always gives a regular language. For the inner palindromic closure operation on such alphabets, the result is similar.

Theorem 2. *The iterated inner palindromic closure of a language over a binary alphabet is regular.*

Proof. According to Theorem 1, for a language L there exists a finite set L_0 with $L_0 \subseteq L$ such that for every word $w \in L$ there is a word $w_0 \in L_0$ with $w_0 \preccurlyeq w$. By Corollary 2, it follows that $\spadesuit^*(L)$ is the union of the sets $SW(w_0) = \{w' \in \text{alph}(w_0)^* \mid w_0 \preccurlyeq w'\}$, for all $w_0 \in L_0$. As all the sets $SW(w_0)$ are regular, it follows that $\spadesuit^*(L)$ is regular. □

It is obvious that the finite inner palindromic closure of some finite language is always regular, since at each step we only obtain words which have at most twice the length of the longest word in the given language. However, when considering the entire class of regular languages the result is not necessarily regular.

Theorem 3. *The finite inner palindromic closure of a regular language is not necessarily regular.*

Proof. We take a positive integer k and a language $L = c_1a_1^+c_2a_2^+\ldots c_ka_k^+b$. We intersect $\spadesuit^k(L)$ with the language given by the regular expression:

$$c_1a_1^+c_2a_2^+\ldots c_ka_k^+b(a_k^+c_k\ldots a_2^+c_2a_1^+c_1)(a_k^+c_k\ldots a_3^+c_3a_2^+c_2)\ldots(a_k^+c_ka_{k-1}^+c_{k-1})a_k^+c_k$$

It is not hard to see that in any word of the intersection the number of a_i's in every maximal unary group adjacent to c_i is the same. Since this is a non-regular language and regular languages are closed under intersection, we conclude. □

It remains an *open problem* whether or not the iterated inner palindromic closure of a regular language L, where $\|\text{alph}(L)\| \geq 3$, is also regular.

We mention that the non-regularity of $\spadesuit^*(L)$ with $\|\text{alph}(L)\| \geq 3$ cannot be obtained by a strategy similar to that by Wang [19], who showed the non-regularity of $D(L)$ with $\|\text{alph}(L)\| \geq 3$. There, the non-regularity of $D(L)$ comes as a consequence of a padding that needs to be added every time we want to construct a longer word as result of consecutive applications of our chosen rule. Consider now the word abc and the language $(abc)^*$ that contains no palindromes of length greater than one. However, $babc \in \spadesuit(abc)$, thus by Lemma 2 we can generate at the beginning as many abc's as we want, $(abc)^*babc$. Hence, we cannot use any more the argument that each palindromic step creates some extra padding at the end of the word whenever we investigate words that contain no palindromes.

4 Parametrised inner palindromic closure

We now discuss the regularity of $\spadesuit^*_{(m,n)}(w)$. Before we state our results, we establish two facts on the avoidance of patterns.

Theorem 4. *There exist infinitely long binary words avoiding both palindromes of length* 6 *and longer, and squares of words with length* 3 *and longer.*

Proof. Rampersad et al. [15] constructed an infinite word w, that is square-free and has no factors from the set $\{ac, ad, ae, bd, be, ca, ce, da, db, eb, ec, aba, ede\}$.

We can show that the morphism γ, defined by

$$\gamma(a) = abaabbab, \qquad \gamma(b) = aaabbbab, \qquad \gamma(c) = aabbabab,$$
$$\gamma(d) = aabbbaba, \qquad \gamma(e) = baaabbab,$$

maps this word w to a word with the desired properties.

As any palindrome of length $n > 2$ contains a shorter palindrome of length $n-2$, a word avoiding palindromes of lengths 6 and 7 also avoids longer ones. Also, each palindrome of length 6 or 7 would occur in the image of some word of length 2. We see that no such palindromes occur in $\gamma(\{ab, ba, bc, cb, cd, dc, de, ea, ed\})$, therefore neither in $\gamma(w)$. We show that $\gamma(w)$ contains no squares other than $aa, bb, abab$ and $baba$ by applying methods used in [15]. □

Theorem 5. *There exist infinitely long ternary words avoiding both palindromes of length* 3 *and longer, and squares of words with length* 2 *and longer.*

Proof. We claim that the morphism ψ, that is defined by

$$\psi(a) = abbccaabccab, \qquad \psi(b) = bccaabbcaabc, \qquad \psi(c) = caabbccabbca,$$

maps all infinite square-free ternary words h to words with the desired properties.

We see that $\psi(h)$ does not contain palindromes of length 3 or 4, since those would occur inside $\psi(u)$ for some square-free word u of length 2. We check that there are no squares other than aa, bb and cc in $\psi(h)$ using standard tools. □

In the sequel, we exhibit a method to construct words whose iterated inner (m, n)-palindromic closure is not regular, for positive integers m, n. We first establish several notations. We associate to an integer $k \geq 2$ a pair of numbers (p_k, q_k) if there exists an infinite word over a k-letter alphabet avoiding both palindromes of length greater or equal to q_k and squares of words of length greater or equal to p_k. If more such pairs exist, we take (p_k, q_k) to be any of them.

Theorem 6. *Let $m > 0$ and $k \geq 2$ be two integers and define $n = \max\{\frac{q_k}{2}, p_k\}$. Let Σ be a k-letter alphabet with $a \notin \Sigma$ and $w = a^m y_1 a y_2 \cdots a y_{r-1} a y_r$ be a word such that $alph(w) = \Sigma \cup \{a\}$, $r > 0$, $y_i \in \Sigma^*$ for all $1 \leq i \leq r$, and there exists j with $1 \leq j \leq r$ and $|y_j| \geq n$. Then $\spadesuit^*_{(m,n)}(w)$ is not regular.*

Proof. Let α be an infinite word over Σ that avoids palindromes of length q_k and squares of words of length p_k. Note that due to Lemma 3, for each prefix u of α longer than n, there exists w_u with $|w_u|_a = r - 1$ such that $ua^m w_u \in \spadesuit^*_{(m,n)}(w)$.

We analyse how the words $ua^m v$ with u being a prefix of α and $|v|_a = r - 1$ are obtained by iterated (m, n)-palindromic closure steps from w. As u contains no a's, no squares of words of length p_k, as well as no palindromes with length greater than q_k, and the application of an (m, n)-palindromic closure step introduces a palindrome in the derived word, we get that the only possible cases of application of the operation in the derivation of $ua^m v$ are the following:

(1) $v = xyz$ and y is the (m,n)-palindromic closure of y' (implicitly, $|y'| < |y|$ and $|y|_a = |y'|_a$); in this case we have that $ua^m v$ is in $\spadesuit_{(m,n)}(ua^m xy'z)$.
(2) $u = u'x$, $v = yz$, and $xa^m y$ is the (m,n)-palindromic closure of $a^m y$ (implicitly, $x = y^R$ and neither x nor y contain any a's); in this case we have that $ua^m v$ is in $\spadesuit_{(m,n)}(u'a^m yz)$.
(3) $u = xyz$ and y is the (m,n)-palindromic closure of y' (implicitly, $|y'| < |y|$ and y' contains no a's); in this case we have that $ua^m v$ is in $\spadesuit_{(m,n)}(xy'za^m v)$.

Since we only apply (m,n)-palindromic closure operations, and the word we want to derive has the form $ua^m v$ with $|a^m v|_a = |w|_a$, it is impossible to apply any palindromic closure step that adds to the derived word more a symbols or splits the group a^m that occurs at the beginning of w. Intuitively, the palindromic closure operations that we apply are localised, due to the restricted form of the operation: they either occur inside u, or inside v, or are centred around a^m.

Moreover, by choosing $n \geq \frac{q_k}{2}$ if at any step we apply a palindromic closure operation of the type (3) above, then the final word u contains a palindrome of length greater than q_k. To see this, we assume, for the sake of a contradiction, that such an operation was applied. Then, we look at the last operation of this kind that was applied. Obviously, none of the operations of type (1) or (2) that were applied after that operation of type (3) could have modified the palindrome of length at least q_k introduced by it in the derived word before a^m. Therefore, that palindrome would also appear in u, a contradiction.

This means that all the intermediate words obtained during the derivation of $ua^m v$ from w have the form $u'a^m v'$ where u' is a prefix (maybe empty) of α and v' has exactly $|w|_a - m$ symbols a. We now look at the kind of operations that can be applied to such a word. In particular, we note that we cannot have more than $|v'| - n$ consecutive derivation steps in which the length of the word occurring after the first sequence of a's is preserved. In other words, we can apply at most $|v'| - n$ consecutive operations that fall in the situation (2).

Indeed, after ℓ such derivation steps one would obtain from $u'a^m v'$ a word $u'v_1 \cdots v_\ell a^m v'$ where v_i^R is a prefix of v' and $|v_i| \geq n$ for every $1 \leq i \leq \ell$. Assume, for the sake of a contradiction, that $\ell > |v'| - n$. Then, there exists j such that $1 \leq j < \ell$ and $|v_j| \geq |v_{j+1}|$. Therefore, $u'v_1 \cdots v_\ell$ contains a square of length at least $2n \geq 2p_k$. But such a square will remain in the derived word for the rest of the derivation, as neither an operation of type (1) nor one of type (2) could introduce letters inside it. Another contradiction with the form of u is reached.

We use this last remark to show by induction on the number of steps in the derivation, that if u is a finite prefix of α and $ua^m v \in \spadesuit^*_{(m,n)}(w)$, then $|u| \leq |v|^3$.

If the derivation has one step, then the statement we want to show holds trivially, as the fact that the prefix u can be added to w implies that $|u| \leq |y_1|$.

Let us now assume that it holds for words obtained in at most k derivation steps, and show it for words obtained in $k+1$ derivation steps. If the last applied step to obtain $ua^m v$ is of type (1), then we obtained $ua^m v$ from $ua^m v'$ for some v' shorter than v. From the induction step we have that $|u| \leq |v'|^3$, and, consequently, $|u| \leq |v|^3$. According to the last made remark, we have that at most the last $|v| - n$ consecutive steps applied were of type (2). In these steps,

the length of u increased by at most $\sum_{n\leq i\leq |v|} i \leq \frac{|v|(|v|+1)}{2}$. Therefore, we get $|u| - \frac{|v|(|v|+1)}{2} \leq (|v|-1)^3$; hence $|u| \leq |v|^3$. This concludes our induction proof.

We now show that the language

$$L = \{ua^m v \in \spadesuit^*_{(m,n)}(w) \mid |u| \geq n, |v|_a = r-1\}$$

is not regular. Since this language is obtained from $\spadesuit^*_{(m,n)}(w)$ by intersection with a regular language, if L is not regular, then $\spadesuit^*_{(m,n)}(w)$ is not regular either.

We consider a word $u_0 a^m v_0 \in L$ such that u_0 is a prefix of α with $|u_0| \geq n$; clearly, L contains such a word. As we have shown above, $|u_0| \leq |v_0|^3$. We now take a prefix u_1 of α with $|u_1| > |v_0|^4$; it follows that $u_1 a^m v_0 \notin L$, thus u_0 and u_1 are in different equivalence classes with respect to the syntactic congruence defined by the language L. However, by the considerations made at the beginning of this proof, there exists v_1 such that $u_1 a^m v_1 \in L$. In the exact same manner we construct a word u_2, that is in a different equivalence class with respect to the syntactic congruence defined by the language L than both u_0 and u_1, and so on. This means we have an infinite sequence $(u_i)_{i\geq 0}$ where any two elements are in different equivalence classes with respect to the syntactic congruence defined by the language L. Thus, the syntactic congruence defined by L has an infinite number of equivalence classes, so L cannot be regular, and we conclude the proof. □

The following theorem follows immediately from the previous results.

Theorem 7. *Let $w = a^p y_1 a \cdots y_{r-1} a y_r$, where $a \notin alph(y_i)$ for $1 \leq i \leq r$.*
(1) If $\|alph(w)\| \geq 3$ and $|y_j| \geq 3$ for some $1 \leq j \leq r$, then for every positive integer $m \leq p$ we have that $\spadesuit^*_{(m,3)}(w)$ is not regular.
(2) If $\|alph(w)\| \geq 4$ and $|y_j| \geq 2$ for some $1 \leq j \leq r$, then for every positive integer $m \leq p$ we have that $\spadesuit^*_{(m,2)}(w)$ is not regular.
(3) If $\|alph(w)\| \geq 5$, then for every positive integer $m \leq p$ we have that $\spadesuit^*_{(m,1)}(w)$ is not regular.
(4) For every positive integers m and n there exists u with $\spadesuit^*_{(m,n)}(u)$ not regular.

Proof. By Theorems 4 and 5 we can take $q_2 = 6$ and $p_2 = 3$, respectively, $q_3 = 3$ and $p_3 = 2$. Therefore, if we take $n = 3$, or $n = 2$, respectively, in the hypothesis of the theorem, then the results (1) and (2) follow for any positive $m \leq p$.

The third statement follows from [14, Theorem 4.15], where an infinite word avoiding both squares and palindromes is constructed. Thus, we can take $p_k = q_k = 1$, so n can be also taken to be 1. Finally, (4) is a consequence of (3). □

In general, the regularity of the languages $\spadesuit^*_{(m,n)}(w)$ for positive integers m and n, and binary words w, $|w| \geq n$, is *left open*. We only show the following.

Theorem 8. *For any word $w \in \{a,b\}^+$ and integer $m \geq 0$, $\spadesuit^*_{(m,1)}(w)$ is regular.*

Proof. Due to the lack of space the technical details are skipped.

The general idea of the proof is to give a recursive definition of $\spadesuit^*_{(m,1)}(w)$. That is, $\spadesuit^*_{(m,1)}(w)$ is expressed as a finite union and concatenation of several

languages $\spadesuit^*_{(m,1)}(w')$, with $|w'| < |w|$, and some other simple regular languages. To this end, we let $x \neq y \in \{a, b\}$ and identify a series of basic cases for which such a definition can be given easily: words that have no unary factor longer than m, words of the form xy^qx, and, finally, words of the form xy^q or y^qx. Building on these basic ingredients, we define $\spadesuit^*_{(m,1)}(w)$ for every word w by, basically, identifying a prefix of w that has one of these forms, separating it from the rest, and then computing, recursively, the iterated closure of the rest of the word.

In order to make this strategy work, one has to implement several steps. The first is to note that if a word w has no maximal unary factor longer than m, then $\spadesuit^*_{(m,1)}(w)$ contains all words that have w as scattered factor.

Further, if $uvxy^pxv^R \in \spadesuit_{(m,1)}(uvxy^q)$ for $q \leq p$, then we can find a sort of normal-form derivation of $uvxy^pxv^R$ by first deriving $uvxy^px$ in one step, and then appending any suffix (in particular v^R) by a process similar to propagation. Similar arguments hold when the factor is prefixed by palindromic closure. Intuitively, we can split the derivation of a word in separate parts and apply our operations only to maximal unary factors and the symbols that bound them (factors of the type xy^qx, y^qx, and xy^q, with the last two as suffixes or prefixes).

Next, the derivation of these basic factors on which the operation is applied can be further normalised. The basic idea is, intuitively, that whenever we start a derivation of a factor xy^qx, the first step that we should make is to split the group of y's in two smaller groups, and continue to derive each of them separately. More precisely, if $x^{\ell_1}y^{h_1}x^{\ell_2}y^{h_2}x^{\ell_3} \in \spadesuit^*_{(m,1)}(xy^qx)$ for some positive integers ℓ_1, ℓ_2, ℓ_3, h_1, and h_2, then there exist positive integers $p, r < q$ such that $x^{\ell_1}y^{h_1}x^{\ell_2}y^{h_2}x^{\ell_3} \in \spadesuit^*_{(m,1)}(xy^pxy^rx)$ and one of the following holds: $p \leq m$ or $r \leq m$ and $p = q - r$; or, $m < p, r$, and $p = m + 2k$ and $r = q - m - k$, or, vice-versa, $r = m + 2k$ and $p = q - m - k$, for some $k > 0$.

Similarly, when we start a derivation from a group xy^q, we first split the group of y's into xy^px and xy^r, with $r < q$, and then apply the above definition to these and repeat the process. Clearly, at every step we can lengthen the words by pumping x's in a group of x's, and by generating $\{a, b\}^*xy\{a, b\}^*$ from xy.

Using all the above, we can now find recursively the formula for $\spadesuit^*_{(m,1)}(w)$ by first separating a prefix having one of the basic forms, derive a word from it as we described, and then work, recursively, on the remaining suffix. □

5 Final remarks

Apart from solving the open problems stated in this article, the study of classes of languages obtained through these operations seems interesting to us. The following initial results show several possible directions for such investigations.

For a class $\mathcal{L}$ of languages, we set $\mathcal{L}^R = \{L^R \mid L \in \mathcal{L}\}$, and for a natural number $k \geq 1$, we define $\mathcal{L}_k = \{L \in \mathcal{L} \mid \|\text{alph}(L)\| = k\}$. Consider the classes

$$\mathcal{P}_{\spadesuit_\ell} = \{L' \mid L' = \spadesuit^*_\ell(L) \text{ for some } L\}$$
$$\mathcal{P}_{\spadesuit_r} = \{L' \mid L' = \spadesuit^*_r(L) \text{ for some } L\}$$
$$\mathcal{P}_{\spadesuit} = \{L' \mid L' = \spadesuit^*(L) \text{ for some } L\}$$

Straightforward, for every language L, $\spadesuit_r(L) = (\spadesuit_\ell(L^R))^R$ and $\spadesuit_r^*(L) = (\spadesuit_\ell^*(L^R))^R$ hold (for both operations the propagation rule works in only one direction). Thus, we immediately get $\mathcal{P}_{\spadesuit_r} = (\mathcal{P}_{\spadesuit_\ell})^R$ and $\mathcal{P}_{\spadesuit_\ell} = (\mathcal{P}_{\spadesuit_r})^R$.

The following result is a consequence of Remark 1.

Lemma 5. *The classes $\mathcal{P}_{\spadesuit_r} \setminus \mathcal{P}_{\spadesuit}$ and $\mathcal{P}_{\spadesuit_\ell} \setminus \mathcal{P}_{\spadesuit}$ are both not empty.*

When we consider only binary alphabets, we have the following statement.

Proposition 4. *$(\mathcal{P}_{\spadesuit})_2 \subsetneq (\mathcal{P}_{\spadesuit_r})_2 = (\mathcal{P}_{\spadesuit_\ell})_2^R$ and $(\mathcal{P}_{\spadesuit})_2 \subsetneq (\mathcal{P}_{\spadesuit_\ell})_2 = (\mathcal{P}_{\spadesuit_r})_2^R$.*

References

1. Bovet, D.P., Varricchio, S.: On the regularity of languages on a binary alphabet generated by copying systems. Inf. Process. Lett. 44, 119–123 (1992)
2. Cheptea, D., Martín-Vide, C., Mitrana, V.: A new operation on words suggested by DNA biochemistry: Hairpin completion. Trans. Comput. pp. 216–228 (2006)
3. Dassow, J., Holzer, M.: Language families defined by a ciliate bio-operation: hierarchies and decision problems. Int. J. Found. Comput. Sci. 16(4), 645–662 (2005)
4. Dassow, J., Mitrana, V., Păun, G.: On the regularity of duplication closure. Bulletin of the EATCS 69, 133–136 (1999)
5. Dassow, J., Mitrana, V., Salomaa, A.: Context-free evolutionary grammars and the structural language of nucleic acids. BioSystems 43, 169–177 (1997)
6. Ehrenfeucht, A., Rozenberg, G.: On regularity of languages generated by copying systems. Discrete Appl. Math. 8, 313–317 (1984)
7. Higman, G.: Ordering by divisibility in abstract algebras. Proc. London Math. Soc. 3(2), 326–336 (1952)
8. Kari, L., Konstantinides, S., Losseva, E., Sosik, P., Thierrin, G.: Hairpin structures in DNA words. In: DNA 2005. LNCS, vol. 3892, pp. 158–170. Springer (2006)
9. Kari, L., Mahalingam, K.: Watson–Crick palindromes in DNA computing. Natural Computing 9(2), 297–316 (2010)
10. Leupold, P.: Languages Generated by Iterated Idempotencies. Ph.D. thesis, Universitat Rovira y Virgili, Tarragona, Spain (2006)
11. Leupold, P., Mitrana, V.: Uniformly bounded duplication codes. RAIRO Theor. Inf. Appl. 41, 411–427 (2007)
12. de Luca, A.: Sturmian words: Structure, combinatorics, and their arithmetics. Theor. Comput. Sci. 183, 45–82 (1997)
13. Martín-Vide, C., Păun, G.: Duplication grammars. Acta Cybernet. 14, 151–164 (1999)
14. Pansiot, J.J.: A propos d'une conjecture de F. Dejean sur les répétitions dans les mots. Discrete Appl. Math. 7, 297–311 (1984)
15. Rampersad, N., Shallit, J., Wang, M.W.: Avoiding large squares in infinite binary words. Theor. Comput. Sci. 339(1), 19–34 (2005)
16. Rounds, W., Ramer, A.M., Friedman, J.: Finding natural languages a home in formal language theory. In: Mathematics of Languages. pp. 349–360. John Benjamins, Amsterdam (1987)
17. Rozenberg, G., Salomaa, A.: Handbook of Formal Languages. Springer-Verlag New York, Inc. (1997)
18. Searls, D.: The computational linguistics of biological sequences. In: Artificial Intelligence and Molecular Biology. pp. 47–120. AAAI Press, Cambridge (1993)
19. Wang, M.W.: On the irregularity of the duplication closure. Bulletin of the EATCS 70, 162–163 (2000)