

# A Quasilinear-Time Algorithm for Tiling the Plane Isohedrally with a Polyomino

Stefan Langerman<sup>\*1</sup> and Andrew Winslow<sup>2</sup>

- 1 Département d'Informatique, Université Libre de Bruxelles, Bruxelles, Belgium  
stefan.langerman@ulb.ac.be
- 2 Département d'Informatique, Université Libre de Bruxelles, Bruxelles, Belgium  
andrew.winslow@ulb.ac.be

---

## Abstract

A plane tiling consisting of congruent copies of a shape is *isohedral* provided that for any pair of copies, there exists a symmetry of the tiling mapping one copy to the other. We give a  $O(n \log^2 n)$ -time algorithm for deciding if a polyomino with  $n$  edges can tile the plane isohedrally. This improves on the  $O(n^{18})$ -time algorithm of Keating and Vince and generalizes recent work by Brlek, Provençal, Fédou, and the second author.

**1998 ACM Subject Classification** F.2.2 Nonnumerical Algorithms and Problems

**Keywords and phrases** Plane tiling, polyomino, boundary word, isohedral

**Digital Object Identifier** 10.4230/LIPIcs.SoCG.2016.50

## 1 Introduction

The 18th of Hilbert's 23 famous open problems posed in 1900 [20] concerned *isohedral* tilings of polyhedra where every pair of copies in the tiling has a symmetry of the tiling that maps one copy to the other (see Figure 1). Hilbert asked for an example of an *anisohedral* polyhedron that admits a tiling, but no isohedral tilings.

Reinhardt [30] was the first to give an example of an anisohedral polyhedron. Along with this example, Reinhardt also stated that a proof that no anisohedral polygons exist was forthcoming, a claim thought to be supported by Hilbert [15]. In fact, Reinhardt (and Hilbert?) were mistaken: no such proof is possible and Heesch provided the first counterexample in 1935 [18] (see Figure 2).

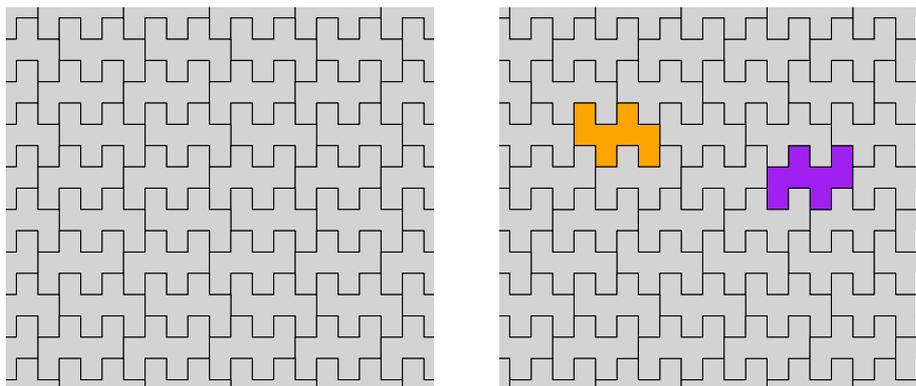
In the 1963, Heesch and Kienzle [19] provided the first complete classification of isohedral tilings. This classification was given as nine *boundary criteria*: conditions on a polygon's boundary that, if satisfied, imply an isohedral tiling and together form a necessary condition for isohedral polygons. Each boundary criterion describes a factorization of the boundary into a specific number of intervals with given properties, e.g., an interval is rotationally symmetric or two intervals are translations of each other. Special cases of this classification have been rediscovered since, including the criterion of Beauquier and Nivat [3] and Conway's criterion, attributed to John H. Conway by Gardner [11, 32] (see Figure 3).

While a complete classification of isohedral tilings exists, many problems in tiling classification and algorithmics remain open. For instance, complete classifications of pentagons

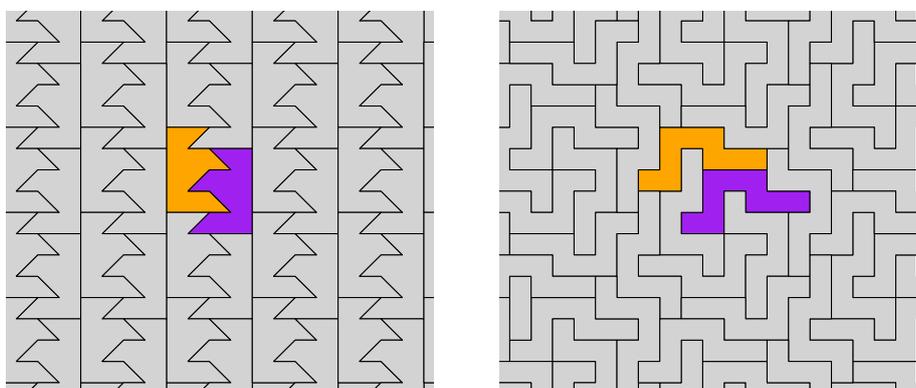
---

\* Directeur de recherches du F.R.S.-FNRS.





■ **Figure 1** Isohedral (left) and anisohedral (right) tilings of a polyomino. There is no symmetry of the right tiling mapping one colored tile to the other.

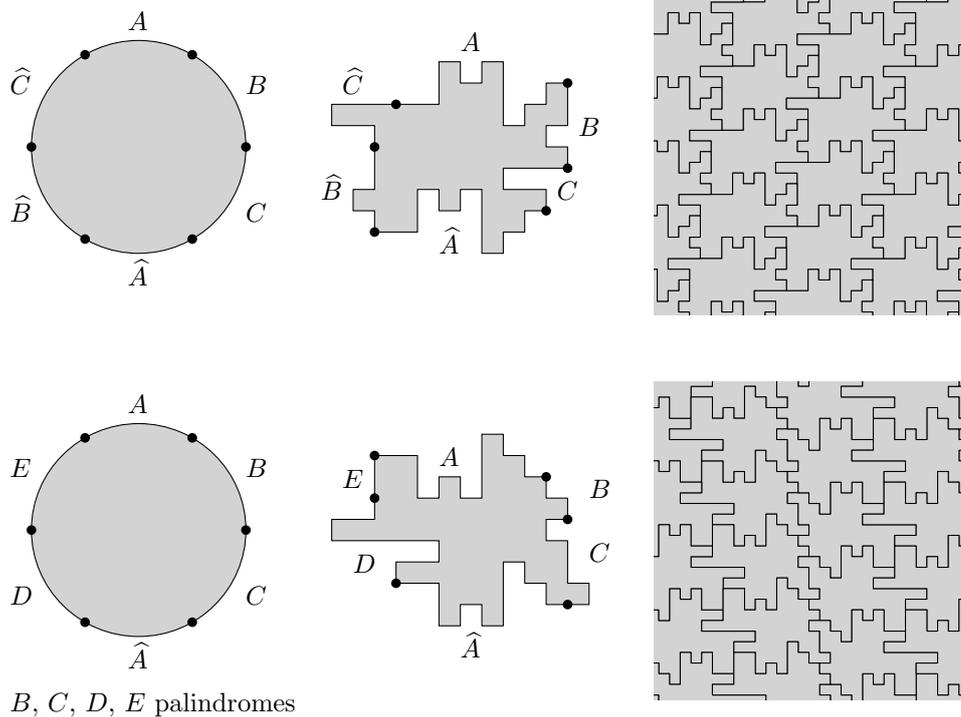


■ **Figure 2** The anisohedral polygon of Heesch [18] and an anisohedral polyomino of Rhoads [31]. There is no symmetry of either tiling mapping one colored tile to the other.

that tile the plane were claimed as early as 1968 [23], but additional pentagons have been discovered as recently as 2015 [26]. The existence of an algorithm for deciding if a polyomino tiles the plane is a longstanding open problem [12, 13], as is the existence of a polygon that tiles only without symmetry [34].

One of the most successful lines of work in tiling algorithmics was initiated by Wijshoff and van Leeuwen [35], who considered tiling the plane using *translated* copies of a polyomino (isohedrally or otherwise). They proved that deciding whether a polyomino admits such a tiling is possible in polynomial time. Their algorithm was subsequently improved by Beauquier and Nivat [3], who gave a simple boundary criterion for polyominoes that admit such a tiling. Subsequent application of more advanced algorithmic techniques led to a sequence of improved algorithms by Gambini and Vuillon [10], Provençal [29], Brlek, Provençal, and Fédou [5], and the second author [36], who gave an optimal  $O(n)$ -time algorithm, where  $n$  is the number of edges on the polyomino's boundary.

The boundary criterion of Beauquier and Nivat matches one of the criteria of Heesch and Kienzle, implying that this problem is a special case of deciding if a polyomino is isohedral. The general problem of isohedrality was proved decidable in 1999 by Keating and Vince [22],



**Figure 3** Two of seven boundary criteria characterizations of isohedral tilings. These criteria were given by Beauquier and Nivat [3] (top) and John H. Conway [11] (bottom). Precise definitions are given in Sections 2.

who gave a matrix-based algorithm running in  $O(n^{18})$  time. Their algorithm does not make use of boundary criteria, which we note yields a straightforward  $O(n^6)$ -time algorithm.

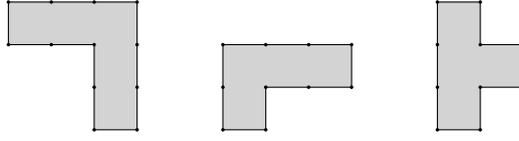
Here we give a  $O(n \log^2 n)$ -time algorithm for deciding if a polyomino is isohedral. The algorithm uses the original boundary characterization of Heesch and Kienzle [19] to decompose the problem into seven subproblems, each of recognizing whether a polyomino's boundary admits a factorization with a specific form. Structural and algorithmic results on a variety of word problems are used, extending the approach of [36] to factorizations of six additional forms. The algorithm also finds a witness tiling and is easily extended to other classes of lattice shapes, e.g. polyhexes and polyiamonds.

## 2 Definitions

Although the main result of the paper concerns geometric tilings, the proof is entirely described using *words*, also called *strings*. We use the term “word” for consistency with terminology in previous work on tilings of polyominoes.

### Polyomino and Tiling

A *polyomino* is a simply connected polygon whose edges are unit length and parallel to one of two perpendicular lines. Let  $\mathcal{T} = \{T_1, T_2, \dots\}$  be an infinite set of finite simply connected closed sets of  $\mathbb{R}^2$ . Provided the elements of  $\mathcal{T}$  have pairwise disjoint interiors and cover the Euclidean plane, then  $\mathcal{T}$  is a *tiling* and the elements of  $\mathcal{T}$  are called *tiles*. Provided every  $T_i \in \mathcal{T}$  is congruent to a common shape  $T$ , then  $\mathcal{T}$  is *monohedral* and  $T$  is the *prototile* of  $\mathcal{T}$ .



■ **Figure 4** Polyominoes with (circular) boundary words  $\mathbf{ur^3d^3lu^2l^2}$ ,  $\mathbf{u^2r^3dl^2dl}$ , and  $\mathbf{u^3(rd)^2ldl}$  (from left to right).

In this case,  $T$  is said to *have* a tiling. A monohedral tiling is also *isohedral* provided, for every pair of elements  $T_i, T_j \in \mathcal{T}$ , there exists a symmetry of  $\mathcal{T}$  that maps  $T_i$  to  $T_j$ . Otherwise the tiling is *anisohedral*.

### Letter

A *letter* is a symbol  $x \in \Sigma = \{\mathbf{u}, \mathbf{d}, \mathbf{l}, \mathbf{r}\}$  representing the directions up, down, left and right. The  $\Theta^\circ$ -rotation of a letter  $x$ , written  $t_\Theta(x)$ , is defined as the letter obtained by rotating  $x$  counterclockwise by  $\Theta^\circ$ , e.g.,  $t_{270}(\mathbf{u}) = \mathbf{r}$ . A special case of  $\Theta^\circ$ -rotations is the *complement* of a letter, written  $\bar{x}$  and defined as  $\bar{x} = t_{180}(x)$ .

### Word and Boundary Word

A *word* is a sequence of letters and the *length* of a word  $W$ , denoted  $|W|$ , is the number of letters in  $W$ . For an integer  $i \in \{1, 2, \dots, |W|\}$ ,  $W[i]$  refers to the  $i$ th letter of  $W$  and  $W[-i]$  refers to the  $i$ th from the last letter of  $W$ . The notation  $W^i$  denotes the repetition of a word  $i$  times. In this work two kinds of words are used: non-circular words and circular words (defining the boundaries of polyominoes). A word is *non-circular* if it has a first letter, and *circular* otherwise. For a circular word  $W$ , an arbitrary but fixed assignment of the letter  $W[1]$  may be used, resulting in a non-circular *shift* of  $W$ . The *boundary word* of a polyomino  $P$ , denoted  $\mathcal{B}(P)$ , is the circular word of letters corresponding to the sequence of directions traveled along cell edges during a clockwise traversal of the polyomino's boundary (see Figure 4).

### Rotation and complement

The rotation (or complement) of a word  $W$ , written  $t_\Theta(W)$  (or  $\overline{W}$ ), is the word obtained by replacing each letter in  $W$  with its rotation (or complement). The *reverse* of a word  $W$ , written  $\widetilde{W}$ , are the letters of  $W$  in reverse order. The *backtrack* of a word  $W$  is denoted  $\widehat{W}$  and defined as  $\widehat{W} = \widetilde{\overline{W}}$ .

### Factor

A *factor* of  $W$  is a contiguous sequence  $X$  of letters in  $W$ , written  $X \preceq W$ . For integers  $1 \leq i, j \leq |W|$  with  $i \leq j$ ,  $W[i..j]$  denotes the factor of  $W$  from  $W[i]$  to  $W[j]$ , inclusive. A factor  $X$  *starts* or *ends* at  $W[i]$  if  $W[i]$  is the first or last letter of  $X$ , respectively. Two factors  $X, Y \preceq W$  may refer the same letters of  $W$  or merely have the same letters in common. In the former case,  $X$  and  $Y$  are *equal*, written  $X = Y$ , while in the latter,  $X$  and  $Y$  are *congruent*, written  $X \equiv Y$ . For instance, if  $W = \mathbf{uuulruuu}$  then  $W[1..3] \equiv W[6..8]$ . A *factorization* of  $W$  is a partition of  $W$  into consecutive factors  $F_1$  through  $F_k$ , written  $W = F_1F_2 \dots F_k$ .

### Prefix, suffix, and center

A factor  $X \preceq W$  is a *prefix* if  $X$  starts at  $W[1]$ , written  $X \preceq_{\text{pre}} W$ . Similarly,  $X \preceq W$  is a *suffix* if  $X$  ends at  $W[-1]$ , written  $X \preceq_{\text{suff}} W$ . The factor  $X \preceq W$  such that  $W = UXV$ ,  $|U| = |V|$ , and  $|X| \in \{1, 2\}$  is the *center* of  $W$ . Similar definitions for words are defined equivalently, e.g., a word is a *prefix* of another word provided it is congruent to a prefix factor of that word.

### Period, composite, and primitive

A word  $X$  is a *period* of  $W$  provided  $W$  is congruent to a prefix of  $X^k$  for some  $k \geq 0$  (introduced by [24]). Alternatively,  $X$  is a prefix of  $W$  and  $W[i] = W[i + |X|]$  for all  $1 \leq i \leq |W| - |X|$ . A word  $X$  is *composite* provided there exists a subword  $Y$  such that  $X = Y^k$  for some  $k \geq 2$ , and otherwise is *primitive*.

### $\Theta$ -drome and gapped mirror.

A word  $X$  is a  $\Theta$ -*drome* provided  $X = Yt_{\Theta+180}(\tilde{Y})$ , e.g., a *palindrome* is a 180-drome. Such a factor is *admissible* provided  $W = XU$  with  $U[-1] \neq t_{\Theta+180}(U)[1]$ .<sup>1</sup>

A pair of disjoint factors  $X, Y \preceq W$  is a *gapped mirror* provided  $X \equiv \hat{Y}$ . Such a pair  $X, Y$  is *admissible* provided  $W = XUYV$  with  $U[1] \neq \overline{U[-1]}$ ,  $V[1] \neq \overline{V[-1]}$ .

## 3 Proof Overview

The remainder of the paper is dedicated to proving the following theorem:

► **Theorem 1.** *Let  $P$  be a polyomino with  $|\mathcal{B}(P)| = n$ . It can be decided in  $O(n \log^2 n)$  time if  $P$  has an isohedral tiling (of the plane).*

Here we survey some of the ideas involved in the proof. The proof starts with a list of the boundary word factorization forms that together characterize the polyominoes capable of tiling isohedrally. These are found in the bordered subregion<sup>2</sup> of Table 10 of [19]<sup>3</sup> excluding the two types of isohedral tilings that use  $60^\circ$  and  $120^\circ$  rotations of the shape. The factorizations can be cross-verified using the incidence and adjacency symbols of a more detailed classification of isohedral tiling types of Grünbaum and Shephard [14], and correspond to the isohedral types IH 1, 4, 28, 2, 3, 5, and 6 in this classification. The factorization forms are:

- Translation:  $ABC\hat{A}\hat{B}\hat{C}$ .
- Half-turn:  $W = ABC\hat{A}DE$  with  $B, C, D, E$  palindromes.
- Quarter-turn:  $W = ABC$  with  $A$  a palindrome and  $B, C$  90-dromes.
- Type-1 reflection:  $W = ABf_{\Theta}(B)\hat{A}Cf_{\Phi}(C)$  for some  $\Theta, \Phi$ .
- Type-2 reflection:  $W = ABC\hat{A}f_{\Theta}(C)f_{\Theta}(B)$ .
- Type-1 half-turn-reflection:  $W = ABC\hat{A}Df_{\Theta}(D)$  with  $B, C$  palindromes.
- Type-2 half-turn-reflection:  $W = ABCDf_{\Theta}(B)f_{\Phi}(D)$  with  $A, C$  palindromes and  $\Theta^\circ - \Phi^\circ = \pm 90^\circ$ .

<sup>1</sup> In this work, several types of factors and pairs of factors have restricted “admissible” versions. Intuitively, admissible versions are maximal in a natural sense for each type. For instance, a  $\Theta$ -drome is admissible if it is the longest  $\Theta$ -drome with its center.

<sup>2</sup> A translation of the caption of Table 10: “The strong border contains 9 major types, from which the others can be thought of as emerging by shrinking lines or line pairs.”

<sup>3</sup> Reproduced on page 326 of [33].

The second author [36] gave a  $O(n)$ -time algorithm for deciding if a boundary word of length  $n$  has a translation factorization. Section 4 gives a  $O(n \log^2 n)$ -time algorithm for deciding if a boundary word of length  $n$  has a half-turn factorization. For the remaining boundary word factorization forms,  $O(n \log n)$ -time or faster algorithms also exist. Due to space constraints, these algorithms are omitted. Brlek, Koskas, and Provençal [4] provide a  $O(n)$ -time algorithm for deciding if a given circular word is the boundary word of a polyomino, and we assume for the remainder of the paper that the input is guaranteed to be the boundary word of a polyomino and thus simple. This assumption of simplicity is used to prove that factors and pairs of factors in a factorization are *admissible*: maximal in a natural sense for each factor (pair) type. E.g., for half-turn factorizations:

► **Lemma 17 (restated).** Let  $P$  be a polyomino and  $\mathcal{B}(P) = ABC\hat{A}DE$  with  $B, C, D, E$  palindromes. Then the gapped mirror pair  $A, \hat{A}$  and palindromes  $B, C, D, E$  are admissible.

For various factors and factor pairs, this implies there are  $O(n)$  or  $O(n \log n)$  candidate factors for these elements of a factorization, and they can be computed in similar time. Note that this alone is not sufficient to solve these problems in quasi-linear time. Without additional structural results, attempting to combine a quasi-linear number of factors into a factorization with one of the 6 forms is at least as hard as searching for cycles of fixed length between 3 and 6 in a graph with  $n$  vertices and  $O(n)$  edges, only known to admit a  $O(n^{1.67})$ -time algorithm [1]. Additional structural results must be used, such as the following for half-turn factorizations:

► **Lemma 7 (restated).** The prefix palindrome factorization of a word  $W$  has the form  $W = X_1^{r_1} X_2^{r_2} \dots X_m^{r_m} Q$  with:

- Every  $X_i$  primitive.
- For all  $3 \leq i \leq m$ ,  $\sum_{j=1}^{i-2} |X_j^{r_j}| \leq |X_i|$  and thus  $m = O(\log |W|)$ .

Such results allow more efficient “batch processing” of factors to achieve quasi-linear running time. It can also be seen by cursory examination that each algorithm returns affirmatively only once witness factorization is found. Witness factorizations define the set of boundary intervals shared by pairs of neighboring tiles in an isohedral tiling, thus the algorithm can also return a witness isohedral tiling if desired.

## 4 Half-Turn Factorizations

► **Definition 2.** A half-turn factorization of a boundary word  $W$  has the form  $W = ABC\hat{A}DE$  with  $B, C, D, E$  palindromes.

### 4.1 Prefix palindrome factorizations

► **Definition 3.** Let  $W$  be a word. A factorization  $W = F_1 F_2 \dots F_{n+1}$  is a *prefix palindrome factorization* of  $W$  provided that the set of prefix palindromes of  $W$  is  $\{F_1 F_2 \dots F_i : 1 \leq i \leq n\}$ .

► **Lemma 4.** Let  $W = PX$  with  $P, W$  palindromes and  $0 < |P| < |W|$ . Then  $W$  has a period of length  $|X|$ . Furthermore, if  $X$  is composite, then  $W$  has a prefix palindrome longer than  $P$ .

**Proof.** Since  $W = PX$  and  $P, W$  are palindromes,  $W = \widetilde{W} = \widetilde{P}\widetilde{X} = \widetilde{X}\widetilde{P} = \widetilde{X}P$ . So  $P$  is a prefix of  $\widetilde{X}P$  and so  $\widetilde{X}$  is a period of  $P$  and of  $W = \widetilde{X}P$ . Since  $\widetilde{X}$  is a period of  $W$  and  $|\widetilde{X}| < |W|$ , there exist words  $Y, Z$  such that  $\widetilde{X} = ZY$ ,  $W = (ZY)^p Z$ , and  $P = (ZY)^{p-1} Z$  for some  $p \geq 1$ . So  $X = YZ$  and since  $W$  is a palindrome,  $Y$  and  $Z$  are palindromes.

If  $X$  is composite, then  $X = G^k$  for some  $k \geq 2$ . So there exist words  $G_1, G_2$  such that  $G = G_1G_2, Y = (G_1G_2)^iG_1, Z = G_2(G_1G_2)^{k-i-1}$ . Since  $Y$  and  $Z$  are palindromes,  $G_1$  and  $G_2$  are palindromes.

Now we construct a prefix palindrome of  $W$ , called  $Q$ , that is longer than  $P$ . Without loss of generality, assume  $|G_2| > 0$ . Then there are two possibilities for the values of  $|G_2|$  and  $|Z|$ :

1.  $|G_2| < |Z|$  and we let  $Q = \tilde{X}^pG_2$ .
2.  $|G_2| = |Z|$  and we let  $Q = PG_2$ .

In the first case,  $Q = \tilde{X}^pG_2 = (G_1G_2)^pG_2 = (G_2G_1)^{kp}G_2$ , so  $Q$  is a palindrome. Also,  $W = (ZY)^pZ$  and  $G_2$  is a prefix of  $Z$ , so  $Q$  is a prefix of  $W$  and  $|P| = |W| - |X| = |W| - |ZY| = p|ZY| - |Y| < |Q| = p|ZY| + |G_2| < p|ZY| + |Z| = |W|$ .

In the second case,  $Y = G_2^i$  and  $Z = G_2^{k-i}$ . So  $Q = PG_2 = (ZY)^{p-1}ZG_2 = G_2^{kp-k}G_2^{k-i}G_2 = G_2^{kp-i+1}$  and  $Q$  is a palindrome. Also,  $W = PX$  and  $G_2$  is a prefix of  $X$ , so  $Q$  is a prefix of  $W$  and  $|P| < |Q| = |P| + |G_2| < |P| + |X| = |W|$ . ◀

The following is a well-known result; see Chapter 2 of Crochemore and Rytter [6].

► **Lemma 5** (Fine and Wilf's theorem [8]). *Let  $W$  be a word with periods of length  $p$  and  $q$ . If  $p + q \leq |W|$ , then  $W$  also has a period of length  $\gcd(p, q)$ .*

► **Lemma 6.** *Let  $P_1, P_2, \dots, P_m$  be the set of prefix palindromes of a word with  $0 < |P_1| < |P_2| < \dots < |P_m|$ . Then for any  $1 \leq i \leq m - 2$ , either  $|P_{i+1}| - |P_i| = |P_{i+2}| - |P_{i+1}|$  or  $|P_i| + |P_{i+1}| < |P_{i+2}|$ .*

**Proof.** Let  $P_{i+1} = P_iX_i$  and  $P_{i+2} = P_iX_iX_{i+1}$ .

Since there are no prefix palindromes of length between  $|P_i|$  and  $|P_{i+1}|$  or  $|P_{i+1}|$  and  $|P_{i+2}|$ , Lemma 4 implies  $X_i$  and  $X_{i+1}$  are primitive and  $P_{i+1}$  and  $P_{i+2}$  have periods of length  $|X_i|$  and  $|X_{i+1}|$ , respectively. Since  $P_{i+1}$  is a prefix of  $P_{i+2}$ , it also has a period of length  $|X_{i+1}|$ .

The lemma permits  $|X_i| = |X_{i+1}|$ , so assume  $|X_i| \neq |X_{i+1}|$ . If  $|X_{i+1}| \leq |P_i|$ , then  $P_{i+1}$  has periods of length  $|X_i|$  and  $|X_{i+1}|$  with  $|X_{i+1}| + |X_i| \leq |P_{i+1}|$ . Then by Lemma 5,  $P_{i+1}$  has a period of length  $\gcd(|X_i|, |X_{i+1}|)$ . This length must be at least  $|X_i|$  and  $|X_{i+1}|$ , otherwise  $X_i$  or  $X_{i+1}$  is not primitive. So  $|X_i| = |X_{i+1}|$ . Otherwise,  $|X_{i+1}| > |P_i|$  and so  $|P_i| + |P_{i+1}| < |X_{i+1}| + |P_{i+1}| = |P_{i+2}|$ . ◀

The next lemma is a strengthening of similar prior results by Apostolico, Breslauer, and Galil [2], I et al. [21], and Matsubara et al. [27].

► **Lemma 7** (Prefix Palindrome Factorization Lemma). *The prefix palindrome factorization of a word  $W$  has the form  $W = X_1^{r_1}X_2^{r_2} \dots X_m^{r_m}Q$  with:*

- Every  $X_i$  primitive.
- For all  $3 \leq i \leq m$ ,  $\sum_{j=1}^{i-2} |X_j^{r_j}| \leq |X_i|$  and thus  $m = O(\log |W|)$ .

**Proof.** We give a constructive proof. Let  $P_1, P_2, \dots, P_n$  be the set of prefix palindromes of  $W$  with  $|P_1| < |P_2| < \dots < |P_n|$ .

Let  $W_i$  be the word such that  $P_{i+1} = P_iW_i$  and let  $Q$  be the word such that  $W = P_nQ$ . So  $W$  has a prefix palindrome factorization  $W_1W_2 \dots W_nQ$ . By Lemma 4, every  $W_i$  is primitive. Moreover, by Lemma 6 either  $|W_i| = |W_{i+1}|$  or  $|P_i| < |W_{i+1}|$  for every  $1 \leq i \leq n - 2$ .

Suppose  $|W_i| = |W_{i+1}|$ . By Lemma 6,  $P_i$  and  $P_{i+1}$  have a common period and thus  $W_i = W_{i+1}$ . More generally, if  $|W_i| = |W_{i+1}| = \dots = |W_{i+c}|$ , then  $W_i = W_{i+1} = \dots = W_{i+c}$ . If  $|P_i| < |W_{i+1}|$ , then  $|W_{i+1}| > |P_i| = \sum_{j=1}^{i-1} |W_j|$ . So the factorization  $W_1W_2 \dots W_nQ$  can

be rewritten as  $X_1^{r_1} X_2^{r_2} \dots X_m^{r_m} Q$  with the property that  $|X_i| \geq \sum_{j=1}^{i-2} |X_j|^{r_j}$ . So for all  $i \geq 4$ ,  $2|X_{i-3}| < |X_i|$  and thus  $m = O(\log |W|)$ . ◀

Such a factorization can be stored using  $O(\log |W|)$  space by simply storing  $|X_i|$  and  $r_i$  for each  $i$ . Additional observations can be used prove that  $|W|$  prefix palindrome factorizations of the suffixes of a word  $W$  can be computed in optimal time:

► **Lemma 8.** *The prefix palindrome factorizations of all shifts of a circular word  $W$  can be computed in  $O(|W| \log |W|)$  total time.*

**Proof.** Lemma 9 of [21] states that the prefix palindrome factorization of a non-circular word  $xY$  can be computed in  $O(\log |Y|)$  time given the factorization of  $Y$ . Thus the factorizations of non-circular word  $WW$  can be enumerated in  $O(|W| \log |W|)$  time, beginning with  $Y = W[-1]$ . Every shift of word  $W$  is a subword of the non-circular word  $WW$ , and the computed factorizations can be trimmed in  $O(\log |W|)$ -time per factorization to be the factorizations of shifts of  $W$ . ◀

Identical results, including a suffix palindrome factorization lemma, clearly hold for suffix palindromes as well.

## 4.2 Algorithm

The main idea is to iterate over all pairs of adjacent letters and guess the form of the palindromes  $D$  and  $E$  in both directions from that location. Specifically, guess what repeated factor  $X_i^{r_i}$  terminates them in their prefix and suffix palindrome factorizations. Then try to complete the factorization using Lemma 16, which decides if it is possible to rewrite a given portion of the boundary as  $L^b ABC \hat{A} R^c$  with  $B, C$  palindromes and  $b, c$  in some range. The results leading up to Lemma 16 provide the necessary structure to achieve this goal. In particular, Lemma 14 shows how to decompose a word into two palindromes, and Lemmas 9 through 11 yield fast detection of a factorization of the form  $BCR^k$  with  $B, C$  palindromes.

► **Lemma 9.** *Let  $W$  be a word with subwords  $L, R$  such that  $W = LR^r$  and  $R \not\prec_{\text{suff}} L$ . Let  $P_1, P_2$  be palindromes such that  $W = P_1 P_2 R^k$  with  $|L| \leq |P_1|$ . Then there exists a palindrome  $P'_2$  and integer  $k'$  such that  $W = P_1 P'_2 R^{k'}$  with  $|P'_2| < |R|$ .*

**Proof.** Since  $|L| \leq |P_1|$ ,  $P_2$  is a suffix of  $R^i$  for some minimal  $i$ . If  $i \leq 1$ , then either  $|P_2| = |R|$  (and  $|P'_2| = 0, k' = k + 1$ ) or  $|P_2| < |R|$  and the claim is satisfied. If  $i \geq 2$ , then there exist words  $Y, Z$  with  $|Y| > 0$  such that  $R = YZ$  and  $P_2 = Z(YZ)^{i-1}$ .

Let  $P'_2 = Z$  and  $k' = i - 1 + k$ . So  $W = P_1 P_2 R^k = P_1 P'_2 (Y P'_2)^{i-1} R^k = P_1 P'_2 R^{k'}$ . Since  $|YZ| = |Y P'_2| = |R|$  and  $|Y| > 0$ , it follows that  $|P'_2| < |R|$ . Since  $P_2 = Z(YZ)^{i-2} YZ$ ,  $Z = P'_2$  is a palindrome. ◀

► **Lemma 10** (Lemma C4 of [9]). *If a word  $X_1 X_2 = Y_1 Y_2 = Z_1 Z_2$  with  $X_2, Y_1, Y_2, Z_1$  palindromes. Then  $X_1$  and  $Z_2$  are palindromes.*

► **Lemma 11.** *Let  $R$  be a primitive word and let  $W = LR^r$ . There is a set of integers  $H$  with  $|H| = O(\log |W|)$  such that  $W = P_1 P_2 R^k$  if and only if it does so with  $|LR^h| \leq |P_1| \leq |LR^{h+1}|$  for some  $h \in H$ . Moreover, given  $|R|$  and the prefix palindrome factorization of  $W$ ,  $H$  can be computed in  $O(\log |W|)$  time.*

**Proof.** We may assume  $|P_2| < |R|$  by Lemma 9. Consider the prefix palindrome factorization of  $W$  as described in Lemma 7. Any solution  $P_1$  ends with one of the repeating subwords  $X_i$  of the factorization. There are three cases:  $|X_i| < |R|$ ,  $|X_i| > |R|$ , and  $|X_i| = |R|$ .

**Case 1:**  $|X_i| < |R|$ . We claim that if  $|X_i| < |R|$ , then  $X_i^{r_i}$  overlaps  $R^r$  in at most two repetitions of  $R$  (and there are at most three values of  $h$ ). Assume, for the sake of contradiction, that  $R^2$  is a subword of  $|X_i^{r_i}|$ . Then  $R^2$  is a word of length at least  $|X_i| + |R|$  with periods of length  $|X_i|$  and  $|R|$ . So by Lemma 5,  $R$  has a period of length  $\gcd(|X_i|, |R|) \leq |X_i| < |R|$ , a contradiction.

**Case 2:**  $|X_i| > |R|$ . We claim that if  $|X_i| > |R|$ , then  $X_i$  cannot repeat in  $R^r$  (and there are at most two values of  $h$ ). Assume, for the sake of contradiction, that  $X_i^2$  is a subword of  $R^r$ . Then by Lemma 5,  $X_i$  has a period of length  $\gcd(|X_i|, |R|) \leq |R| < |X_i|$ . So by Lemma 4, the factorization given was not a prefix palindrome factorization, a contradiction.

**Case 3:**  $|X_i| = |R|$ . We claim that if  $|X_i| = |R|$ , then  $h = 0$  suffices. Suppose  $|LR^h| \leq |P_1| \leq |LR^{h+1}|$  for some  $h \geq 1$ . By Lemma 4,  $P_1$  has a period of length  $|X_i| = |R|$ . Let  $Y, Z$  be words such that  $YZ$  is a period of  $P_1$  and  $|R| - |Z| = |P_2|$ . So  $P_1 = (YZ)^p Y$  for some  $p \geq 1$  and  $Y, Z$  are palindromes.

Since  $LR^{h+1} = (YZ)^p Y P_2$  and  $|YZYP_2| = 2|R|$ ,  $YZ = Y P_2 = R$ . So  $LR = (YZ)^{p-h} Y P_2 = P_1' P_2$ , where  $P_1' = (YZ)^{p-h} Y$  and thus is a palindrome. So there exists a  $P_1'$  with  $|LR^0| \leq |P_1'| \leq |LR^1|$ .

**Computing  $H$ .** The value of  $h$  in case 3 is always 0. For case 1, use the values of  $h$  such that  $X_i^{r_i}$  contains the last letter of  $LR^h$  or  $LR^{h+1}$ . For case 2, use the values of  $h$  such that the prefix palindrome ending at the unique repetition of  $X_i$  has length between  $|LR^h|$  and  $|LR^{h+1}|$ . ◀

► **Lemma 12.** *Let  $R$  be a primitive word and let  $W = LR^r$ . Assume that  $|R|$  and the prefix and suffix palindrome factorizations of  $W$  are given. Then it can be decided in  $O(\log |W|)$  time if  $W = P_1 P_2 R^k$  with  $P_1, P_2$  palindromes and  $|L| \leq |P_1|$ .*

**Proof.** We may assume  $|P_2| < |R|$  by Lemma 9. First, use Lemma 11 to compute a  $O(\log |W|)$ -sized candidate set of integers  $H$  such that a solution exists if and only if  $LR^{h+1} = P_1 P_2$  with  $|LR^h| \leq |P_1| \leq |LR^{h+1}|$  for some  $h \in H$ . By Lemma 10, it suffices to check for such solutions with at least one of the following types of palindromes:

- The longest prefix palindrome of  $LR^{h+1}$  with length at least  $|LR^h|$ .
- The longest suffix palindrome of  $LR^{h+1}$  with length less than  $|R|$ .

Compute the longest prefix palindromes of  $LR^{h+1}$  for all values of  $h$  in  $O(\log |W|)$  total time using a two-finger scan of (1) the prefix palindrome factorization of  $W$  and (2) the values of  $h$ . Use a second two-finger scan of (1) these prefix palindromes and (2) the suffix palindrome factorization of the last  $|R|$  letters of the suffix palindrome factorization of  $LR^r$  to search for a solution  $P_1, P_2$ .

The longest suffix palindrome of  $LR^{h+1}$  (with length less than  $R$ ) is invariant for  $h$  and can be computed in  $O(\log |W|)$  time, using the last  $|R|$  letters of the suffix palindrome factorization of  $LR^r$ . Call the length of this palindrome  $\lambda$ . Use a scan of the prefix palindrome factorization to determine if a prefix palindrome of  $W$  has length  $|LR^h| - \lambda$  for some value of  $h$ . ◀

Next, we develop a second result that is combined with the previous lemma to obtain Lemma 16.

► **Lemma 13.** *Let  $L$  and  $R$  be words such that  $L \not\prec_{\text{pre}} R$ . Let  $A_i$  be the longest common prefix of  $L^{l-i} R$  and a word  $U$ . Let  $k = l - \lceil |A_0|/|L| \rceil$ . Then:*

- For all  $i$  with  $0 \leq i \leq k$ ,  $A_i = A_0$  and  $|A_i| < |L^{l-k}|$ .
- For all  $i$  with  $k+2 \leq i \leq l$ ,  $|A_i| - |L^{l-i}| = |A_{k+2}| - |L^{l-(k+2)}|$ .

**Proof.** Let  $k$  be the maximum  $k$  such that  $|A_k| < |L^{l-k}|$ . We show that this value of  $k$  has the desired properties, including that  $k = l - \lceil |A_0|/|L| \rceil$ .

**Property 1.** Let  $0 \leq i \leq k$ . Since  $A_i$  is a prefix of  $L^{l-i}R$  and  $|A_i| < |L^{l-i}|$ ,  $A_i$  is the longest common prefix of  $L^l$  and  $U$ . This is true for all choices of  $A_i$ , and thus all  $A_i$  are equal.

**Property 2.** By definition,  $|A_{k+1}| \geq |L^{l-(k+1)}|$  and so  $L^{l-(k+1)}$  is a prefix of  $U$ . Since  $L \not\prec_{\text{pre}} R$ , the length of  $A_{k+2}$ , the longest common prefix of  $L^{l-(k+2)}R$  and  $U$ , must be less than  $|L^{l-(k+1)}|$ . So  $|A_{k+2}| < |L^{l-(k+1)}|$  and  $L$  is a period of  $A_{k+2}$ .

Let  $R_1, R_2$  be words such that  $R = R_1R_2$  and  $A_{k+2} = L^{l-(k+2)}R_1$ . Then  $|R_1| < |L|$  and so  $R_1$  is the longest common prefix of  $R$  and  $L$ .

So for all  $i \geq k+2$ , the longest common prefix of  $L^{l-i}R$  and  $L^{l-(k+1)}$  is  $L^{l-i}R_1$ . Moreover, since  $|L^{l-i}R_1| < |L^{l-(k+1)}|$  and  $L^{l-(k+1)}$  is a prefix of  $U$ , the longest common prefix of  $L^{l-i}R$  and  $U$  is also  $L^{l-i}R_1$ .

**Property 3.** Finally, we prove that  $k = l - \lceil |A_0|/|L| \rceil$ . Since  $|A_0| = |A_i| < |L^{l-i}|$  for all  $0 \leq i \leq k$ , it follows that  $|A_0| < |L^{l-k}| = |L|(l-k)$ . Then by algebra,  $k < l - |A_0|/|L|$ .

Since  $|A_{k+1}| \geq |L^{l-(k+1)}|$ , it must be that  $U$  has a prefix  $L^{l-(k+1)}$ . So  $|A_0| = |A_k| \geq |L^{l-(k-1)}| = |L|(l-k-1)$ . Then by algebra  $k+1 \geq l - |A_0|/|L|$ . So  $k < l - |A_0|/|L| \leq k+1$ . ◀

► **Lemma 14.** Let  $W$  be a word and  $l_1, l_2$  integers. Assume the prefix and suffix palindrome factorizations of  $W$  are given. Then it can be decided in  $O(\log |W|)$  time if there exist palindromes  $P_1, P_2$  such that  $W = P_1P_2$  with  $|P_1| \geq l_1, |P_2| \geq l_2$ .

**Proof.** By Lemma 10, such a pair of palindromes exist if and only if there exists such a pair such that either  $P_1$  is the longest prefix palindrome of  $W$  with  $|P_1| \leq |W| - l_2$  or  $P_2$  is the longest suffix palindrome of  $W$  with  $|P_2| \leq |W| - l_1$ . Scan each factorization in  $O(\log |W|)$  time to find these specific palindromes, and then scan the opposite factorizations for a second palindrome to complete  $W$ . ◀

The following result comes from a trivial modification of Theorem 9.1.1 of [17] to allow for circular words, namely giving the concatenation of two copies of a corresponding non-circular word as input, and returning  $\infty$  if the output has length more than  $\text{lcm}(|X|, |Y|)$ .

► **Lemma 15** (Theorem 9.1.1 of [17]). Two circular words  $X, Y$  can be preprocessed in  $O(|X| + |Y|)$  time to support the following queries in  $O(1)$ -time: what is the longest common factor of  $X$  and  $Y$  starting at  $X[i]$  and  $Y[j]$ ?

► **Lemma 16.** Let  $W$  be a circular word. Let  $W = L^lZ$  and  $W = YR^r$  such that  $L \not\prec_{\text{pre}} Z$ ,  $R \not\prec_{\text{suff}} Y$ , and  $L, R$  are primitive. Assume that the prefix and suffix palindrome factorizations of every shift of  $W$  are given. It can be decided in  $O((l+r) \log |W|)$  time if there exist positive integers  $b, c$  such that  $W = L^bAP_1P_2\widehat{A}R^c$  with  $A, \widehat{A}$  admissible and  $P_1, P_2$  palindromes.

**Proof.** The approach is to iteratively search for a solution for each value of  $b$ , carrying out the same algorithm on each value. Then performing an identical, symmetric iteration through the values of  $c$ . First assume  $b$  is a fixed value and  $c$  is not. Let  $A_i$  be the longest word such that  $L^bA_i$  and  $\widehat{A}_iR^i$  are a prefix and suffix of  $W$ , respectively. Lemma 13 implies that there exists an integer  $k$  such that:

- For all  $i$  with  $0 \leq i \leq k$ ,  $|L^b A_i|$  is fixed and  $|\widehat{A_i R^i}| = |A_0| + |R|i$ .
- For all  $i$  with  $k + 2 \leq i \leq l$ ,  $|\widehat{A_i R^i}|$  is fixed and  $|L^b A_i| = |L^b| + |A_{k+2}| - |R|(i - (k + 2))$ .
- $k$  can be computed in  $O(1)$  time assuming a data structure allowing  $O(1)$  time longest common prefix queries for suffixes of  $W$  and  $\widehat{W}$  is given.

In other words,  $k$  is an efficiently-computable integer that partitions the values of  $i$  into three parts: one with a single value ( $i = k + 1$ ) and two others where either  $|L^b A_i|$  or  $|\widehat{A_i R^i}|$  is fixed and the other is a linear set. Handle the case of  $i = k + 1$  individually by using Lemma 14 to check if the word between  $L^b A_{k+1}$  and  $\widehat{A_{k+1} R^{k+1}}$  has a factorization into two palindromes. Next, check that all  $A_i, \widehat{A_i}$  except  $i = 0$  are admissible by verifying  $L^b[-1] \neq \widehat{R^i}[1]$ . Also handle  $i = 0$  individually, including checking admissibility. Lemma 12 is used to handle the remaining two cases in  $O(\log |W|)$  time each.

**Case 1:  $0 \leq i \leq k$ .** In this case,  $|L^b A_i|$  is fixed and  $|\widehat{A_i R^i}| = |A_0| + |R|i$ . If  $k - 1 < 3$ , then handle all three cases individually in  $O(\log |W|)$  time using Lemma 14. Otherwise handle only  $i = k$  similarly.

By Lemma 13,  $A_i = A_0$  for all  $0 \leq i \leq k - 1$  and  $|A_0| < |R^{r-k}|$ . So  $\widehat{A_0 R^i}$  for all  $0 \leq i \leq k - 1$  and  $R^r$  are suffixes of  $W$ . Also, for all  $i \leq k - 1$ ,  $|\widehat{A_0 R^i}| \leq |\widehat{A_0 R^{k-1}}| \leq |\widehat{A_0 R^k}| - |R| \leq |R^r| - |R|$ . So for some  $R'$  with  $|R'| = |R|$ ,  $\{\widehat{A_0 R^i} : 0 \leq i \leq k - 1\} = \{(R')^i \widehat{A_0} : 0 \leq i \leq k - 1\}$ . Let  $L'$  be such that  $W = L^b A_i L'(R')^k \widehat{A_0}$ . Then a solution factorization exists if and only if there exist palindromes  $P_1 P_2 = L'(R')^{k-i}$  for some  $0 \leq i \leq k - 1$ .

First, search for solutions with  $|P_1| \geq |L'|$ . We first prove that  $R'$  is primitive, allowing Lemma 12 to be invoked. Suppose, for the sake of contradiction, that  $R'$  has a period of length  $p$  with  $p < |R'|$ . So  $(R')^2$  has periods of length  $|R'|$  and  $p$  such that  $|R'| + p < |(R')^2|$  and so by Lemma 5 has a period of length  $\gcd(|R'|, p) < |R'|$ . Then since  $(R')^2$  contains  $R$  as a subword,  $R$  also has a period of length  $p$  and thus is not primitive, a contradiction.

For solutions with  $|P_1| < |L'|$ , Lemma 10 implies that it suffices to check for solutions with the longest possible  $P_1$  (longest possible  $P_2$  is handled when performing the symmetric iteration over values of  $c$ ). To do so, scan the prefix palindrome factorization starting at  $L'[|P_1| + 1]$  for a palindrome of length  $|L'| - |P_1| + |(R')^{k-i}|$  with  $0 \leq i \leq k - 1$ .

**Case 2:  $k + 2 \leq i \leq l$ .** In this case,  $|\widehat{A_i R^i}|$  is fixed and  $|L^b A_i| = |L^b| + |A_{k+2}| - |R|(i - (k + 2))$ . Then there exists a word  $R'$  such that  $W = L^b(\widehat{R})^{r-i} A_r R' \widehat{A_i R^i}$  for all  $k + 2 \leq i \leq l$ . Let  $L'$  be the suffix of  $\widehat{R} A_r$  of length  $|R|$ . Then  $W = L^b A_r (L')^{r-(k+2)} R' \widehat{A_i R^i}$ . So there exists a pair of palindromes  $P_1, P_2$  with  $W = L^b A_i P_1 P_2 \widehat{A_i R^i}$  for some  $k + 2 \leq i \leq l$  if and only if  $(L')^{r-i} R' = P_1 P_2$  for some  $k + 2 \leq i \leq l$ . This situation is identical to that encountered in the previous case – handle in the same way.

**Handling overlap.** The description of the algorithm so far has ignored the possibility that  $|L^b A_i| + |\widehat{A_i R^i}| > |W|$ , i.e., that  $L^b A_i$  and  $\widehat{A_i R^i}$  “overlap”. For the case of  $0 \leq i \leq k$ , this occurs when  $|L^b A_0| + |\widehat{A_0 R^i}| > |W|$ . Restricting the values of  $i$  to satisfy  $0 \leq i \leq \min(k, \lfloor (|W| - b|L| - 2|A_0|)/|R| \rfloor)$  ensures that  $W$  can be decomposed as claimed. For the case of  $k + 2 \leq i \leq l$ , this occurs when  $|L^b R^{r-i} A_r| + |\widehat{A_i R^i}| > |W|$ . Restricting the values of  $i$  to satisfy  $\max(\lceil (2|A_r| + 2r|R| + b|L| - |W|)/|R| \rceil, k + 2) \leq i \leq l$  ensures that  $W$  can be decomposed as claimed. Check the individually handled cases, namely  $i = k, k + 1$ , for overlap individually.

**Running time.** The running time of this algorithm is  $O((l + r) \log |W|)$ , since the amount of time spent for each value of  $b$  and  $c$  is  $O(\log |W|)$  to handle individual values of  $i$  and

$O(\log |W|)$  to handle each large case by Lemma 12. However, this assumes a data structure enabling  $O(1)$  time common prefix queries on  $W$  and  $\widehat{W}$ . Compute such a data structure in  $O(|W|)$  time using Lemma 15. Since  $\Omega(|W|)$  time must be spent to decide if a boundary word has a half-turn factorization, such a computation has no additional asymptotic cost. ◀

► **Lemma 17.** *Let  $P$  be a polyomino and  $\mathcal{B}(P) = ABC\widehat{A}DE$  with  $B, C, D, E$  palindromes. Then the gapped mirror pair  $A, \widehat{A}$  and palindromes  $B, C, D, E$  are admissible.*

**Proof.**

**$A, \widehat{A}$  is admissible.** It cannot be that  $|B| = |C| = 0$ , since then  $ABC\widehat{A} = A\widehat{A}$  is non-simple. If  $BC[1] = \overline{BC}[-1]$ , then  $|B|, |C| > 0$  and thus  $B[-1] = BC[1] = \overline{BC}[-1] = \overline{C}[1]$  and  $BC$  is non-simple. So  $BC[1] \neq \overline{BC}[-1]$  and by symmetry,  $DE[1] \neq \overline{DE}[-1]$ . So  $A, \widehat{A}$  are admissible.

**$B, C, D, E$  are admissible.** Consider the pairs of non-equal consecutive letters in  $W$ . These pairs come from sets  $\mathcal{R} = \{\mathbf{lu}, \mathbf{ur}, \mathbf{rd}, \mathbf{dl}\}$  and  $\mathcal{L} = \{\mathbf{ul}, \mathbf{ld}, \mathbf{dr}, \mathbf{ru}\}$ , and Proposition 6 of [7] states that the number of pairs from  $\mathcal{R}$  is four more than the number from  $\mathcal{L}$ . Also, any palindrome contains an equal number of consecutive letter pairs from  $\mathcal{L}$  and  $\mathcal{R}$ .

If  $|A| = 0$ , then  $W$  has factorization  $W = BCDE$  with the four consecutive-letter pairs from  $\mathcal{R}$  not contained in any factor, i.e., for each factor  $X \in \{B, C, D, E\}$ ,  $W = XY$  with  $Y[-1]X[1], X[-1]Y[1] \in \mathcal{R}$ . Since  $X$  is a palindrome,  $X[-1]Y[1] = X[1]Y[1] \in \mathcal{R}$  and so  $Y[1] \neq Y[-1]$ . Thus  $X$  is admissible.

If  $|A| > 0$ , then  $|BC| > 0$ , since otherwise  $A[-1]\widehat{A}[1] = A[-1]\overline{A}[-1]$  is a subword and  $W$  is non-simple. Without loss of generality,  $|B| > 0$ . If  $|C| = 0$ , then  $W = BY$  with  $Y[1] = \widehat{A}[1] = \overline{A}[-1] \neq A[-1] = Y[-1]$  and  $B$  is admissible. If  $|C| > 0$ , then  $C[1] = C[-1] \neq \widehat{A}[1] = A[-1]$ . So  $W = BY$  with  $Y[-1] = A[-1] \neq C[1] = Y[1]$  and so  $B$  is admissible. By symmetry, it is also the case that  $C, D$  and  $E$  are also admissible. ◀

► **Theorem 18.** *Let  $P$  be a polyomino with  $|\mathcal{B}(P)| = n$ . It can be decided in  $O(n \log^2 n)$  time if  $\mathcal{B}(P)$  has a half-turn factorization.*

**Proof.** First, compute the prefix palindrome factorizations of each shift of  $W$  by computing and truncating the prefix palindrome factorizations of the  $|W|$  longest suffixes of  $W$  using Lemma 8. Similarly compute the suffix palindrome factorization of every shift of  $W$ . By Lemma 8, this takes  $O(|W| \log |W|)$  total time.

Next, compute the admissible factors (including zero-length factors), i.e., palindromes maximal about their center, by computing and truncating the maximal palindromes of  $WW$  output by Manacher's  $O(|W|)$ -time algorithm [25]. Each admissible factor  $F$  is contained in a prefix palindrome factorization as  $X_1^{r_1} X_2^{r_2} \dots X_i^j$  with either  $1 \leq i \leq m$  and  $0 \leq j < r_i$ , or  $i = m$  and  $j = r_m$  if  $F$  is the longest prefix palindrome of the word. If  $j < r_i$ , call  $X_i^{r_i}$  the *terminator* of  $F$ , otherwise call  $Q^1$  the terminator of  $F$ . Either all or none of the prefix palindromes with a given terminator are admissible. Similar definitions and observations apply to suffix palindrome factorizations.

For each admissible factor  $W$ , mark the two terminators (one prefix, one suffix) of the factor. Locating the prefix and suffix terminators for each of the  $2|W|$  admissible factors takes  $O(|W| \log |W|)$  total time.

Without loss of generality, every solution half-turn factorization has  $|E| > 0$ . Search for half-turn factorizations  $ABC\widehat{A}DE$  by iterating over possible first letters of  $E$ . By Lemma 17, only solutions with admissible  $D$  and  $E$  must be considered. This corresponds

to a palindromes  $D$  and  $E$  starting and ending with a marked terminators  $X_s^{r_s}$  and  $X_p^{r_p}$ , respectively. For each such terminator pair  $X_s^{r_s}, X_p^{r_p}$ , use Lemma 16 with  $L = X_p, l = r_p, R = \widetilde{X}_s, r = r_s$  to check for a partial factorization  $ABC\widehat{A}$  to complete the factorization along with  $D$  and  $E$  with the marked terminator pair. By Lemma 7,  $X_s$  and  $X_p$  are primitive and by definition of palindrome factorizations,  $l$  and  $r$  are maximal.

Checking for a partial factorization using Lemma 16, each pair  $X_s^{r_s}, X_p^{r_p}$  takes  $O((r_s + r_p) \log |W|)$  time,  $O(\log |W|)$  time for each admissible factor involved. Moreover, each admissible factor is involved in  $O(\log |W|)$  pairs of terminators:  $O(\log |W|)$  prefix (suffix) terminators when  $E$  ( $D$ ). So  $O(\log^2 |W|)$  total time is spent per admissible factor and in total the the algorithm takes  $O(|W| \log^2 |W|)$  time. ◀

## 5 Conclusion

This work demonstrates that not just polynomial, but quasilinear-time algorithms exist for deciding tiling properties of a polyomino. It remains to be seen if a linear-time algorithm exists, or whether a super-linear lower bound for one of the factorization forms exists. The slowest algorithm is for half-turn factorizations, so it seems natural to attack this special case first.

► **Open Problem 1.** *Can it be decided in  $o(n \log^2 n)$ -time if a polyomino  $P$  with  $|\mathcal{B}(P)| = n$  has a half-turn factorization?*

► **Open Problem 2.** *Can it be decided in  $O(n)$ -time if a polyomino  $P$  with  $|\mathcal{B}(P)| = n$  has an isohedral tiling of the plane?*

For monohedral tilings containing only translations of the prototile, a polyomino has such a tiling only if it has one that is also isohedral [3, 35]. Does this remain true for tilings using other sets of transformations of the prototile? Modifying the anisohedral tile of Heesch [18] (see [16]) proves that the answer is “no” for tilings with reflected tiles, while an example of Rhoads [31] proves that the answer is “no” for tilings with  $90^\circ$  rotations of tiles. This leaves one possibility open:

► **Open Problem 3.** *Does there exist a polyomino  $P$  that has a tiling containing only translations and  $180^\circ$  rotations of  $P$  and every such tiling is anisohedral?*

As mentioned in Section 3, there are isohedral tiling types (characterized by boundary factorizations) that cannot be realized by polyominoes due to angle restrictions. Moreover, the boundary factorization forms here also apply to general polygons, under appropriate definitions of “boundary word”. Extending the algorithms presented here to polygons, along with developing algorithms for the remaining boundary factorizations is a natural goal. However, significant challenge remains in efficiently converting a polygon’s boundary into a word that can be treated with the approach used here.

► **Open Problem 4.** *Can it be decided in  $O(n \log^2 n)$  time if a polygon with  $n$  vertices has an isohedral tiling of the plane?*

Observe that pairs of tiles in a tiling that can be mapped to each other via a symmetry of the tiling induces a partition of the tiles. Define a tiling to be  $k$ -isohedral if the partition has  $k$  parts, e.g., an isohedral tiling is 1-isohedral. Thus  $k$ -isohedral tilings are a natural generalization of isohedral tilings that allow increasing complexity; specifically, they cannot be characterized by a single boundary factorization. A natural generalization of the problem considered here is as follows:

► **Open Problem 5.** *Can it be decided efficiently if a polyomino has a  $k$ -isohedral tiling?*

An approach described by Joseph Myers [28] achieves a running time of approximately  $n^{O(k^2)}$ , though a precise analysis of the running time has not been performed. A fixed-parameter tractable algorithm also may be possible. On the other hand, a proof of NP-hardness is unlikely, since it implies, for each  $c \in \mathbb{N}$ , the existence of prototiles whose only tilings are  $k$ -isohedral for  $k \geq c$ . Such tiles are only known to exist for  $c \leq 10$  [28].

**Acknowledgements.** The authors wish to thank anonymous reviewers for comments that improved the correctness of the paper.

---

## References

- 1 N. Alon, R. Yuster, and U. Zwick. Finding and counting given length cycles. *Algorithmica*, 17(3):209–223, 1997.
- 2 A. Apostolico, D. Breslauer, and Z. Galil. Parallel detection of all palindromes in a string. *Theoretical Computer Science*, 141:163–173, 1995.
- 3 D. Beauquier and M. Nivat. On translating one polyomino to tile the plane. *Discrete & Computational Geometry*, 6:575–592, 1991.
- 4 S. Brlek, M. Koskas, and X. Provençal. A linear time and space algorithm for detecting path intersection. In *DGCI 2009*, volume 5810 of *LNCS*, pages 397–408. Springer Berlin Heidelberg, 2009.
- 5 S. Brlek, J.-M. X. Provençal, and Fédou. On the tiling by translation problem. *Discrete Applied Mathematics*, 157:464–475, 2009.
- 6 M. Crochemore and W. Rytter. *Text Algorithms*. Oxford University Press, 1994.
- 7 A. Daurat and M. Nivat. Salient and reentrant points of discrete sets. *Discrete Applied Mathematics*, 151:106–121, 2005.
- 8 N. J. Fine and H. S. Wilf. Uniqueness theorems for periodic functions. *Proceedings of the American Mathematical Society*, 16:109–114, 1965.
- 9 Z. Galil and J. Seiferas. A linear-time on-line recognition algorithm for “Palstar”. *Journal of the ACM*, 25(1):102–111, 1978.
- 10 L. Gambini and L. Vuillon. An algorithm for deciding if a polyomino tiles the plane by translations. *RAIRO – Theoretical Informatics and Applications*, 41(2):147–155, 2007.
- 11 M. Gardner. More about tiling the plane: the possibilities of polyominoes, polyiamonds, and polyhexes. *Scientific American*, pages 112–115, August 1975.
- 12 C. Goodman-Strauss. Open questions in tiling. Online, published 2000. URL: <http://comp.uark.edu/~strauss/papers/survey.pdf>.
- 13 C. Goodman-Strauss. Can’t decide? undecide! *Notices of the American Mathematical Society*, 57(3):343–356, 2010.
- 14 B. Grünbaum and G. C. Shephard. The eighty-one types of isohedral tilings in the plane. *Mathematical Proceedings of the Cambridge Philosophical Society*, 82(2):177–196, 1977.
- 15 B. Grünbaum and G. C. Shephard. Isohedral tilings of the plane by polygons. *Commentarii Mathematici Helvetici*, 53(1):542–571, 1978.
- 16 B. Grünbaum and G. C. Shephard. Tilings with congruent tiles. *Bulletin of the American Mathematical Society*, 3:951–973, 1980.
- 17 D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997.
- 18 H. Heesch. Aufbau der ebene aus kongruenten bereichen. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen*, 1:115–117, 1935.
- 19 H. Heesch and O. Kienzle. *Flächenschluss: System der Formen lückenlos aneinander-schliessender Flachteile*. Springer, 1963.

- 20 D. Hilbert. Mathematical problems. *Bulletin of the American Mathematical Society*, 8(10):437–479, 1902.
- 21 T. I. S. Sugimoto, S. Inenaga, H. Bannai, and M. Takeda. Computing palindromic factorizations and palindromic covers on-line. In A. S. Kulikov, S. O. Kuznetsov, and P. Pevzner, editors, *CPM 2014*, volume 8486 of *LNCS*, pages 150–161. Springer, Switzerland, 2014.
- 22 K. Keating and A. Vince. Isohedral polyomino tiling of the plane. *Discrete and Computational Geometry*, 21(4):615–630, 1999.
- 23 R. B. Kershner. On paving the plane. *The American Mathematical Monthly*, 75(8):839–844, 1968.
- 24 D. E. Knuth, J. H. Morris, and V. R. Pratt. Fast pattern matching in strings. *SIAM Journal on Computing*, 6(2):323–350, 1977.
- 25 G. K. Manacher. A new linear-time “on-line” algorithms for finding the smallest initial palindrome of a string. *Journal of the ACM*, 22(3):346–351, 1975.
- 26 C. Mann, J. McCloud-Mann, and D. Von Derau. Convex pentagons that admit  $i$ -block transitive tilings. Technical report, arXiv, 2015. URL: <http://arxiv.org/abs/1510.01186>.
- 27 W. Matsubara, S. Inenaga, A. Ishino, A. Shinohara, T. Nakamura, and K. Hashimoto. Efficient algorithms to compute compressed longest common substrings and compressed palindromes. *Theoretical Computer Science*, 410:900–913, 2009.
- 28 J. Myers. Polyomino, polyhex, and polyiamond tiling. Online, updated February 2012. URL: <http://www.polyomino.org.uk/mathematics/polyform-tiling/>.
- 29 X. Provençal. *Combinatoire des mots, géométrie discrète et pavages*. PhD thesis, Université du Québec à Montréal, 2008.
- 30 K. Reinhardt. Zur zerlegung der euklidischen räume in kongruente polytope. *Sitzungsberichte der Preussischen Akademie der Wissenschaften*, pages 150–155, 1928.
- 31 G. C. Rhoads. Planar tilings polyominoes, polyhexes, and polyiamonds. *Journal of Computational and Applied Mathematics*, 174:329–353, 2005.
- 32 D. Schattschneider. Will it tile? try the Conway criterion! *Mathematics Monthly*, 53(4):224–233, 1980.
- 33 D. Schattschneider. *Visions of Symmetry: Notebooks, Periodic Drawings, and Related Work of M. C. Escher*. W. H. Freeman and Company, 1990.
- 34 J. E. S. Socolar and J. M. Taylor. An aperiodic hexagonal tile. *Journal of Combinatorial Theory, Series A*, 118(8):2207–2231.
- 35 H. A. G. Wijshoff and J. van Leeuwen. Arbitrary versus periodic storage schemes and tessellations of the plane using one type of polyomino. *Information and Control*, 62:1–25, 1984.
- 36 A. Winslow. An optimal algorithm for tiling the plane with a translated polyomino. In *26th International Symposium on Algorithms and Computation (ISAAC)*, 2015.