

## AN ALGORITHM FOR DECIDING IF A POLYOMINO TILES THE PLANE

IAN GAMBINI<sup>1</sup> AND LAURENT VUILLON<sup>2</sup>

**Abstract.** For polyominoes coded by their boundary word, we describe a quadratic  $O(n^2)$  algorithm in the boundary length  $n$  which improves the naive  $O(n^4)$  algorithm. Techniques used emanate from algorithmics, discrete geometry and combinatorics on words.

**Mathematics Subject Classification.** 68R15, 52C20.

### 1. INTRODUCTION

During the DMCCG conference held at the Institut Henri Poincaré (IHP) in July 2001, we discussed with Alberto del Lungo some problems about polyominoes. One of his concerns was the design of a fast algorithm for computing the number of polyominoes that tile the plane by translations. What he really had in mind was probably their enumeration according to some convenient parameter. The algorithmic approach, by providing computational evidence, is a convenient way to get some insight about the algebraicity or rationality of certain classes of polyominoes. Let us recall some achievements along these lines.

Tilings, regular or not, have puzzled lots of people from ancient times up to now; and even now, despite of the efforts of many mathematicians these objects remain mysterious. Indeed these objects reveal an incredible amount of simple to state problems that translate into very complex combinatorial ones [1, 4, 9], like for example the squaring of a square [10]. Golomb [11] in his book presents many aspects of polyominoes and in particular he searches how to tile a finite

---

*Keywords and phrases.* Polyominoes, tiling the plane by translation, theorem of Beauquier-Nivat, pseudo-square, pseudo-hexagon, enumeration of special classes of polyominoes.

<sup>1</sup> Laboratoire d'Informatique Fondamentale de Marseille, CNRS UMR 6166, Université de la Méditerranée, 163 avenue de Luminy - Case 901, 13288 Marseille Cedex 9, France;  
Ian.Gambini@lif.univ-mrs.fr

<sup>2</sup> Laboratoire de Mathématiques, CNRS UMR 5127, Université de Savoie, 73376, le Bourget-du-Lac, France; Laurent.Vuillon@univ-savoie.fr

© EDP Sciences 2007

figure of the plane by polyominoes. The enumeration of general polyominoes is a difficult problem and there is no closed formula to count them. However, recent work, conducted mainly by the Bordeaux school and its satellites, allowed to enumerate some very restrictive classes like the directed, parallelogram, convex ones according to various parameters such as the half-perimeter, area, height, width, and some other refinements [4, 8, 14, 15]. More precisely closed formula are known for parallelogram polyominoes [8], for symmetry classes of parallelogram polyominoes [15], polyominoes with notion of convexity [3, 5] and symmetry classes of convex polyominoes in the square lattice [14].

Nevertheless, from the algorithmic point of view Nivat and Beauquier found a characterization of polyominoes that tile the plane by translations [2]. We use this characterization to build our algorithm for deciding if a given polyomino tiles the plane by translation. The methods of this article use techniques from algorithmic, discrete geometry and combinatorics on words.

## 2. DEFINITIONS AND NOTATION

A *polyomino* is a simply connected union of unit squares, that is a union of unit squares without holes. Let  $P$  be a polyomino. A *tiling by translations* of  $P$  is a partition of the whole plane by translated images of  $P$ . A polyomino that tiles the plane by translation is called a *tile*.

Let  $\Sigma = \{a, b, \bar{a}, \bar{b}\}$  be a four letter alphabet. A *reduced word* on  $\Sigma$  is a word on the free group over  $\Sigma$  where all cancellations are done (namely each occurrence of  $a\bar{a}$ ,  $\bar{a}a$ ,  $b\bar{b}$  and  $\bar{b}b$  is replaced by  $\epsilon$  the empty word). Let  $\mathbf{b}(P)$  be the *boundary word* of  $P$  that is the reduced word in the free group on  $\{a, b\}$  where  $a$  represents a right step,  $b$  an up step,  $\bar{a}$  a left step and  $\bar{b}$  a down step that codes the boundary of the polyomino  $P$  in the following way. Starting from an origin on the boundary of  $P$ , the boundary word  $\mathbf{b}(P)$  is the concatenation of labels of boundary unit segments read in trigonometric order. The starting point is not meaningful. Thus the boundary word  $\mathbf{b}(P)$  is a cyclic word.

We define the  $\bar{\phantom{x}}$  operator on  $\Sigma^+$  by

- (i)  $\overline{(\alpha)} = \bar{\alpha}$  if  $\alpha \in \Sigma = \{a, b, \bar{a}, \bar{b}\}$ ;
- (ii)  $\overline{(u.v)} = \overline{(v)}.\overline{(u)}$ ;
- (iii)  $\overline{\bar{a}} = a$  and  $\overline{\bar{b}} = b$ .

The following characterization of tiling polyominoes is due to Beauquier and Nivat [2]:

**Theorem 1** (Beauquier, Nivat). A polyomino  $P$  tiles the plane by translations if and only if the boundary word  $\mathbf{b}(P)$  is equal up to a cyclic permutation of the symbols to  $X \cdot Y \cdot Z \cdot \bar{X} \cdot \bar{Y} \cdot \bar{Z}$  where one of the variables in the factorization may be empty.

If the boundary word is equal to  $X \cdot Y \cdot Z \cdot \bar{X} \cdot \bar{Y} \cdot \bar{Z}$  (resp.  $X \cdot Y \cdot \bar{X} \cdot \bar{Y}$ ) such a polyomino is called *pseudo-hexagon* (resp. *pseudo-square*).

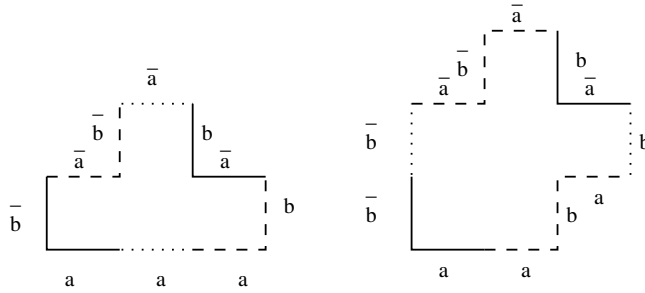


FIGURE 1. Polyominoes and factorizations.

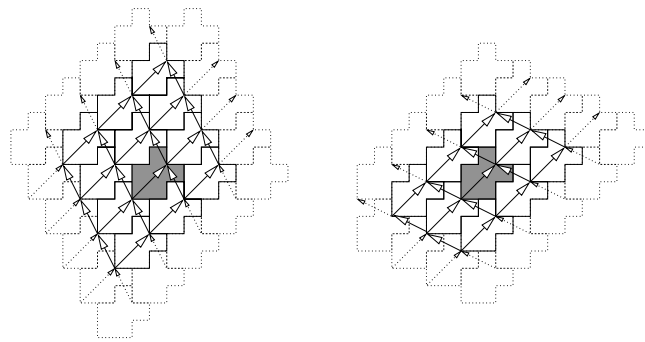


FIGURE 2. Two regular tilings.

For example, the polyomino on the left in Figure 1 is a pseudo-hexagon and the boundary word is equal to  $X \cdot Y \cdot Z \cdot \bar{X} \cdot \bar{Y} \cdot \bar{Z} = a \cdot ab \cdot \bar{a}b \cdot \bar{a} \cdot \bar{b}a \cdot \bar{b}a$  (where  $X = a, Y = ab, Z = \bar{a}b$ ).

In fact, a polyomino  $P$  may have many factorizations of its contour word. For example, in Figure 1 the boundary word of the right polyomino has the factorizations  $\bar{b}a \cdot aba \cdot b \cdot \bar{a}b \cdot \bar{a}b\bar{a} \cdot \bar{b}$  and  $\bar{b}a \cdot a \cdot bab \cdot \bar{a}b \cdot \bar{a} \cdot \bar{b}a\bar{b}$ .

A *regular tiling* is a tiling by translations of a polyomino  $P$  such that each tile in the tiling has the same surrounding by translated copies of the tile  $P$  according to a given factorization of its contour word (such tilings are also called in the literature lattice tilings); see Figure 2 for two regular tilings from the two factorizations of the contour word mentioned above. Each factorization leads to a regular tiling of the plane by translations as follows. If  $P$  is a pseudo-hexagon, the factorization  $\mathbf{b}(P) = X \cdot Y \cdot Z \cdot \bar{X} \cdot \bar{Y} \cdot \bar{Z}$  defines 6 sides of the tile where the sides in correspondence are identified by the pairings  $X, \bar{X}, Y, \bar{Y}, Z, \bar{Z}$ . The translations corresponding to these pairings allow then to tile the whole plane in a regular way. In the case of pseudo squares the construction with 4 sides is similar. Observe that two distinct factorizations of the boundary word of  $P$  give two distinct regular tilings of the plane.

## 3. ALGORITHM

Let  $n = |\mathbf{b}(P)|$  be the length of the boundary word of  $P$ . In the following algorithm the indices of the boundary word  $\mathbf{b}(P)$  (or  $\mathbf{b}$  for simplicity from here on when there are indices) are taken modulo  $n$ . For example, the letter  $\mathbf{b}[-1]$  is of course the letter  $\mathbf{b}[n-1]$ .

```

PS=0; PH=0;
// Step 1: Searching correspondence between  $\mathbf{b}[0]$  and  $\overline{\mathbf{b}[i]}$ 
for  $i = 1$  to  $n - 1$ 
  if  $\mathbf{b}[0] = \overline{\mathbf{b}[i]}$  then
    // Step 2: Propagation
    // Searching the largest  $\mathbf{b}[x_1..y_1] = \overline{\mathbf{b}[x_2..y_2]}$ 
    //  $\mathbf{b}[x_1..y_1]$  containing 0 and  $\mathbf{b}[x_2..y_2]$  containing  $i$ 
     $x_1 = 0; y_1 = 1; x_2 = i; y_2 = i + 1;$ 
    while  $(\mathbf{b}[x_1 - 1] = \overline{\mathbf{b}[y_2]}) \{x_1 = x_1 - 1, y_2 = y_2 + 1\}$ 
    while  $(\mathbf{b}[y_1] = \overline{\mathbf{b}[x_2 - 1]}) \{y_1 = y_1 + 1, x_2 = x_2 - 1\}$ 
    // end of Step 2
     $U = \mathbf{b}[y_1 + 1..x_2 - 1]; V = \mathbf{b}[y_2 + 1..x_1 - 1];$ 
    if  $|U| = |V|$  then
      if  $U = \overline{V}$  then PS=PS+1 // Step 3: Pseudo-square
      PH=PH+ KMP( $\overline{U}, VV$ ) // Step 4: Pseudo-hexagon
    end if
  end if
end for

```

The end of this section explains the algorithm step by step.

**Instance:** the boundary word  $\mathbf{b}[0..n-1]$  of length  $n$  of a polyomino  $P$ .

**Answer:** the number of factorizations in pseudo-squares and in pseudo-hexagons tiling the plane by translation.

**Step 1.** For each position  $i$  from 1 to  $n-1$ , we try to match with the complementary letter of value  $\mathbf{b}[0]$ .

**Step 2.** We make a propagation (scanning back and forth the boundary words) in order to have two sides of maximal length in correspondence. In other words, for each position of value  $\mathbf{b}[0]$ , by propagation we find two complementary words  $X$  and  $\overline{X}$  on the boundary word starting from  $X = \mathbf{b}[0]$  and  $\overline{X} = \overline{\mathbf{b}[0]}$  and extending the pair of complementary words  $(X, \overline{X})$  in order to find the longest  $X$ . Then by this method we find a factorization of  $\mathbf{b}(P)$  by  $X \cdot U \cdot \overline{X} \cdot V$ .

**Step 3.** We now check that the remaining sides  $U$  and  $V$  have same length. And we answer that the polyomino is a pseudo-square if  $U = \overline{V}$  that is if we have found a factorization on  $X \cdot Y \cdot \overline{X} \cdot \overline{Y}$  with  $Y = U$ .

**Step 4.** We check if the polyomino is a pseudo-hexagon by searching four more sides in two-by-two correspondence, that is the factorizations  $U = Y \cdot Z$  and  $V = \overline{Y} \cdot \overline{Z}$ . We use the following property: if such factorization exists, it is provided by an occurrence of the word  $\overline{U} = \overline{Z} \cdot \overline{Y}$  in  $VV = \overline{Y} \cdot \overline{Z} \cdot \overline{Y} \cdot \overline{Z}$ .

This part can be done for instance by the KMP algorithm of Knuth *et al.* [13] or by the algorithm of Boyer-Moore [6].

**Answer**

The variable PS (resp. PH) gives the number of factorizations of  $\mathbf{b}(P)$  (the boundary word of  $P$ ) by pseudo-squares (resp. pseudo-hexagons). If PS=0 and PH=0 then  $P$  does not tile the plane by translation.

### 3.1. PROOF OF THE ALGORITHM

By step 2 of the algorithm, we have the following property. For each position of  $\overline{\mathbf{b}[0]}$ , by propagation we find two complementary words  $X$  and  $\overline{X}$  on the boundary word starting from  $X = \mathbf{b}[0]$  and  $\mathbf{b}[i] = \overline{X} = \overline{\mathbf{b}[0]}$  and extending the pair of complementary words  $(X, \overline{X})$  in order to find the longest  $X$ . Then by this method we find a factorization of  $\mathbf{b}(P)$  by  $X \cdot U \cdot \overline{X} \cdot V$ .

By this method, for each couple  $(\mathbf{b}[0], \mathbf{b}[i] = \overline{\mathbf{b}[0]})$  the algorithm finds by propagation a unique couple  $(X, \overline{X})$  with  $X$  of maximal length. Step 3 and Step 4 find a factorization if it exists.

Thus given a boundary word of a polyomino  $P$  there are 3 cases to consider, either there is factorization A) by pseudo-hexagon or B) a factorization by pseudo-square or C) no factorization. And for each case we have to prove that the algorithm finds it.

- $P$  is a pseudo-hexagon.

In this case the boundary word can be factorized up to a cyclic permutation of letters on  $X \cdot Y \cdot Z \cdot \overline{X} \cdot \overline{Y} \cdot \overline{Z}$  and we may assume without loss of generality that  $X$  contains the letter  $\mathbf{b}[0]$  (otherwise we make a cyclic permutation of letters). As the algorithm propagates to the left and to the right for each position of  $\overline{\mathbf{b}[0]}$ , let be  $\ell$  the  $\ell^{th}$  element of  $X$  corresponding to  $\mathbf{b}[0]$ .  $X_\ell = \mathbf{b}[0]$ , so  $\overline{X}_{n-\ell+1} = \overline{\mathbf{b}[0]}$  where  $|X| = |\overline{X}| = n$ . The Step 2 of the algorithm finds at most a couple of complementary words  $X'$  and  $\overline{X'}$  containing respectively the words  $X$  and  $\overline{X}$ .

a) If  $X' = X$  then with the help of the KMP-algorithm Step 4 produces the good factorization  $X \cdot Y \cdot Z \cdot \overline{X} \cdot \overline{Y} \cdot \overline{Z}$ .

b) When  $|X'| > |X|$  there is a difficulty and we proceed by contradiction. Assuming that the algorithm finds such pair of complementary words  $(X', \overline{X'})$  then  $X' = LXR$  and  $Y'$  does exist such that the factorization of  $P$  is equal to  $X \cdot Y \cdot Z \cdot \overline{X} \cdot \overline{Y} \cdot \overline{Z} = X \cdot RY' \cdot Z' \cdot \overline{R} \cdot \overline{X} \cdot \overline{L} \cdot \overline{Y'} \cdot \overline{Z'} \cdot L$ . By this equality we have  $Y = RY'$ ,  $Z = Z' \cdot \overline{R}$ . Thus  $\overline{Y} = \overline{Y'} \cdot \overline{R}$ ,  $\overline{Z} = R \cdot \overline{Z'}$ . If we use this information in the factorization  $X \cdot Y \cdot Z \cdot \overline{X} \cdot \overline{Y} \cdot \overline{Z}$  we obtain  $X \cdot Y \cdot Z \cdot \overline{X} \cdot \overline{Y'} \cdot \overline{R} \cdot R \cdot \overline{Z'}$ . We find a contradiction because all the letters of  $\overline{R} \cdot R$  cancel two by two ( $\overline{R} \cdot R = \overline{r}_p \cdots \overline{r}_2 \overline{r}_1 r_1 r_2 \cdots r_p = \overline{r}_p \cdots \overline{r}_2 r_2 \cdots r_p = \cdots = \epsilon$ ). This means in particular that the boundary word of  $P$  is not a reduced word and by construction the boundary word of  $P$  is a reduced word.

- $P$  is a pseudo-square.

Here the boundary word can be factorized up to a cyclic permutation of letters on  $X \cdot Y \cdot \overline{X} \cdot \overline{Y}$  and we assume also that  $X$  contains the letter  $\mathbf{b}[0]$ . As in the previous case, the algorithm finds at most a couple of complementary words  $X'$  and  $\overline{X'}$  containing respectively the words  $X$  and  $\overline{X}$ .

a) If  $X' = X$  then by step 3 it finds the good factorization in  $X \cdot Y \cdot \overline{X} \cdot \overline{Y}$ .

b) When  $X' \neq X$  then there is another difficulty. We proceed by contradiction. Assume that the algorithm finds  $X' = LXR$  and then it exists  $Y'$  such that  $X \cdot Y \cdot \overline{X} \cdot \overline{Y} = X \cdot RY'\overline{R} \cdot \overline{X} \cdot \overline{L} \overline{Y'}\overline{L}$ . By this equality we have  $Y = RY'\overline{R}$ . Thus  $\overline{Y} = \overline{RY'}\overline{R}$ . If we use this information in the factorization  $X \cdot Y \cdot \overline{X} \cdot \overline{Y}$  we obtain  $X \cdot RY'\overline{R} \cdot \overline{X} \cdot \overline{RY'}\overline{R}$ . If we replace  $R$  by  $r_1r_2 \cdots r_p$  where  $r_i$ 's are letters then

$$X \cdot Y \cdot \overline{X} \cdot \overline{Y} = Xr_1r_2 \cdots r_p Y' \overline{r_p} \cdots \overline{r_2} \overline{r_1} \cdot \overline{X} \cdot r_1r_2 \cdots r_p \overline{Y'} \overline{r_p} \cdots \overline{r_2} \overline{r_1}.$$

There are many factorizations in pseudo-square and we will work on the following one  $Xr_1 \cdot Y'' \cdot \overline{r_1} \overline{X}r_1 \cdot \overline{Y''} \cdot \overline{r_1}$  where  $Y'' = r_2 \cdots r_p Y' \overline{r_p} \cdots \overline{r_2}$ ,  $\overline{Y''} = r_2 \cdots r_p \overline{Y'} \overline{r_p} \cdots \overline{r_2}$  in order to show that the contour word is not one of a polyomino.

We have  $Xr_1 Y'' \overline{r_1} \overline{X}r_1 \overline{Y''} \overline{r_1}$  and we will now show by an argument of discrete geometry that this is not a boundary word of a simply connected union of unit squares (*i.e.* of a polyomino). In this decomposition  $r_1$  is just a letter then for the reasoning we will take  $r_1 = a$  (the reasoning is the same with  $r_1 = b, \bar{a}, \bar{b}$ ). We use tools from discrete geometry introduced by Daurat and Nivat [7]. Since a polyomino is a simply connected union of squares (the boundary word delimits squares inside the polyomino  $P$  (noted I-squares) and squares outside the polyomino  $P$  (noted O-squares)). A corner on the boundary of  $P$  is called *salient* if it is surrounded by one I-square and three O-squares. A corner on the boundary of  $P$  is called *reentrant* if it is surrounded by three I-squares and 1 O-square. Daurat and Nivat proved in [7] that for any polyomino  $P$  the number  $S(P)$  of its salient points and the number  $R(P)$  of its reentrant points satisfy  $S(P) = R(P) + 4$ , see Figure 3. For example, if  $P$  is a pseudo-square with boundary word  $XY\overline{X}\overline{Y}$  the number of salient points associated with  $X$  is equal to the number of reentrant points associated with  $\overline{X}$  (by this reasoning  $S(X) = R(\overline{X})$ ,  $S(Y) = R(\overline{Y})$ ,  $R(X) = S(\overline{X})$  and  $R(Y) = S(\overline{Y})$ ). Thus by Daurat-Nivat theorem, it follows that the four points where  $X, \overline{X}, Y, \overline{Y}$  connect are salient.

In our case we have the factorization in  $XaY''\overline{a}\overline{X}a\overline{Y''}\overline{a}$  and on the plane we have to place 4 segments associated with  $a, \bar{a}, a, \bar{a}$  according to the factorization. In fact each segment determines two points on the boundary. We find the same relation as in the previous example:  $S(X) = R(\overline{X})$ ,  $S(Y) = R(\overline{Y})$ ,  $R(X) = S(\overline{X})$  and  $R(Y) = S(\overline{Y})$  and we just have to consider the 8 remaining points. The factorization is  $XaY''\overline{a}\overline{X}a\overline{Y''}\overline{a}$

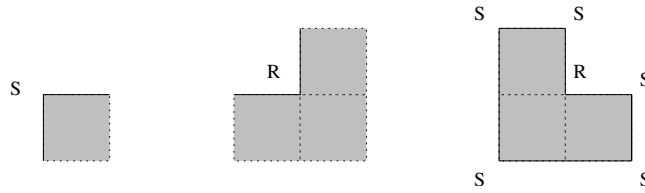


FIGURE 3. Salient and reentrant points.

with  $X = x_1 \cdots x_m$  and  $Y'' = y_1 \cdots y_n$ . Then we have to compute the difference between  $S(x_m a) + S(a y_1) + S(y_n \bar{a}) + S(\bar{a} \bar{x}_m) + S(\bar{x}_1 a) + S(a \bar{y}_n) + S(\bar{y}_1 \bar{a}) + S(\bar{a} x_1)$  and  $R(x_m a) + R(a y_1) + R(y_n \bar{a}) + R(\bar{a} \bar{x}_m) + R(\bar{x}_1 a) + R(a \bar{y}_n) + R(\bar{y}_1 \bar{a}) + R(\bar{a} x_1)$ . But if the point associated with two letters  $uv$  is salient (resp. reentrant) then by construction the point associated with  $\bar{v} \bar{u}$  is reentrant (resp. salient). If  $S(uv) = 1$  then  $R(\bar{v} \bar{u}) = 1$ . By this property  $S(x_m a) + S(a y_1) + S(y_n \bar{a}) + S(\bar{a} \bar{x}_m) + S(\bar{x}_1 a) + S(a \bar{y}_n) + S(\bar{y}_1 \bar{a}) + S(\bar{a} x_1) = R(x_m a) + R(a y_1) + R(y_n \bar{a}) + R(\bar{a} \bar{x}_m) + R(\bar{x}_1 a) + R(a \bar{y}_n) + R(\bar{y}_1 \bar{a}) + R(\bar{a} x_1)$ . And globally for the polyomino  $P$  associated with the boundary word  $X a Y'' \bar{a} \bar{X} \bar{a} \bar{Y}'' \bar{a}$  we have  $S(P) = R(P)$ . This is in contradiction with the result on salient and reentrant points. Thus  $P$  is not simply connected and cannot be a polyomino.

- $P$  does not tile the plane.

In this case by the characterization of Beauquier and Nivat there is no factorization on  $X \cdot Y \cdot \bar{X} \cdot \bar{Y}$  nor  $X \cdot Y \cdot Z \cdot \bar{X} \cdot \bar{Y} \cdot \bar{Z}$ . Then the algorithm fails in steps 3 and 4 to find a characterization and answer that  $P$  does not tile the plane by translation.

### 3.2. COMPLEXITY OF THE ALGORITHM

Let  $n$  be the length of the boundary word associated with  $P$ .

In the first step, the algorithm tries to find all the positions of value  $\overline{\mathbf{b}[0]}$  in the boundary word with complexity  $O(n)$ .

In step 2, the propagation give complexity  $O(n)$ . Thus the total complexity for steps 1 and 2 is  $O(n \times n)$ .

In step 3, we try to find a factorization by checking if  $U = \bar{V}$  and the complexity of this verification is  $O(n)$ . Remark also that step 3 makes just the continuation of step 2 and thus the complexities are added. Thus the total complexity for steps 1, 2 and 3 remains  $O(n \times (n + n))$ .

In step 4, we try to find a factorization by using the KMP algorithm and according to the complexity of KMP algorithm this step is on  $O(m + k)$  where  $m$  is the length of  $VV$  and  $k$  the length of  $U$ . Remark that step 4 just make the continuation of step 2 and step 3 and then we add the complexity of both parts. Thus the computation of the total complexity of the algorithm gives an algorithm on  $O(n \times (n + n + (n + n))) = O(n^2)$ .

## 4. ENUMERATION OF POLYOMINOES BY COMPUTER

We can use our algorithm to compute the number of polyominoes with only pseudo-square factorizations (only PS), with only pseudo-hexagon factorizations (only PH) and with both factorizations in pseudo-square and pseudo-hexagon. The last column is the number of polyominoes of length  $n$  that tile the plane by translations. In fact, factorizations exist only for even-length perimeter and of course there is no factorization for odd-length perimeter. In literature, authors use the half-perimeter in order to enumerate the polyominoes that is the length of the boundary word divided by 2. The enumeration of polyominoes in this table is just the sequence A002931 of the On-line Encyclopedia of integer sequences by Sloane.

We present the result for half-perimeters being between 2 and 18.

Half-perimeter	polyominoes	only PS	only PH	both	tiles
2	1	1	0	0	1
3	2	0	0	2	2
4	7	0	4	3	7
5	28	0	20	8	28
6	124	1	82	17	100
7	588	8	298	46	352
8	2938	40	1007	103	1150
9	15268	170	3326	220	3716
10	81826	523	10394	513	11430
11	449572	1624	31918	1126	34668
12	2521270	4729	95767	2529	103025
13	14385376	13448	282816	5688	301952
14	83290424	37180	824720	12989	874889
15	488384528	102074	2383628	29630	2515332
16	2895432660	276668	6828850	68569	7174087
17	17332874364	745724	19452798	159064	20357586
18	104653427012	1999420	55084940	371115	57455475

In the spirit of the works of Leroux *et al.* [14, 15], we will complete this study by investigating symmetry classes of pseudo-hexagons and pseudo-squares. These results help to understand better the combinatorics of the polyominoes that tile the plane by translations and may be useful for deriving a closed formula or a recurrence relation for the number of pseudo-squares or pseudo-hexagons or regular tilings. In this direction Alberto del Lungo and co-authors obtained by using the so-called ECO method the enumeration of parallelogram polyominoes (polyominoes with two non-crossing paths from an origin to an end with only right and up steps) and convex polyominoes [1, 8]. The counting of pseudo-square and pseudo-hexagon parallelogram polyominoes begs therefore for a closed formula and Alberto asked us this question in July 2001, but still now we don't have the method to enumerate such classes of polyominoes.



## REFERENCES

- [1] E. Barcucci, A. Del Lungo, E. Pergola and R. Pinzani, ECO: a methodology for the Enumeration of Combinatorial Objects. *J. Difference Equ. Appl.* **5** (1999) 435–490.
- [2] D. Beauquier and M. Nivat, On translating one polyomino to tile the plane. *Discrete Comput. Geom.* **6** (1991) 575–592.
- [3] M. Bousquet-Mélou, A method for the enumeration of various classes of column-convex polygons. *Discrete Math.* **154** (1996) 1–25.
- [4] M. Bousquet-Mélou. Habilitation. LABRI Université de Bordeaux 1 (1996).
- [5] S.J. Chang and K.Y. Lin. Rigorous results for the number of convex polygons on the square and honeycomb lattices. *J. Phys. A* **21** (1988) 2635–2642.
- [6] T.H. Cormen, C.E. Leiserson and R.L. Rivest, *Introduction to algorithms*. Chapt. 34, MIT Press (1990) 853–885.
- [7] A. Daurat and M. Nivat. *Salient and Reentrant Points of Discrete Sets*, in *Proc. of the ninth International Workshop on Combinatorial Image Analysis (IWCIA 2003)*, volume 12 of Electronic Notes in Discrete Mathematics. Elsevier (2003).
- [8] A. Del Lungo, E. Duchi, A. Frosini and S. Rinaldi, *Enumeration of convex polyominoes using the ECO method*, in *Discrete Models for Complex Systems, DMCS'03*, edited by M. Morvan and É. Rémy, Discrete Mathematics and Theoretical Computer Science Proceedings AB, 103–116.
- [9] M. Delest and X. Viennot, Algebraic languages and polyominoes enumeration. *Theoret. Comput. Sci.* **34** (1984) 169–206.
- [10] I. Gambini, A Method for Cutting Squares Into Distinct Squares. *Discrete Appl. Math.* **98** (1999) 65–80.
- [11] S.W. Golomb. *Polyominoes*, Princeton science library (1994).
- [12] P. Hubert and L. Vuillon. Complexity of cutting words on regular tilings. *Eur. J. Combin.* **28** (2007) 429–438.
- [13] D.E. Knuth, J.H. Morris and V.R. Pratt. Fast pattern matching in strings. *SIAM J. Comput.* **6** (1997) 323–350.
- [14] P. Leroux, E. Rassart and A. Robitaille, Enumeration of symmetry classes of convex polyominoes in the square lattice. *Adv. Appl. Math.* **21** (1998) 343–380.
- [15] P. Leroux and E. Rassart, Enumeration of symmetry classes of parallelogram polyominoes. *Ann. Sci. Math. Québec* **25** (2001) 53–72.

Communicated by C. Choffrut.

Received February 11, 2005. Accepted April 3, 2006.