Theoretical Computer Science 502 (2013) 76–87

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Exhaustive generation of gominoes

Jérôme Fortier [a,*], Alain Goupil [b], Jonathan Lortie [a], Jérôme Tremblay [a]

[a] LaCIM, Université du Québec à Montréal, C. P. 8888 Succursale "Centre-Ville", Montréal (QC), H3C 3P8, Canada

[b] Département de mathématiques et d'informatique, Université du Québec à Trois-Rivières, 3351 boul. des Forges, C.P. 500, Trois-Rivières (QC), G9A 5H7, Canada

ARTICLE INFO

Keywords:
Polyomino
Site-perimeter
Gomino
Inscribed polyomino

ABSTRACT

In this paper, we consider the generation of three classes of polyominoes, distinguished by their connectivity type. We present a two-player game called *gomino*, and we show how this game induces an algorithm to generate these sets of polyominoes according to their site-perimeter. We then use a variation of the gomino algorithm to obtain exact expressions for the number of polyominoes inscribed in a rectangle of size $b \times k$ with area $bk - r$, where $r \leq 5$.

1. Introduction

The well-known *polyomino*, introduced by Golomb [1] in the field of recreational mathematics, is a combinatorial object which has been related to a variety of challenging problems, such as tilings [2,3], games [4], and enumeration problems. Indeed, some special classes of polyominoes conveniently encode integer partitions, and they are therefore commonly used in number theory and in mathematical physics to describe representations of the symmetric group.

There exist different types of polyomino; here, by *polyomino* we mean a finite edge-connected union of unit lattice squares (pixels) in the discrete plane. (See Fig. 1).

The enumeration problem for general polyominoes is considered difficult and is still open. Nevertheless, the number a_n of polyominoes with n cells has been computed up to $n = 56$ [5,6], and the asymptotic behavior of the sequence $\{a_n\}_{n\geq 0}$ is partially known by the relation $\lim_{n\to\infty} \{a_n\}^{1/n} = \mu$, where $3.90 < \mu < 4.64$ [7,8]. However, several subclasses were successfully enumerated by imposing geometrical constraints on the polyominoes. For instance, it is known [9,10] that the number of parallelogram polyominoes having semi-perimeter $n+1$ is the nth *Catalan number* (sequence M1459 in [11]). We refer the reader to the survey of Viennot [12] for the exact enumeration of various classes of polyominoes. See also [13–15] for further results and [16] for the enumeration of a subclass of parallelogram tiles. Recently, the enumeration of some families of polyominoes inscribed in a rectangle was obtained in [17].

There exists no closed formula for the number of polyominoes according to the area, and therefore brute-force algorithms remain essentially the only method for counting them so far. Several algorithms have been proposed in the literature, and we refer the reader to Redelmeier [18] for an inductive algorithm and to Jensen [5] for a faster one based on the matrix transfer method [10]. Both algorithms are exponential, and the key point in Jensen's algorithm is to generate column configurations that can lead to polyominoes rather than the polyominoes themselves. That strategy saves time, since the number of configurations is less than the number of polyominoes, but the drawback is the space/time tradeoff, since an exponential amount of memory is required in order to store those configurations.

In this paper, we propose yet another method for polyomino generation based on constraining polyominoes with respect to a parameter called the *site-perimeter* [19,20]. The site-perimeter is the number of cells outside a polyomino that touch

* Corresponding author. Tel.: +1 418 580 5181; fax: +1 514 987 8274.
E-mail addresses: jerome.fortier@gmail.com (J. Fortier), Alain.Goupil@uqtr.ca (A. Goupil), lortie.jonathan@courrier.uqam.ca (J. Lortie), tremblay.jerome@uqam.ca (J. Tremblay).

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2012.02.032

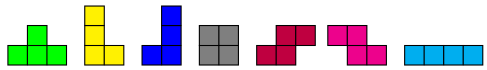


Fig. 1. Some well-known polyominoes with four cells.

its border. Unlike Jensen and Redelmeier, our aim is the exhaustive generation of polyominoes and not only their enumeration. Therefore, the computation time of our algorithm has to be at least proportional to the number of objects that we generate, but the runtime memory requirements are very low. Our approach is inspired by the well-known game *go* [21], and it uses two types of stone, namely black and white.

The paper is structured as follows. Section 2 describes the classes of polyominoes that we are considering and their *go*-like description. Section 3 introduces a game based on *go* that generates upon termination a polyomino having a fixed number of black stones delimiting its contour. The exhaustive generation algorithm is presented in Section 4 with specializations to polyominoes with different types of hole. Some experimental results follow. In Section 5, we show how modified versions of the algorithm can be used to enumerate polyominoes according to their usual perimeter and also to obtain exact formulas for polyominoes inscribed in a rectangle whose area is nearly maximal.

2. Preliminaries

A 4-path (respectively, 8-path) is a path that uses only horizontal and vertical unit steps (respectively, horizontal, vertical, and diagonal steps) as shown below.

(a) 4 directions (b) 8 directions

Accordingly, a path is encoded by letters in the set $\mathcal{M} = \{a, \bar{a}, b, \bar{b}, c, \bar{c}, d, \bar{d}\}$. A subset $S \subseteq \mathbb{Z}^2$ is 4-connected (respectively, 8-connected) if any pair of its points is connected by a 4-path (respectively, 8-path). A polyomino P is a finite 4-connected set of points, usually represented with unit squares, in the $\mathbb{Z}^2$ plane.

Subclasses are obtained by using connectedness constraints. A polyomino P is *simply* 4-connected (respectively, *simply* 8-connected) if its complement $\mathbb{Z}^2 \setminus P$ is 4-connected (respectively, 8-connected). Since the 4-connectedness is stronger than the 8-connectedness, it is clear that the simply 4-connected polyominoes are also simply 8-connected, but not conversely.

The algorithm proposed for the exhaustive generation of polyominoes is sufficiently general to take into account all these types of connectedness. More precisely, we first show how to generate the simply 4-connected polyominoes, then the simply 8-connected polyominoes and finally, the polyominoes that have no special type of connectedness. Fig. 2 presents an example of each case.

A more intuitive definition uses the notion of *hole*.

Definition 1 (*Holes*). Let P be a polyomino. A 4-*hole* (respectively, 8-hole) of P is a bounded 4-connected (respectively, 8-connected) component of $\mathbb{Z}^2 \setminus P$.

The simply 4-connected (respectively, 8-connected) polyominoes are those having no 4-holes (respectively, 8-holes). For instance, in Fig. 2, P_2 has no 8-hole but has one 4-hole, which in this case is 8-connected to the unbounded component of $\mathbb{Z}^2 \setminus P$, so it is not an 8-hole. The polyomino P_3 has one 8-hole, which can also be seen as two 4-holes.

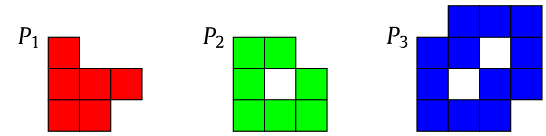
In order to generate a polyomino, we build its *contour*, which is defined as follows (see [19,20] for more details).

Definition 2. The *outside contour* of a polyomino P is the set $P^+ \subseteq \mathbb{Z}^2 \setminus P$ of all points such that every $b \in P^+$ is a 4-neighbor of some point in P. The size of P^+ is called the *site-perimeter* of P. Dually, the *inside contour* of P is the subset $P^- \subseteq P$ such that each of its points is an 8-neighbor of some point in $\mathbb{Z}^2 \setminus P$.

It is clear that the inside and outside contours of a polyomino are unique. When we imagine $\mathbb{Z}^2$ as an infinite *go* board, we can represent the outside contour P^+ of a polyomino P with black stones and the inside contour P^- with white stones (see Fig. 3). Hence we have the following definition.

Definition 3. A *gomino* is a pair $G = (P^+, P^-)$ of sets in $\mathbb{Z}^2$ for which there exists a polyomino P such that P^+ is the outside contour of P and P^- is the inside contour of P.

The analogy with *go*, besides the fact that we need two distinct colors to represent both contours, is that, according to *go* terminology, a simply 4-connected polyomino P along with its outside contour P^+ is *Black's territory*, and P^+ delimits this territory with a minimal number of cells. The algorithm described in the next section generates every gomino with n black stones (hence, every polyomino with site-perimeter n) by alternatively placing black and white stones in a way two *go* players could.



Simply 4-connected   Simply 8-connected   Not simply connected

Fig. 2. The three types of connectedness.

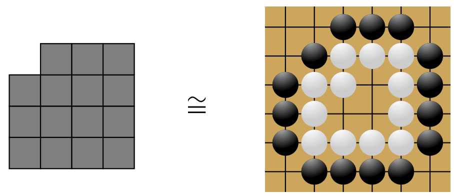


Fig. 3. A polyomino and its gomino representation.

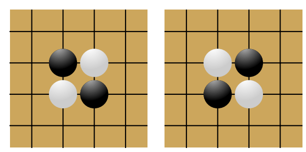


Fig. 4. X-patterns.

3. A solitaire game

The game of *gomino* is in some sense a simpler version of *go*. From a given initial position, two players **B** and **W** walk on the board $\mathbb{N} \times \mathbb{Z}$ while placing a stone on every visited grid point that was previously empty. **B** has n black stones in hand and tries to surround a territory by returning to his/her original position, while **W** tries to prevent **B** from returning by placing white stones.

Before going further, we introduce some useful notation. The sequence of positions of the players after each stage is denoted by two functions

$$\mathbf{B}, \mathbf{W} : \mathbb{N} \to \mathbb{N} \times \mathbb{Z},$$

with starting points

$$\mathbf{B}(0) = (0, 0); \qquad \mathbf{W}(0) = (1, 0). \tag{1}$$

The following notion is useful to efficiently take into account the 4-holes.

Definition 4 (*X-pattern*)**.** An *X-pattern* is a square-shaped set of four stones such that one diagonal is black and the other one is white. There are two possible X-patterns (see Fig. 4).

3.1. Forbidden positions

To prevent repetitions of the same polyomino by translation, we suppose the initial position of **W** to be the lowest stone of the leftmost column of a polyomino P. That point is also called *the root* of P. **B** cannot play below his/her initial position in column 0 and **W** cannot play in column 0. We call the positions in this set *forbidden positions*.

3.2. Rules of movement

Player **B** starts first, and the two players play alternatively according to the following rules.

B: this player moves one unit step anywhere in his/her 8-neighborhood excluding the forbidden positions and the grid points occupied by a white stone. Note that only the moves on empty grid points increase the total number of black stones on the *go* board. Moreover, **B**'s first move must be in direction d. This restriction forces **B** to travel in a counterclockwise manner around some polyomino.

W: this player must 4-connect with **B**'s current position using a minimal number of steps. He/she can do so by moving at each stage as many steps as necessary on empty or white positions in the 8-neighborhood of **B**'s previous place, subject to the two following restrictions.

1. *Right-hand rule.* At the end of each stage, **W** must be located on the left of the arrow describing **B**'s last movement. The right-hand rule forces **W** to remain inside the polyomino whose boundary is built by player **B**.
2. *Hole restriction.* **W** may not create an X-pattern. The hole-restriction avoids patterns creating a 4-hole in the polyomino. This restriction will be lifted later to include polyominoes with 4-holes.

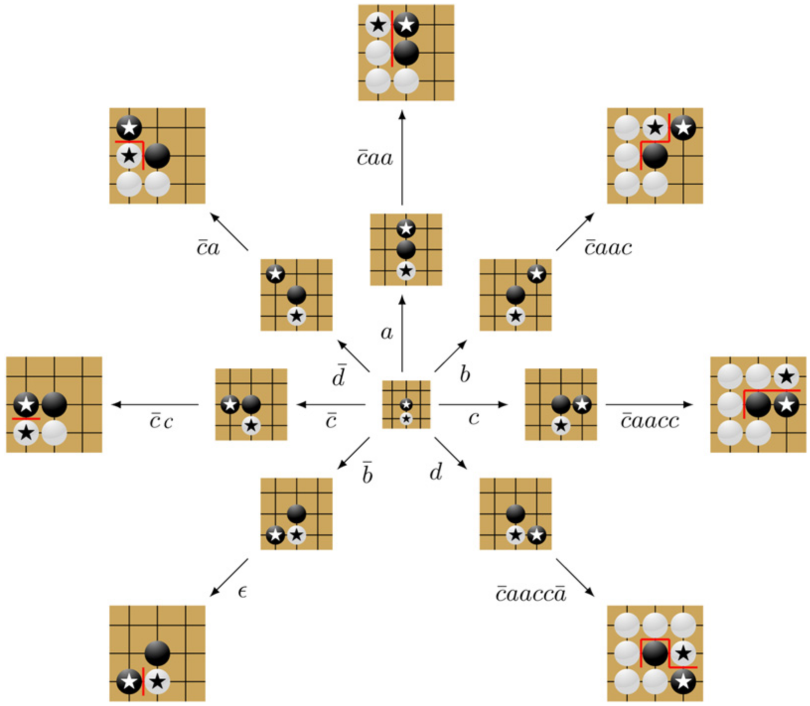


Fig. 5. Allowed moves, up to a rotation.

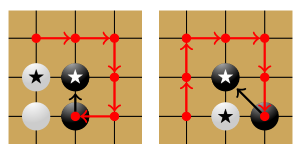


Fig. 6. Order of the 8-neighbourhood.

The multiple stones placed by **W** at a given stage reflect the fact that **B** is surrounding **W** with a minimal number of stones. This means that every black stone is in contact with at least one white stone that **W** eventually places.

Fig. 5 summarizes the set of allowed moves when **W** starts in the position right under **B**. Starred stones (★) indicate the current positions of each player. Note that four rotations of the figure (one for each orientation of **W** with respect to **B**) are required to summarize the complete set of moves, and, since the directions in $\mathcal{M}$ are fixed, the words encoding each move are different from one rotation to another. Observe also that the move $\bar{c}acc\bar{a}$ for **W** and each of its rotations are always avoided because of the hole restriction, but they will be nevertheless needed for the generation of polyominoes with 4-holes.

3.3. Winning conditions

We say that **B** *wins* if he/she succeeds in returning to his/her initial position. In this case, the steps performed by **B** and **W** mark respectively the outside and the inside contours of a unique polyomino P.

If, at a given stage, **W** is unable to 4-connect with **B** by using one of the allowed moves, we say that **W** *wins*. Furthermore, when no sequence of moves leads to a win for **B**, then **W** wins. As a consequence, there is a bijection between polyominoes and gomino games which **B** wins.

Theorem 5. *Every simply* 4*-connected polyomino P is described by a unique gomino game G which* **B** *wins.*

Proof. Let P be a simply 4-connected polyomino, and let P^+ and P^- be its outside and inside contours. We construct, by induction, a gomino game that describes P, and we prove that this game is unique.

Starting with $\mathbf{B}(0)$ and $\mathbf{W}(0)$ as in Eq. (1), we set $\mathbf{W}(0)$ to be the root of P. Since the first move of **B** is d, we have $\mathbf{B}(1) = (1, -1)$, which is also a point of P^+, and $\mathbf{W}(1) = (1, 0)$.

Now suppose that **B** has already made n steps such that every black stone placed so far is in P^+ and every white stone is in P^-. Let $\overrightarrow{b_n}$ denote the arrow from $\mathbf{B}(n-1)$ to $\mathbf{B}(n)$. At stage n, we are, up to a rotation, in one of the two situations of Fig. 6, where the black arrow is $\overrightarrow{b_n}$.

Define $\mathbf{B}(n+1)$ as the first point in P^+ that is encountered by looking clockwise around $\mathbf{B}(n)$, starting from **W**'s current position. This clockwise scan is illustrated by the red path in Fig. 6. Since P^- is the set of all 8-neighbors of P^+ inside P, every

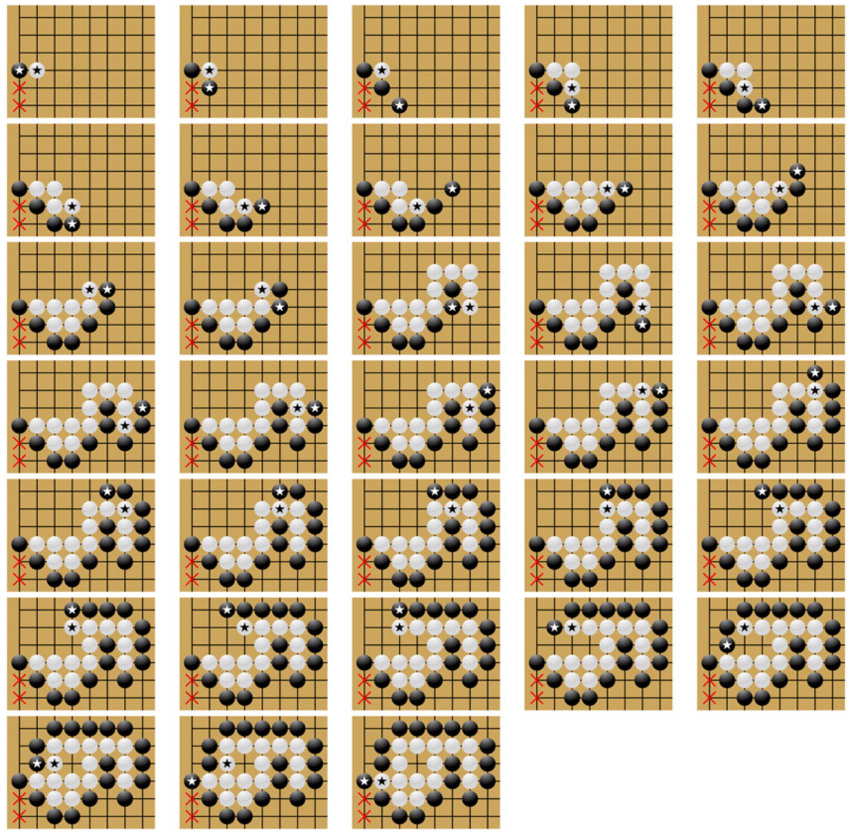


Fig. 7. A full example of a gomino game that leads to a polyomino.

previous point on the red path belongs to P^-. By inspection of **W**'s allowed moves (Fig. 5), we see that the only valid move for **W** is to follow these points until he/she reaches $\mathbf{B}(n+1)$. It is possible to do so since, by induction, no black stone is on P^-.

It remains to prove that our choice of $\mathbf{B}(n+1)$ is the only one possible. But we could not have chosen a previous point since those points belong to P^-, and we want the black stones to cover P^+ for the game to describe P. Also, if we chose another point, then **W** would have had to walk on a point of P^+ to 4-connect with $\mathbf{B}(n+1)$, and the game would not describe P, since the white stones must cover P^-. □

Remark. A direct consequence of the previous theorem is that the sequence of moves performed by **B** when he/she wins describes a unique polyomino, which is conveniently encoded by a word on $\mathcal{M}$. For instance, Fig. 7 presents a complete gomino game which **B** wins, and the resulting polyomino is encoded by the word $w = ddcbba\bar{a}dbaa\bar{d}\bar{c}\bar{c}\bar{c}\bar{c}\bar{b}\bar{a}\bar{b}$. **W**'s empty moves are omitted from the figure and the red crosses on the board mark the forbidden positions.

4. The exhaustive generation algorithm

We now use the bijection between gomino games and polyominoes to describe our main algorithm for generating all simply 4-connected polyominoes. We call this *the gomino algorithm*. Recall that, when we refer to a *game*, it is allowed to be unfinished.

The gomino algorithm. *It accepts a game as input and returns all possible games that can be played from there.*

1. Computation of the list of valid moves

   B *moves: any empty allowed position or black stone in the* 8*-neighborhood of the current position is a valid move.*

   W *moves follow immediately* **B** *moves. As illustrated in* Fig. 5, **W** *has only one valid move. This potential* **W** *move is tried on the board, and it is discarded if it does not respect the restrictions.*
2. Create new games

   With each of the remaining moves, create a new game where it is played. For every n-moves game, one iteration of the algorithm creates up to seven $(n+1)$*-moves games that are added to a set of games in progress. The input game is no longer needed and may be safely discarded.*
3. Apply recursion

   The current games are checked against the winning conditions. **W** *wins are discarded and* **B** *wins, which correspond to polyominoes, are saved.*

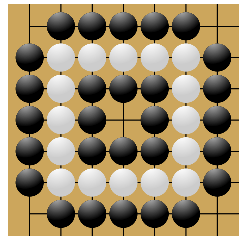
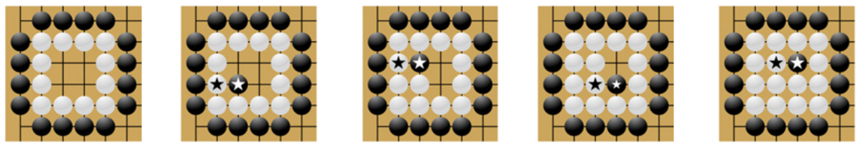


Fig. 8. A gomino with no white liberty.



(a) A gomino. (b) Position 1. (c) Position 2. (d) Position 3. (e) Position 4.

Fig. 9. A gomino and the four initial positions used for enumerating its holes.

The algorithm is then applied again to every remaining game in progress, until all games are finished. Games are independent of each other and can therefore be processed separately.

4.1. 4-holes

We can extend our algorithm to include all simply 8-connected polyominoes by lifting the hole restriction that forbids X-patterns. **W**'s move $\bar{c}aacc\bar{a}$ then becomes valid, marking the smallest holes.

Note that the proof of Theorem 5 does not involve 4-holes. Hence Theorem 5 remains true for simply 8-connected polyominoes where the hole restriction is lifted. However, we may look for new X-patterns at each stage in order to count 4-holes.

Proposition 6. *In any gomino G describing a simply 8-connected polyomino P, there are as many occurrences of the X-pattern in G as there are 4-holes in P.*

Proof. We proceed by induction on the number of 4-holes in a polyomino P. First, consider the case of a single 4-hole. There is at least one X-pattern, because it is the only pattern that prevents a 4-path from connecting the interior of the hole to the exterior while still allowing an 8-path to do so. If there were such a 4-path, there would not be a hole in the first place. If we have two X-patterns and only one 4-hole, there are two 8-paths connecting the 4-hole to the outside of P. Indeed, given a point z in the 4-hole and a point w outside of the 4-hole and outside of P, we can connect z to w with two different 8-paths not crossing P, each path passing through a different X-pattern. These two paths form an 8-loop surrounding a part of P. There must also exist a part of P outside the loop, otherwise z would not be inside a 4-hole. There is no 4-path connecting the inside of the 8-loop to the outside of the 8-loop, so P is not 4-connected: a contradiction. So there is only one X-pattern.

Now, suppose that, for all simply 8-connected polyominoes with $(n-1)$ 4-holes, there are exactly $(n-1)$ X-patterns. Let P be a simply 8-connected polyomino with n 4-holes. There is at least one 4-hole that we can fill to obtain a simply 8-connected polyomino P'. We call this 4-hole the nth 4-hole. The polyomino P' has $(n-1)$ 4-holes, so by induction it has $(n-1)$ X-patterns. Using the same argument as in the initial case, the addition of the nth 4-hole adds exactly one X-pattern. Therefore P has n X-patterns. □

4.2. 8-holes

To build polyominoes with 8-holes, we apply a slightly modified version of the gomino algorithm that considers the empty positions inside the gomino. These positions must be filled with black or white stones until **W** has no more liberty, in which case there is no empty position in the 4-neighborhood of any white stone (see Fig. 8). Note that we do not need to add stones inside 4-holes connected to the exterior: those are already enumerated by the basic algorithm.

As in the basic gomino algorithm, holes are enumerated by fixing their lowest-left cell. To do so, we run the algorithm with every possible starting position, scanned from left to right and from bottom to top. A valid starting position for **B** must already have a white stone on its left. This position is the starting position of **W**. **B** wins if both **B** and **W** have walked back on their initial positions. However, this time, **W** does not necessarily step on his/her initial position immediately after **B**, but **B** always steps back there before **W** does. Hence, **B** wins when **W** steps back on his/her initial position, even if his/her turn is not finished. This ensures that the path followed by **W** is closed and, therefore, bounds a 8-hole. After having enumerated all holes with given **B**(0) and **W**(0), we place a white stone on **B**(0) and let the algorithm search the next starting position. Fig. 9 illustrates the initial positions required to enumerate all possible holes of a given gomino.

Besides the new starting and ending conditions, we must add a new move for **B** in order to enumerate both types of hole. At the beginning of a game, **B** is allowed to win immediately by staying in place. **W** is then forced to make a complete turn

Table 1
Number of polyominoes of site-perimeter n according to their connectivity type.

n	Simply 4-connected	Simply 8-connected	8-holes allowed
4	1	1	1
5	0	0	0
6	2	2	2
7	4	4	4
8	12	12	12
9	32	32	32
10	110	110	110
11	340	340	340
12	1193	1209	1209
13	4080	4256	4272
14	14786	15974	16166
15	53428	60232	61849

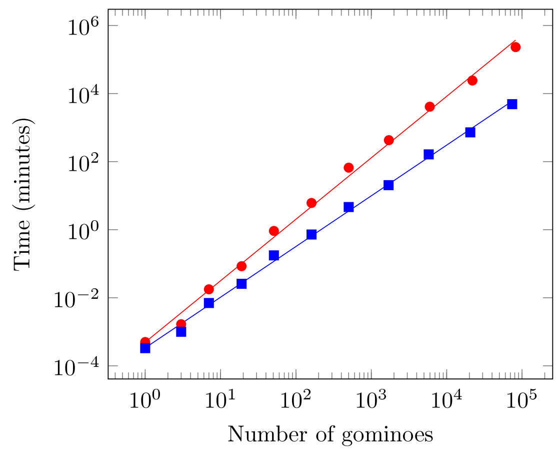


Fig. 10. Computation time for the first two columns of Table 1.

around **B**, thus ending the game. This move is necessary and sufficient to produce single stone holes which would be omitted otherwise.

4.3. Experimental results

We have implemented the three versions of the algorithm with the mathematical software Sage [22] and distributed the computation to several computers with the software |P|P|S|S| (see [23]). The results for the number of polyominoes with a site-perimeter up to $n = 15$ are reported in Table 1.

The size of the tree of games constructed by the algorithm can be roughly bounded above by 7^{2n+1}, since **B** has access to at most seven possible moves and would not perform more than $2n$ moves because it is impossible for him/her to walk more than twice on the same point (**W** could not follow him/her if he/she did). The total number of games, just like the number of games which **B** wins, is therefore bounded by an exponential growth rate. It would be interesting to obtain more precise bounds for those quantities.

In fact, analyzing the relationship between the number of gomino games with $\leq n$ black stones and the number of such games which **B** wins seems to be interesting, as the graph in Fig. 10 suggests a polynomial correspondence between the computation time of the algorithm that we have implemented (which is, basically, the number of games) and the number of polyominoes (simply 4-connected as $\blacksquare$, simply 8-connected as $\bullet$). Indeed, for a given n with $i = 1$ if the hole restriction is applied and $i = 0$ if it is not, let $\gamma_i(n)$ be the total number of gomino games with $\leq n$ black stones and let $\beta_i(n)$ be the number of those games in which **B** wins. Linear regression on the data in Fig. 10, with $R^2 = 0.9985$ for $i = 1$ and $R^2 = 0.9985$ for $i = 0$, suggests that we have

$$\gamma_0(n) \approx \beta_0(n)^{2.0376} \quad \text{and} \quad \gamma_1(n) \approx \beta_1(n)^{1.6665}.$$

5. Enumerative applications

It is possible to modify the gomino algorithm in order to apply it to some other problems on polyominoes. Here, we present two examples.

Table 2
Number of standardized 4-connected side polyominoes of inner site-perimeter r.

r	2	3	4	5	6	7	8	9	10	11
Polyominoes	0	1	1	5	13	54	182	782	3080	13 250

5.1. Enumeration according to the perimeter

Despite the fact that the site-perimeter is the natural parameter of the gomino algorithm, one can use it to generate polyominoes according to their *usual* perimeter. Indeed, it is easy to see that each **W** move in Fig. 5 adds a fixed number of edges to the perimeter of the polyomino. These edges are displayed in red in the figure. Consequently, in order to generate all polyominoes up to a certain perimeter p, one has to count the number of edges added at each stage. **W** is then set to win when the counter exceeds p instead of winning when there are no black stones left.

One easy early-winner heuristic for that version consists in computing, at each stage, the distance between the current position and the initial position. If that distance exceeds the remaining length, then **W** wins, and it becomes useless to continue. It is also worth noting that the code given by the sequence of **B** moves instantly gives the perimeter of the generated polyomino, according to the same principle that **W**'s unique response to a **B** move adds a fixed number of edges.

5.2. Inscribed polyominoes of area $\max - r$

Another modified version of the gomino algorithm can be used in order to obtain exact expressions for the number of polyominoes inscribed in a rectangle that have area close to the area of the rectangle. A polyomino is said to be *inscribed* in a rectangle if it touches the four sides of that rectangle. For a rectangle of size $b \times k$ and a non-negative integer r, let $p_{\max - r}(b, k)$ be the number of polyominoes inscribed in a $b \times k$ rectangle with area $bk - r$. When $r = 0$ and $r = 1$, it is easy to see that $p_{\max - 0}(b, k) = 1$ and $p_{\max - 1}(b, k) = bk$. In this section, we give exact expressions when $r = 2, 3, 4, 5$ and the size of the rectangle is sufficiently large.

The strategy used to obtain exact expressions consists in removing r cells from a $b \times k$ rectangle full of cells in all $\binom{bk}{r}$ ways and then discarding the forbidden cases where removed cells break the remaining set of cells in several 4-connected components. In such a situation, a smaller polyomino can be formed in the corner, on the side, or in the center of the rectangle. The gomino algorithm can be used to enumerate those situations, when we change the rules of the game.

Given a rectangle $R \subset \mathbb{Z}^2$ and a polyomino $P \subset R$, we say that P is a *corner polyomino* if it touches exactly two incident sides of R and contains the corresponding corner cell of R. Similarly, P is a *side polyomino* if it touches exactly one side of R. In both cases, the *inner site-perimeter of P with respect to R* is defined to be $|P^+ \cap R|$.

We want to generate, up to translation, every corner and side polyomino according to their inner site-perimeter. We are assuming that, for a rectangle R of size $b \times k$, we have $b, k > r$, and, to fix ideas, we assume that the corner and side polyominoes touch, respectively, the bottom-left corner and the left side of R. We call the polyominoes generated with these assumptions *standardized*. By using the same ideas as in Theorem 5, it is easy to see that standardized polyominoes are generated using the gomino algorithm with the following two games.

Game 1 (*The Side Gomino Game*).
Make the following changes to the rules of the standard game.

- ***Black stones count.*** *The stones that* **B** *places on column* 0 *are not counted, the others are counted, and* **B** *may place at most r counted black stones.*
- ***Forbidden positions.*** *Same as in the classical game, but row r is added to the forbidden positions for* **B**. *This is to prevent* **B** *from going too far on column* 0 *by leaving uncounted black stones behind.*

The numbers of standardized simply 4-connected side polyominoes for small values of the inner site-perimeter r, generated with Game 1, are given in Table 2.

Game 2 (*The Corner Gomino Game*).
Make the following changes to the rules of the standard game.

- ***Black stones count.*** *The stones that* **B** *places on row* -1 *or column* 0 *are not counted, the others are counted, and* **B** *may place at most r counted black stones.*
- ***Forbidden positions.*** *Same as in the classical game, with new forbidden positions for* **B** *at row* -2, *row r, and column r. Row* -2 *is forbidden to make sure that the polyomino is inscribed in a corner and the other new forbidden positions are again to prevent* **B** *from going too far with uncounted black stones.*

Again, we have generated the simply 4-connected corner polyominoes for small values of the inner site-perimeter, and the results appear on the bottom line of Table 3. An exact formula can be given for a subclass of corner polyominoes. Let us call $C(r, h)$ the number of standardized corner polyominoes with inner site-perimeter $r < b, k$ and leftmost black stone of row 0 on column h. Table 3 presents some values of $C(r, h)$.

Table 3
Values of $C(r, h)$ and the number of standardized 4-connected corner polyominoes of inner site-perimeter r.

h	r								
	2	3	4	5	6	7	8	9	10
2	1	1	3	7	25	78	305	1155	4698
3	0	2	4	13	40	155	566	2264	9195
4	0	0	5	13	59	219	892	3620	15147
5	0	0	0	13	50	244	1084	4737	20801
6	0	0	0	0	35	184	999	4944	23607
7	0	0	0	0	0	96	654	3955	21563
8	0	0	0	0	0	0	267	2255	15204
9	0	0	0	0	0	0	0	750	7602
10	0	0	0	0	0	0	0	0	2123
Total	1	3	12	46	209	976	4767	23680	119940

Proposition 7. *The numbers $C(r, r)$ are counting left factors of Motzkin words, and their generating function is the following:*

$$\sum_{r\geq 2} C(r, r)x^r = \frac{x}{2}\left(\sqrt{\frac{(1+x)}{(1-3x)}} - 1\right). \tag{2}$$

Proof. The leftmost black stone of row 0 is on column r, and that stone is counted. The sequence of moves that **B** does to place his/her remaining $(r-1)$ stones can only use directions $\bar{b}$, $\bar{c}$, and $\bar{d}$, since, if **B** makes another move, he/she would run out of stones before reaching column 0. Moreover, by hypothesis, these moves must not make **B** step on row 0, and it is easy to check from Fig. 5 that this is the only possible case in which a white stone prevents **B** from doing one of his/her three possible moves.

The word encoding such a sequence of moves is therefore a left factor of length $(r-1)$ of a Motzkin word, and the generating function of these words is well known (see for instance [24]). $\square$

Rows and diagonals different from the main diagonal of Table 3 do not correspond, to the awareness of the authors, to any known sequences of positive integers. Let us now return to the enumeration of inscribed polyominoes of area max $-r$.

Proposition 8. *For $r = 2, 3, 4, 5$, the numbers $p_{\max -r}(b, k)$ of polyominoes of area $bk - r$ inscribed in a rectangle of size $b \times k$ are given by the following formulas:*

$$p_{\max -2}(b, k) = \binom{bk}{2} - 4, \quad b, k \geq 3. \tag{3}$$

$$p_{\max -3}(b, k) = \binom{bk}{3} - 4(bk-3) - 2(b+k-4) - 12, \quad b, k \geq 4. \tag{4}$$

$$\begin{aligned} p_{\max -4}(b, k) = {} & \binom{bk}{4} - 4\binom{bk-3}{2} - 2(bk-4)(b+k-4) - 8(bk-5) \\ & - 4(bk-6) - (b-2)(k-2) - 2(b+k-6) - 26, \quad b, k \geq 5. \end{aligned} \tag{5}$$

$$\begin{aligned} p_{\max -5}(b, k) = {} & \binom{bk}{5} - 4\binom{bk-3}{3} - 2\binom{bk-4}{2}(b+k-4) - 8\binom{bk-5}{2} \\ & - 4\binom{bk-6}{2} - (b-2)(k-2)(bk-5) - 2(b+k-6)(bk-6) - 26bk \\ & + 4(b+k) + 128, \quad b, k \geq 6. \end{aligned} \tag{6}$$

Proof. As mentioned before, the idea of the proof is to remove r cells from a full $b \times k$ rectangle in all $\binom{bk}{r}$ ways and then look at forbidden cases in which the remaining set of cells is not a polyomino. An important set of forbidden cases appear when a corner polyomino is isolated by removing r cells. For $r = 2$, the only case appears when we remove two cells from a diagonal next to a corner (see Fig. 11) disconnecting the corner cell. There are four such situations, corresponding to each corner of the rectangle, and this explains Eq. (3). For $r = 3$, all forbidden cases appear in Fig. 11. The first three pictures for $r = 3$ in Fig. 11 describe the situations in which a corner polyomino of inner site-perimeter 3 is formed. The next picture in Fig. 11 describes the situation in which a side polyomino of site-perimeter 3 is formed. There are $2(b+k-4)$ such side polyominoes, one for each place around the perimeter of the rectangle. The last picture in Fig. 11 shows the case in which we remove two cells as in the case $r = 2$, and a third cell, represented by a red stone, is removed anywhere among the $bk - 3$ remaining cells outside this corner polyomino and its outside contour. Putting all these cases together gives Eq. (4).

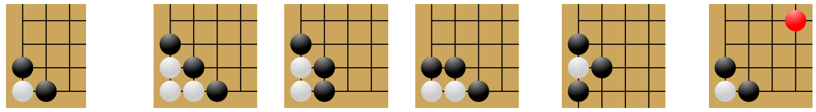


Fig. 11. The forbidden cases of max -2 and max -3.

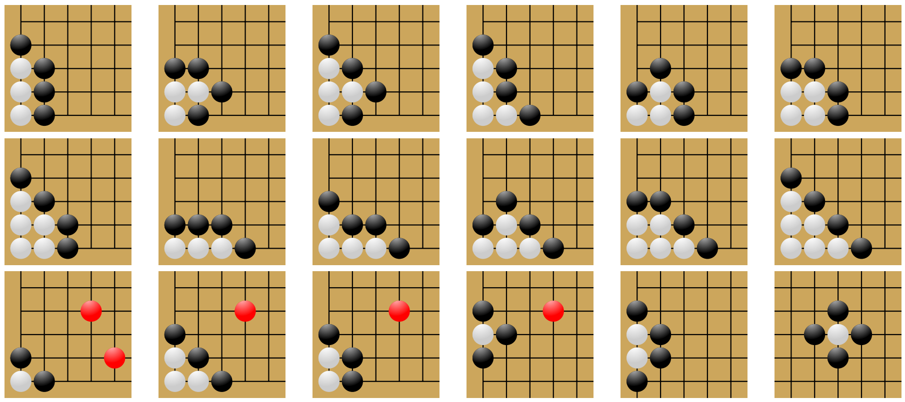


Fig. 12. The forbidden cases of max -4.

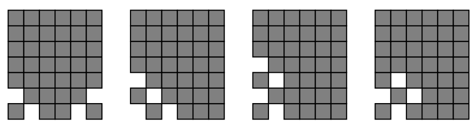


Fig. 13. The forbidden cases of max -4 removed more than once.

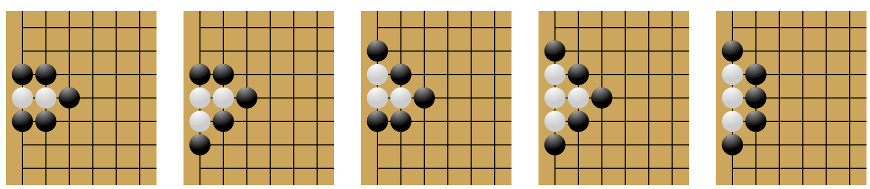


Fig. 14. The side-polyominoes for max -5.

For $r = 4$, the two first rows in Fig. 12 show the 12 corner polyominoes with inner site-perimeter 4. The first picture on the last row in Fig. 12 shows the situation in which a corner cell is isolated with two stones, and the remaining two stones, represented in red, are removed from anywhere in the remaining $bk-3$ positions. The next two pictures show the situation in which a corner polyomino is isolated with three stones, and the remaining stone is taken from the remaining available cells. The fourth and fifth pictures show the cases in which a side polyomino has inner site-perimeter of length 3 and 4, respectively, and the number of side polyominoes is given below the corresponding picture. Finally, there is the case in which a unique cell not on the side of the rectangle is isolated with four stones.

These sets of cases are not all disjoint. There are four situations in which forbidden cases were removed twice. They are shown in Fig. 13. So 22 units have to be added to the count. Thus putting together all these cases gives, for $b, k > 4$,

$$\begin{aligned} p_{\max -4}(b,k) = &\binom{bk}{4} - 4\binom{bk-3}{2} - 2(bk-4)(b+k-4) - 8(bk-5) \\ &- 4(bk-6) - (b-2)(k-2) - 2(b+k-6) - 4\cdot 12 + 22, \end{aligned} \tag{7}$$

which is equal to Eq. (5).

For $r = 5$, the proof is similar to the case $r = 4$, but there are several cases, and we have to be cautious. First, we remove from $\binom{bk}{5}$ the 46 corner polyominoes of inner site-perimeter 5 (see Table 3) in each of the four corners and the five types of side polyomino of inner site-perimeter 5 illustrated in Fig. 14:

$$\text{Five stones: } 4\cdot 46 + 2(b+k-4) + 4(b+k-6) + 4(b+k-8). \tag{8}$$

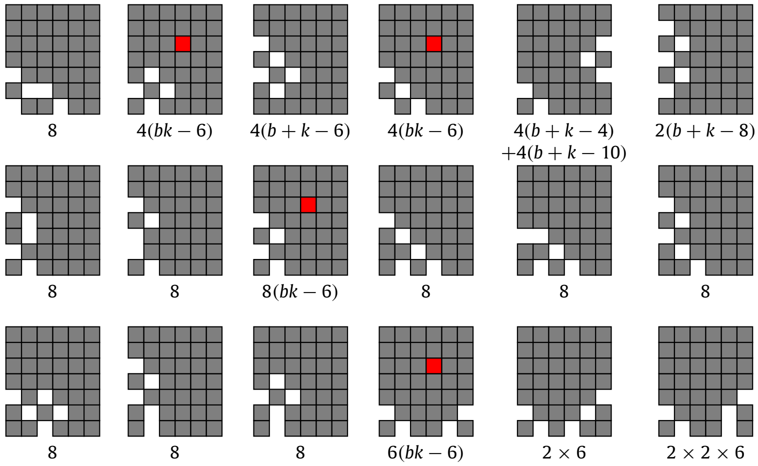


Fig. 15. The forbidden cases of max −5 removed more than once.

We do not show the 46 corner polyominoes here, for obvious reasons. Next, we remove from the expression obtained the 12 corner polyominoes, the side polyomino, and the central polyomino of site-perimeter 4 that appear in Fig. 12, to which are added one free red stone that can be placed anywhere in the rectangle but outside the small polyomino and its site-perimeter:

$$\text{Four stones: } 4\left[5(bk-7)+6(bk-8)+(bk-9)\right]+2(b+k-6)(bk-6)+(b-2)(k-2)(bk-5). \tag{9}$$

Then we remove the 3 corner polyominoes and the side polyomino of inner site-perimeter 3 that appear in Fig. 11, to which are added two red free stones. After that, we remove the unique corner polyomino of inner site-perimeter 2 with three free red stones:

$$\text{Three stones: } \quad 4\left[\binom{bk-6}{2}+2\binom{bk-5}{2}\right]+2\binom{bk-4}{2}(b+k-4). \tag{10}$$

$$\text{Two stones: } \quad 4\binom{bk-3}{3}. \tag{11}$$

After the removal of all the disconnected sets of cells, we have to add those that have been removed more than once. We have collected them in 18 cases that appear in Fig. 15. The number that has to be added because of these cases is the following:

$$108+4(b+k-10)+2(b+k-8)+4(b+k-6)+4(b+k-4)+22(bk-6). \tag{12}$$

If we subtract from $\binom{bk}{5}$ the expressions in Eqs. (8)–(11) and add the expression in (12), we obtain the desired expression for $p_{\max -5}(b, k)$. □

The formulas in the previous proposition have been verified with an algorithm that generates polyominoes of a given area inscribed in a given rectangle. This algorithm is described in [25].

There are also formulas for the number of max $-r$ polyominoes inscribed in a rectangle with precisely r columns and more than r rows that can be obtained from the formulas in Proposition 8 by removing the cases in which an 8-connected segment of length r crosses the r-width of the rectangle. For example, we have

$$p_{\max -3}(3, k) = \binom{3k}{3} - 10(k-2) - 12(k-1) - k - 2 \tag{13}$$

$$p_{\max -4}(4, k) = \binom{4k}{4} - 4\binom{4k-3}{2} - 16\binom{k}{2} - 79(k-2) - 12. \tag{14}$$

For the tenacious reader who would like to pursue the computations presented in the previous proposition, there are more and more forbidden cases that have to be removed more than once, and finding an algorithm for generating them is, to our knowledge, an open problem. What is likely is that the exhaustive generation of corner, side, and inner polyominoes with respect to their site-perimeter comes into play.

6. Conclusion

Since the number of polyominoes grows exponentially with respect to the site-perimeter, polyomino generation, as described in this paper, rapidly becomes impracticable. An interesting problem would be to develop an algorithm for counting polyominoes according to their site-perimeter but avoiding such exhaustive generation. We also believe that the search for exact expressions for the numbers $C(r, h)$ presented in Table 3 is an interesting challenge, since these numbers are related to Motzkin words.

References

[1] S.W. Golomb, Checker boards and polyominoes, American Mathematical Monthly 61 (1954) 675–682.
[2] D. Beauquier, M. Nivat, On translating one polyomino to tile the plane, Discrete & Computational Geometry 6 (1991) 575–592.
[3] S.W. Golomb, Polyominoes: Puzzles, Patterns, Problems, and Packings, Princeton Academic Press, Princeton, 1996.
[4] M. Gardner, Mathematical games, Scientific American (1958) Sept. 182–192, Nov. 136–142.
[5] I. Jensen, Enumerations of lattice animals and trees, Journal of Statistical Mechanics 102 (2001) 865–881.
[6] N.J.A. Sloane, The On-Line Encyclopedia of Integer Sequences, Number of fixed polyominoes with n cells, 2005. http://oeis.org/A001168.
[7] I. Jensen, A.J. Guttmann, Statistics of lattice animals (polyominoes) and polygons, Journal of Physics 33 (2000) L257–L263.
[8] D.A. Klarner, R.L. Rivest, A procedure for improving the upper bound for the number of n-ominoes, Technical Report, Stanford, CA, USA, 1972.
[9] G. Pólya, On the number of certain lattice polygons, Journal of Combinatorial Theory 6 (1969) 102–105.
[10] R.P. Stanley, Enumerative Combinatorics, Cambridge University Press, Cambridge, 1999.
[11] N.J.A. Sloane, The on-line encyclopedia of integer sequences, 2011. Published electronically at http://www.research.att.com/?njas/sequences.
[12] X.G. Viennot, A survey of polyomino enumeration, in: P. Leroux, C. Reutenauer (Eds.), Séries formelles et combinatoires algébrique, 4th International Conference, Montréal, June 15-19, 1992, in: Publications du LACIM, vol. 11, Springer, 1992, pp. 399–420.
[13] M. Bousquet-Mélou, Codage des polyominos convexes et équations pour l'énumération suivant l'aire, Discrete Applied Mathematics 48 (1994) 21–43.
[14] M. Bousquet-Mélou, A method for the enumeration of various classes of column-convex polygons, Discrete Mathematics 154 (1996) 1–25.
[15] M. Bousquet-Mélou, New enumerative results on two-dimensional directed animals, Discrete Mathematics 180 (1998) 73–106.
[16] S. Brlek, A. Frosini, S. Rinaldi, L. Vuillon, Tilings by translation: enumeration by a rational language approach, Electronic Journal of Combinatorics 13 (2006).
[17] A. Goupil, H. Cloutier, F. Nouboud, Enumeration of polyominoes inscribed in a rectangle, Discrete Applied Mathematics 158 (2010) 2014–2023.
[18] D.H. Redelmeier, Counting polyominoes: yet another attack, Discrete Mathematics 36 (1981) 191–203.
[19] M. Bousquet-Mélou, A. Rechnitzer, The site perimeter of bargraphs, Advances in Applied Mathematics 31 (2003) 86–112.
[20] N. Sieben, Polyominoes with minimum site-perimeter and full set achievement games, European Journal of Combinatorics 29 (2008) 108–117.
[21] Go (game)—wikipedia, the free encyclopedia, 2011.
[22] W. Stein, et al., Sage Mathematics Software (Version 4.6.2), The Sage Development Team, 2011. http://www.sagemath.org.
[23] Louwrentius, |P|P|S|S|-(Distributed) Parallel Processing Shell Script, 2011. http://code.google.com/p/ppss.
[24] E.S. Egge, T. Mansour, 132-avoiding two-stack sortable permutations, Fibonacci numbers, and Pell numbers, Discrete Applied Mathematics 143 (2004) 72–83.
[25] H. Cloutier, Énumération de polyominos à deux et à trois dimensions, Master's thesis, Université du Québec à Trois-Rivières, Trois-Rivières, 2010.