An algorithm for deciding if a polyomino tiles the plane by translations

Ian Gambini* and Laurent Vuillon†

Abstract: *We explain a fast algorithm to decide if a polyomino tiles the plane by translations. More precisely, if the polyomino has a boundary word of length n then the algorithm decides if the polyomino tiles the plane by translations in $O(n^2)$ operations (rather than $O(n^4)$ operations for the naive method). This new algorithm uses techniques from algorithmic, discrete geometry and combinatorics on words.*

Keywords: Polyominoes, Tiling of the plane by translation, Theorem of Beauquier-Nivat, Pseudo-square, Pseudo-hexagon, Enumeration of special class of polyominoes.

1 Introduction

During the DMCCG conference held at the institut Henri Poincaré 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 algebricity or rationality of certain classes of polyominoes. Let us recall some achievements along these lines.

*Laboratoire d'Informatique Fondamentale de Marseille, Parc Scientifique et Technologique de Luminy, 163 avenue de Luminy - Case 901, 13288 Marseille Cedex 9, France, Ian.Gambini@lif.univ-mrs.fr

†Laboratoire de Mathématiques, CNRS UMR 5127, Université de Savoie, 73376, le Bourget-du-Lac, France, Laurent.Vuillon@univ-savoie.fr

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 remains mysterious. This kind of problems are complex combinatorial problems [5, 11, 1], like for example the squaring of a square [13]. The activity of tilings and studying polyominoes have many interests in mathematics. Golomb [14] in is book presents many aspects of polyominoes and in particular he searches how to tile a finite figure of the plane by polyominoes. Thurston [19] and Fournier [12] give algorithms to tile a finite figure of the plane without hole by dominoes. This study for various forms of polyominoes can be made by Caley graphs and gives the method of tilings groups [7, 19]. Let mention that tiling the plane by polyominoes is practically an hard problem and in general is an undecidable problem [3].

Recent work, conducted mainly by the Bordeaux school and its satellites, allowed to enumerate some very restrictive classes like the directed ones, parallelogram, convex according to various parameters such that the half-perimeter, area , height, width, and some other refinements [5, 10, 18, 17]. The problem of enumeration in general is hard and still now we don't have closed formula for enumerating polyominoes. The enumeration formulas was found for parallelogram polyominoes [10], for symmetry classes of parallelogram polyominoes [18], polyominoes with notion of convexity [4, 6] and symmetry classes of convex polyominoes in the square lattice [17].

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

2 Definitions and notation

A *polyomino* or tile 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.

Let $\Sigma = \{a, b, \bar{a}, \bar{b}\}$ be a four letter alphabet. A *reduced word* on Σ is a word on the free group over Σ where all cancellations are done (namely each occurrence of $a\bar{a}$, $\bar{a}a$, $b\bar{b}$ and $\bar{b}b$ are replaced by ϵ 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 bottom 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 $\overline{u}$ operator on Σ^+ by
$(i)\overline{(\alpha)} = \overline{\alpha}$ if $\alpha \in \Sigma = \{a, b, \bar{a}, \bar{b}\}$;
$(ii)\overline{(u.v)} = \overline{(v)}.\overline{(u)}$.

The following characterization of tiling polyominoes is 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 $X \cdot Y \cdot Z \cdot \overline{X} \cdot \overline{Y} \cdot \overline{Z}$ where one of the variable in the factorization may be empty.*

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

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

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

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 boundary word (such tilings are also called in the literature lattice tilings) see Figure 2 for two regular tilings from the two factorizations of the boundary 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 $\mathbf{b}(P) = X \cdot Y \cdot Z \cdot \overline{X} \cdot \overline{Y} \cdot \overline{Z}$ defines 6 sides of the tile where the sides in correspondence are identified by the pairings $(X, \overline{X})$,$(Y, \overline{Y})$ and $(Z, \overline{Z})$. 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.

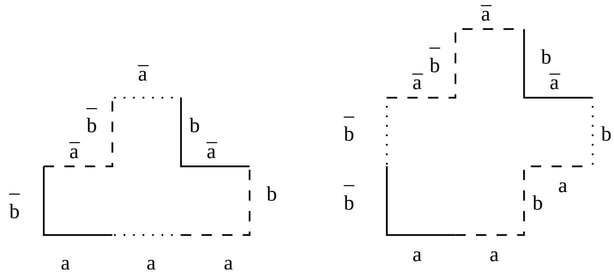

Figure 1: Polyominoes and factorizations.

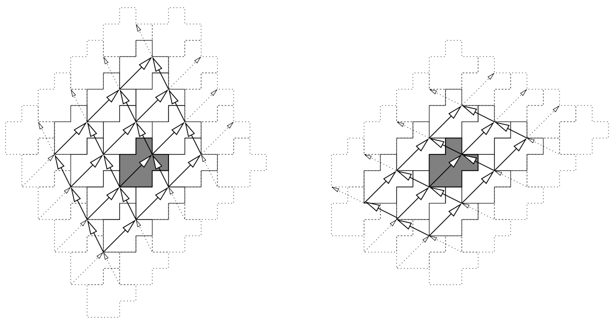

Figure 2: two regular tilings.

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 in the sequel when there are indices) must be taken modulo n and will be between $[0, \cdots, n-1]$. For example, the letter $\mathbf{b}[-1]$ is of course the letter $\mathbf{b}[n-1]$.

PS=0; PH=0;
// Step 1: Searching i such that $\mathbf{b}[0] = \overline{\mathbf{b}[i]}$
for $i = 1$ to $n - 1$
    if $\mathbf{b}[0] = \overline{\mathbf{b}[i]}$ **then**
        // Step 2: Propagation
        // Searching the largest $\mathbf{b}[x_1..y_1] = \overline{\mathbf{b}[x_2..y_2]}$
        // $\mathbf{b}[x1..y1]$ *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 $(\mathbf{b}[x_1 - 1] = \overline{\mathbf{b}[y_2]})$ $\{x1 = x1 - 1, y_2 = y_2 + 1\}$

while ($\mathbf{b}[y_1] = \overline{\mathbf{b}[x_2 - 1]}$) $\{y_1 = y_1 + 1, x_2 = x_2 - 1\}$
$U = \mathbf{b}[y_1..x_2 - 1]$; $V = \mathbf{b}[y_2..x_1 - 1]$;
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 word) in order to have two sides in correspondence of maximal length. In other words, for each position of value $\overline{\mathbf{b}[0]}$, we find by propagation 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 $X \cdot U \cdot \overline{X} \cdot V$.

A necessary condition to find a solution is to have X and $\overline{X}$ well placed. To do this, we just check if the remaining sides U and V have same length.

Step 3

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 factorization $U = Y \cdot Z$ 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, Maurris and Pratt or by the algorithm of Boyer-Moore [8].

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 $\overline{\mathbf{b}[0]}$, we find by propagation 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 ℓ the ℓ^{th} element of X corresponding to $\mathbf{b}[0]$. It exists ℓ such that $X_\ell = \mathbf{b}[0]$, so $\overline{X}_{|X|-\ell-1} = \overline{\mathbf{b}[0]}$. The Step 2 of the algorithm finds using $X_\ell = \mathbf{b}[0]$ and $\overline{X}_{|X|-\ell-1} = \overline{\mathbf{b}[0]}$ at most a couple of complementary words X' and $\overline{X'}$ containing respectively the words X and $\overline{X}$.

  a) If $X' = X$ then Step 4 produces with the help of the KMP-algorithm 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. Assume that the algorithm finds such pair of complementary words $(X', \overline{X'})$ then $X' = LXR$ with $R \neq \epsilon$ or $L \neq \epsilon$. Suppose that $R \neq \epsilon$, the demonstration is identical if we suppose $L \neq \epsilon$. If $|R| \leq |Y|$ it exists Y' and Z' such that the factorization of P is equal to $X \cdot Y \cdot Z \cdot \overline{X} \cdot \overline{Y} \cdot \overline{Z} = X \cdot RY' \cdot Z' \; \overline{R} \cdot \overline{X} \cdot \overline{L} \; \overline{Y'} \cdot \overline{Z'} L$. By this equality we have $Y = RY', Z = Z' \; \overline{R}$. 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 $X \cdot Y \cdot Z \cdot \overline{X} \cdot \overline{Y'} \; \overline{R} \cdot R \; \overline{Z'}$. 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.

If $|R| > |Y|$, the reasoning is the same.

- 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 $X \cdot Y \cdot \overline{X} \cdot \overline{Y} = X \cdot RY' \overline{R} \cdot \overline{X} \cdot \overline{L} \; \overline{Y'} L$. By this equality we have $Y = RY' \overline{R}$. Thus $\overline{Y} = R \overline{Y'} \; \overline{R}$. If we use this information in the factorization $X \cdot Y \cdot \overline{X} \cdot \overline{Y}$ we obtain $X \cdot RY' \overline{R} \cdot \overline{X} \cdot R \overline{Y'} \; \overline{R}$. If we replace R by $r_1 r_2 \cdots r_p$ where r_i's are letters then

  $$X \cdot Y \cdot \overline{X} \cdot \overline{Y} = X r_1 r_2 \cdots r_p Y' \overline{r}_p \cdots \overline{r}_2 \overline{r}_1 \cdot \overline{X} \cdot r_1 r_2 \cdots r_p \overline{Y'} \; \overline{r}_p \cdots \overline{r}_2 \overline{r}_1.$$

  There are many factorizations in pseudo-square and we will work on the following one $X r_1 \cdot Y'' \cdot \overline{r_1} \; \overline{X} r_1 \cdot \overline{Y''} \cdot \overline{r_1}$ 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 boundary word is not one of a polyomino.

  We have $X r_1 Y'' \overline{r_1} \; \overline{X} r_1 \overline{Y''} \; \overline{r_1}$ 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 [9]. 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 one O-square. Daurat and Nivat proved in [9] 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 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.

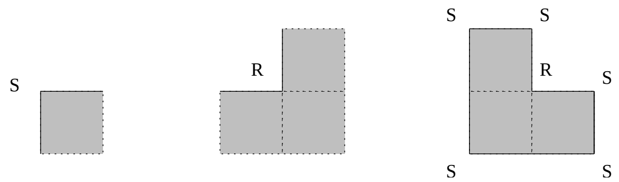

Figure 3: Salient and reentrant points.

In our case we have the factorization in $XaY''\overline{a}\ \overline{X}a\overline{Y''}\ \overline{a}$ and we have to place on the plane 4 segments associated with $a, \bar{a}, a, \bar{a}$ according to the factorization. In fact each segment determines on the boundary two points. We find the same relation than 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 $XaY''\overline{a}\ \overline{X}a\overline{Y''}\ \overline{a}$ with $X = x_1 \cdots x_m$ and $Y'' = y_1 \cdots y_n$. Then we have to compute the difference between $S(x_m a) + S(ay_1) + S(y_n\overline{a}) + S(\overline{a}\ \overline{x_m}) + S(\overline{x_1}a)) + S(a\overline{y_n}) + S(\overline{y_n}\ \overline{a}) + S(\overline{a}x_1)$ and $R(x_m a) + R(ay_1) + R(y_n\overline{a}) + R(\overline{a}\ \overline{x_m}) + R(\overline{x_1}a)) + R(a\overline{y_n}) + R(\overline{y_n}\ \overline{a}) + R(\overline{a}x_1)$. But if the point associated with two letters uv is salient (resp. reentrant) then by construction the point associated with $\overline{v}\ \overline{u}$ is reentrant (resp. salient). If if $S(uv) = 1$ then $R(\overline{v}\ \overline{u}) = 1$. By this property $S(x_m a) + S(ay_1) + S(y_n\overline{a}) + S(\overline{a}\ \overline{x_m}) + S(\overline{x_1}a)) + S(a\overline{y_n}) + S(\overline{y_n}\ \overline{a}) + S(\overline{a}x_1) = R(x_m a) + R(ay_1) + R(y_n\overline{a}) + R(\overline{a}\ \overline{x_m}) + R(\overline{x_1}a)) + R(a\overline{y_n}) + R(\overline{y_n}\ \overline{a}) + R(\overline{a}x_1)$. And globally for the polyomino P associated with the boundary word $XaY''\overline{a}\ \overline{X}a\overline{Y''}\ \overline{a}$ 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 $X \cdot Y \cdot \overline{X} \cdot \overline{Y}$ nor $X \cdot Y \cdot Z \cdot \overline{X} \cdot \overline{Y} \cdot \overline{Z}$. Then the algorithm fails in Step 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 try to find all the positions of value $\overline{\mathbf{b}[0]}$ in the boundary word with complexity $O(n)$.

In step 2, the propagations gives 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 $U = \overline{V}$ 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 make just 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 for the length of the perimeter with odd values and of course there is no factorization for even length of the perimeter. In the literature, authors use the half-perimeter in order to enumerate the polyominoes.

We present the result for half-perimeter 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	2 938	40	1 007	103	1 150
9	15 268	170	3 326	220	3 716
10	81 826	523	10 394	513	11 430
11	449 572	1 624	31 918	1 126	34 668
12	2 521 270	4 729	95 767	2 529	103 025
13	14 385 376	13 448	282 816	5 688	301 952
14	83 290 424	37 180	824 720	12 989	874 889
15	488 384 528	102 074	2 383 628	29 630	2 515 332
16	2 895 432 660	276 668	6 828 850	68 569	7 174 087
17	17 332 874 364	745 724	19 452 798	159 064	20 357 586
18	104 653 427 012	1 999 420	55 084 940	371 115	57 455 475

In the spirit of the works of Leroux, Rassart and Robitaille [17, 18], we will complete this study by investigating symmetry classes of pseudo-hexagons and pseudo-squares. These researches help in better understand the combinatorics of the polyominoes that tile the plane by translations and to find a close 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 find enumeration by ECO method 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, 10]. Maybe the next step will be to enumerate pseudo-square parallelogram polyominoes and pseudo-hexagon parallelogram polyominoes. Alberto posed this question to us 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*, Journal of Difference Equations and Applications, **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] R. Berger, *The undecidability of the domino problem*, Mem. Amer. Math. Soc. 66, 1966.

[4] M. Bousquet-Mélou, *A method for the enumeration of various classes of column-convex polygons*, Discrete Math., **154** (1996), 1–25.

[5] M. Bousquet-Mélou, Habilitation. LABRI Université de Bordeaux 1, 1996.

[6] S. J. Chang and K. Y. Lin, *Rigorous results for the number of convex polygons on the square and honeycomb lattices*, J. Phys. A: Math. Gen., **21** (1988), 2635–2642.

[7] J.H. Conway and J.C. Lagarias, *Tilings with polyominoes and combinatorial group theory*, J. Combin. theory, **53** (1990), 183–208.

[8] Cormen, T.H., Leiserson, C.E., Rivest, R.L., *Introduction to algorithms*, Chapter 34, pp 853-885, MIT Press, 1990.

[9] A. Daurat and M. Nivat, *Salient and reentrant points of discrete sets*, in Proc. of the nineth International Workshop on Combinatorial Image Analysis (IWCIA 2003), volume 12 of Electronic Notes in Discrete Mathematics. Elsevier, 2003.

[10] 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 , Michel Morvan an Éric Rémila (eds.), Discrete Mathematics and Theoretical Computer Science Proceedings AB, 103–116.

[11] M. Delest and X. Viennot,*Algebraic languages and polyominoes enumeration*, Theor. Comp. Sci., **34** (1984), 169–206.

[12] J. C. Fournier, *Pavage des figures planes sans trous par des dominos: Fondement graphique de l'algorithme de Thurston, parallélisation, unicité et décomposition*, Theor. Comp. Sci., **159** (1997), 105–128.

[13] I. Gambini, *A method for cutting squares into distinct squares*, Discrete Applied Mathematics, **98** (1999), 65-80.

[14] S. W. Golomb, *Polyominoes*, Princeton science library 1994.

[15] P. Hubert and L. Vuillon, *Complexity of cutting words on regular tilings*, preprint LAMA number 03-10a.

[16] D.E. Knuth, J.H. Morris and V.R.Pratt, *Fast pattern matching in strings*, SIAM Journal of computing, Vol 6 **2**, (1997), 323–350.

[17] P. Leroux, E. Rassart and A. Robitaille, *Enumeration of symmetry classes of convex polyminoes in the square lattice*, Advances in Apllied Mathematics, **21** (1998), 343–380.

[18] P. Leroux and E. Rassart, *Enumeration of symmetry classes of parallelogram polyminoes*, Annales des Sciences Mathématiques du Québec, **25** (2001), 53–72.

[19] W.P. Thurston, *Conway's tiling group*, Amer. Math. Monthly (1990), 757–773.