

On the Enumeration of Permutominoes

Ana Paula Tomás^(✉)

DCC & CMUP, Faculdade de Ciências, Universidade do Porto, Porto, Portugal
apt@dcc.fc.up.pt

Abstract. Although the exact counting and enumeration of polyominoes remain challenging open problems, several positive results were achieved for special classes of polyominoes. We give an algorithm for direct enumeration of permutominoes [13] by size, or, equivalently, for the enumeration of grid orthogonal polygons [23]. We show how the construction technique allows us to derive a simple characterization of the class of convex permutominoes, which has been extensively investigated [5]. The approach extends to other classes, such as the row convex and the directed convex permutominoes.

1 Introduction

The generation of geometric objects has applications to the experimental evaluation and testing of geometric algorithms. No polynomial time algorithm is known for generating polygons uniformly on a given set of vertices. Some generators employ heuristics [1, 7] or restrict to certain classes of polygons, e.g., monotone, convex or star-shaped polygons [22, 24]. Numerous related problems have also been extensively investigated, as the exact counting or enumeration of polyominoes [10]. These remain challenging open problems in computational geometry and enumerative combinatorics. A *polyomino* is an edge-connected set of unit squares on a regular square lattice (grid). Polyominoes are defined up to translations. In this paper, we give an algorithm for the enumeration of *permutominoes* by *size*, or, equivalently, for the enumeration of *grid orthogonal polygons* by the number of vertices [23]. Research on permutominoes has focused the enumeration of some subclasses of permutominoes according to the size and the characterization of pairs of permutations defining various classes of permutominoes [20]. Polyominoes are usually enumerated by area (i.e., number of cells). The direct enumeration of polyominoes is a computational problem of exponential complexity. An overview of the main developments concerning direct and indirect approaches is given in [3]. Jensen's transfer-matrix algorithm [14] – an indirect method – is currently the most powerful algorithm for counting fixed polyominoes. Exact counts are known for polyominoes that have up to 56 cells [3, 15]. As far as we can see, Jensen's algorithm cannot be adapted for

Partially supported by CMUP (UID/MAT/00144/2013), which is funded by FCT (Portugal) with national (MEC) and European structural funds through the programs FEDER, under the partnership agreement PT2020.

counting permutominoes. Our algorithm for direct enumeration of permutominoes is based on INFLATE-PASTE, a construction technique we developed in [23].

The rest of the paper is organized as follows. Sections 2 and 3 introduce fundamental background, in particular, the INFLATE-PASTE technique. Section 4 describes our enumeration algorithm for generic permutominoes. In Sect. 5, we see how to tailor the approach to count some specific classes (or to create instances in such sets), such as the convex and the row-convex permutominoes. Section 6 concludes the paper.

2 Preliminaries

A polygon is called orthogonal if its edges meet at right angles (of $3\pi/2$ radians at reflex vertices and $\pi/2$ at convex vertices). If r is the number of reflex vertices of an n -vertex orthogonal polygon, then $n = 2r + 4$ (e.g. [19]). The grid orthogonal polygons (*grid ogons*) were introduced in [23] as a relevant class for generation. A *grid ogon* is an orthogonal polygon without collinear edges, embedded in a regular square grid and that has exactly one edge in each line of its minimal bounding square. The grid ogons correspond to the *permutominoes* introduced in [5, 9, 13]. A permutomino is a polyomino that is given by two suitable permutations of $\{1, 2, \dots, r + 2\}$, for $r \geq 0$, which define the sequences of the x and y coordinates of its vertices. Its *size* is $r + 1$ and represents the width of its minimal bounding square. We adopt the notion of size given in [9], which is slightly different from [5] (where it is defined as one plus the width of the minimal bounding square). The topological border of a permutomino of size $r + 1$ is a grid ogon with r reflex vertices, and so, it has $2r + 4$ vertices in total. Conversely, the region delimited by a n -vertex grid ogon is a permutomino of size $r + 1$. All polyominoes we consider are simply-connected and, similarly, all polygons are simple and without holes.

3 The Inflate-Paste Technique

The INFLATE-PASTE technique¹ was proposed in [23] for creating n -vertex grid ogons (i.e., permutominoes), at random, as sketched in Fig. 1.

The algorithm yields an n -vertex grid ogon in $O(n^2)$ time. It exploits the fact that every n -vertex grid ogon results from a unit square by applying INFLATE-PASTE $r = (n - 4)/2$ times. INFLATE-PASTE glues a new rectangle to a grid ogon to obtain a new one with 1 more reflex vertex. The rectangle is glued by PASTE to an horizontal edge incident to a convex vertex v , is fixed at v and must be in a region that we called *the free neighbourhood of v* (Fig. 2b)). This region is denoted by $FSN(v)$ and consists of the external points that are rectangularly visible from v in the quadrant with origin v that contains the horizontal edge $e_H(v)$ and the inversion of the vertical edge $e_V(v)$, incident to v . Here, the inversion of $e_V(v)$ is its reflection with respect to v . Two points z

¹ Demos at <http://www.dcc.fc.up.pt/~apt/genpoly>.

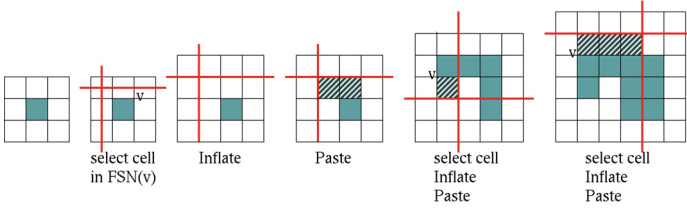


Fig. 1. Using INFLATE-PASTE to create a permutomino with 3 reflex vertices (size 4).

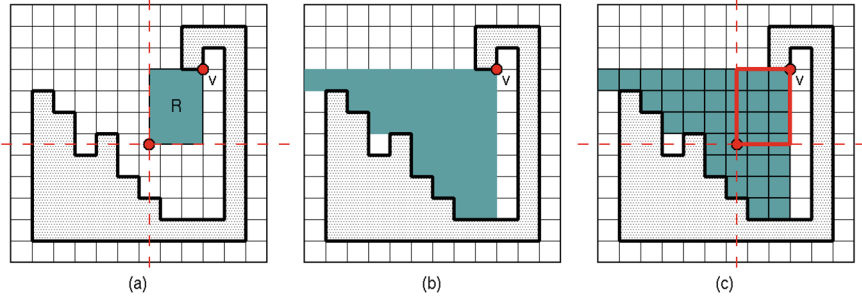


Fig. 2. INFLATE-PASTE: (a) gluing a rectangle to v (b) $FSN(v)$ is the dark shaded region (c) the rectangle is defined by v and the center of a cell of $FSN(v)$.

and w are *rectangularly visible* if the axis-aligned rectangle that has z and w as opposite corners does not intersect the interior of the polygon [18]. At each step, the algorithm selects a convex vertex v and a cell c in $FSN(v)$. The center of c and v define the rectangle that is glued, but the cell is *inflated* first. The INFLATE operation keeps the grid regular: the grid lines are shifted to insert two new lines – one horizontal and one vertical line – for the new edges, that will meet at the center of c . We assume that the starting unit square is inside a 3×3 square, whose boundary contains no edges of P (see Fig. 1). The boundary is kept free along the method. In this way, $FSN(v)$ is always a bounded region and a Ferrers diagram, with origin at v . Hence it can be defined by a sequence of integers, representing the number of cells that form each row of the diagram. Any cell in $FSN(v)$ can be used for growing the polygon using v . Hence, for the instance shown in Fig. 2(c), we can make $9 + 7 + 6 + 4 + 4 + 3 + 2 = 35$ distinct grid ogons using the selected vertex v .

4 Direct Enumeration of Permutominoes

ECO was introduced in [2] as a construction paradigm for the *enumeration of combinatorial objects* of a given class, by performing local transformations that increase a certain parameter (the size) of the objects. In this section we propose a direct enumeration procedure for the grid ogons (i.e., permutominoes) using INFLATE-PASTE. The enumeration algorithm is as follows.

```

PERMUTOMINOENUM( $P, S, G, n_0, r$ )
  if  $r = 0$  then return fi
  MAKEEMPTYSTACK( $T$ )
  while not ISEMPY( $S$ ) do
     $v := \text{POP}(S)$  /*  $v$  has coordinates  $(v_x, v_y)$  */
     $e_H(v) :=$  the horizontal edge of  $P$  that contains  $v$ 
    if ISCONVEX( $v, P$ ) then
      for  $C$  in FREENEIGHBOURHOOD( $v, P, G$ ) do
         $(p, q) :=$  the southwest corner of  $C$ 
        INFLATEGRID( $p, q, G$ )
         $w_1 := \text{NEWVERTEX}(p + 1, v_y)$ 
         $w_2 := \text{NEWVERTEX}(p + 1, q + 1)$ 
         $w_3 := \text{NEWVERTEX}(v_x, q + 1)$ 
        PASTERECTANGLE( $v, [w_1, w_2, w_3], P$ )
        PUSHCONVEX( $[w_1, w_2, w_3], e_H(v), S, P$ )
        OUTPUTPOLYGON( $P, n_0 + 2$ ) /* or increment a counter */
        PERMUTOMINOENUM( $P, S, G, n_0 + 2, r - 1$ )
        CUTRECTANGLE( $v, [w_1, w_2, w_3], P$ )
        POPCONVEX( $[w_1, w_2, w_3], e_H(v), S, P$ )
        DEFLATEGRID( $p + 1, q + 1, G$ )
      done
    fi
    PUSH( $v, T$ );
  done
  while not ISEMPY( $T$ ) do
    PUSH(POP( $T$ ),  $S$ )
  done

```

Here, P is the initial polygon, G a representation of the grid lines, S a stack that contains the convex vertices of P that are available for expansion, n_0 the number of vertices of P and r the maximum number of reflex vertices of the polygons. PERMUTOMINOENUM enumerates recursively all descendants of P that have up to $n_0 + 2r$ vertices. If initially $P := \{(1, 1), (2, 1), (2, 2), (1, 2)\}$ (w.r.t. the standard 2D cartesian coordinate system and given in CCW-order), $S := \{(1, 2), (2, 2)\}$, and $n_0 = 4$, then the algorithm will enumerate (or count) all grid ogons that have up to $2r + 4$ vertices. Nevertheless, in our description of the algorithm in pseudocode, we assume that P is represented by a doubly-linked circular list and that the vertices are linked by pointers to the grid lines that contain them (do not keep their coordinates explicitly). In the same way, the stack S contains pointers to the vertices that are in the stack and the new vertices w_1, w_2 and w_3 keep a similar representation. This means that, in the pseudocode, $p, q, p + 1, v_y, q + 1, v_x$ refer to the pointers to the corresponding horizontal and vertical grid lines (not simple coordinates). INFLATEGRID(p, q, G) shifts the vertical grid lines $x > p$, one position to the right and the horizontal grid lines $y > q$ one position upwards, and inserts two new lines $x = p + 1$ and $y = q + 1$. DEFLATEGRID($p + 1, q + 1, G$) does the reverse operation, removing the lines $x = p + 1$ and $y = q + 1$ from the grid. This implies that the coordinates of the vertices of P change accordingly. A trail-stack T is used to restore the

contents of the stack S in the end of the function (we can make a copy of S at start and use it, instead). This is important for exploring the higher branches of the enumeration search tree. The call to `PUSHCONVEX` puts the new convex vertices on the top of the stack: w_2 and w_3 are convex vertices always and w_1 is convex if it does not lie in the interior of $e_H(v)$. For the correctness of the enumeration algorithm, it does not matter how the algorithm sorts these new vertices to push them onto the stack, but we can adopt a particular order, as we discuss below. After the recursive call to `PERMUTOMINOENUM`, we restore the polygon, the stack and grid before we proceed to the next cell in $FSN(v)$. This is done by the calls to `CUTRECTANGLE`, `POPCONVEX` and `DEFLATEGRID`. During the enumeration procedure, the bottom horizontal edge cannot move upwards, which is ensured by placing the initial unit square in a 2×3 grid.

The correctness and completeness of the `INFLATE-PASTE` construction are proved in [23] and follow from the analysis of the horizontal partition of orthogonal polygons (see Fig. 3). The *horizontal partition* of an n -vertex grid ogon P consists of $r + 1$ rectangles and is obtained by adding all horizontal chords incident to the reflex vertices of P . Each face of the horizontal partition, which is a rectangle, gives a node of its *dual graph* and two nodes are connected by an edge in the graph if the two corresponding rectangles are adjacent. Now, our enumeration procedure is based on the existence of a unique depth-first generating tree for each polygon P , once we fix an order for visiting the dual graph of its horizontal partition. One possibility is to define the order as the one induced by a clockwise walk around the polygon, starting at the lowest rectangle, from its SW-vertex. In Fig. 3, we used numbers to indicate the order in which the rectangles (i.e., the nodes of the dual graph) are found.

Figure 4 shows all permutominoes with at most 8 vertices and their construction using the enumeration algorithm. In the figure, we used crosses to indicate convex vertices that can no longer be used for expanding a polygon (to ensure

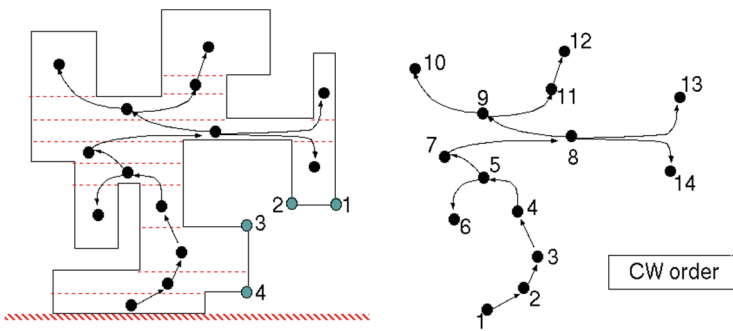


Fig. 3. The horizontal partition of a grid ogon and the unique tree induced by a depth-first visit of its dual graph, if we start from the SW-vertex and walk around the polygon in clockwise order. If we fix this order, only the vertices with labels 1, 2, 3, and 4 remain active for expansion of this instance in our method (with 1 on the top of the stack). The bottom horizontal edge never moves.

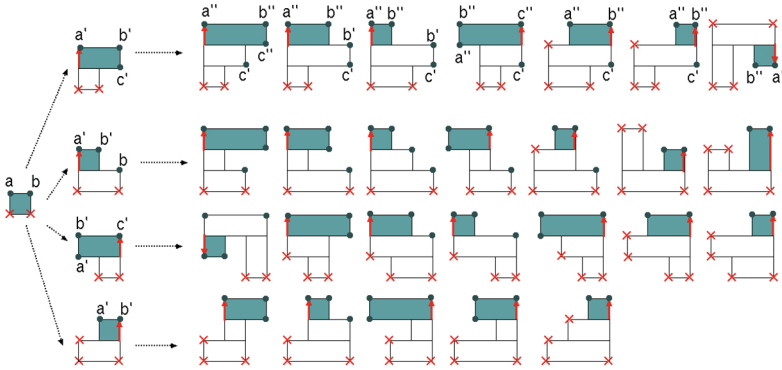


Fig. 4. Permutominoes of size 1, 2, and 3 (i.e., with 4, 6, and 8 vertices) and their horizontal partitions. In each instance, an arrow is attached to the vertex v used in the last step to create the instance (v is no longer a vertex of the polygon).

uniqueness). They will not be in the stack at that stage if the enumeration algorithm proceeds to expand such instance. Since the enumeration is in depth-first, ISCONVEX checks if a vertex is still active, as PASTE can render one vertex reflex.

In contrast to other existing methods for the enumeration of polyominoes, PERMUTOMINOENUM, for permutominoes, does not need to keep an exponential number of state configurations in order to count them correctly. Each permutomino is generated exactly once and, hence, there is no need to check for repetitions. Nevertheless, the running time of the algorithm is dominated by the number of permutominoes generated (and thus it is exponential).

By restricting the set of convex vertices that are active for expansion and the notion of free neighbourhood we can design enumerators for particular subclasses, based on the INFLATE-PASTE construction. In particular, for the convex permutominoes [5, 11], the directed convex permutominoes, the row-convex permutominoes, which, by $\pi/2$ -rotation, yield the column-convex permutominoes [4], as well as, for the thin, spiral, and min-area permutominoes [17].

Indeed, an algorithm for enumerating the *convex* permutominoes by size was published in [11]. Its running cost is proportional to the number of permutominoes generated. It is quite easy to design a specialized version of our algorithm for enumerating convex permutominoes with identical complexity. Actually, as we will see, for convex permutominoes the free neighbourhoods are linear (rectangles of width 1) and only the two topmost convex vertices can be active.

5 Tailoring the Algorithm to Specific Subclasses

Although the exact counting and enumeration of polyominoes remain challenging open problems, several positive results were achieved for special classes of polyominoes [6, 8, 16], namely for the class of convex polyominoes and some

of its subfamilies (e.g., directed-convex polyominoes, parallelogram polyominoes, stack polyominoes, and Ferrers diagrams). The larger class of row-convex (resp. column-convex) polyominoes was considered also [12]. A polyomino is said to be *row-convex* (resp. *column-convex*) if all its rows (resp. *columns*) are connected, i.e., the associated orthogonal polygon is *y-monotone* (*x-monotone*). A polyomino is *convex* if it is both row-convex and column-convex (see Fig. 5).

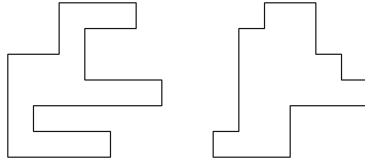


Fig. 5. A row-convex and a convex permutomino.

These classes, which satisfy convexity and/or directness conditions, have been studied using different approaches and are fairly well characterized, for some parameters, e.g., area and perimeter [6]. The corresponding classes of permutominoes have been addressed too [4, 5, 9].

5.1 Convex Permutominoes

The analysis of the transformations performed by INFLATE-PASTE during the application of PERMUTOMINOENUM allow us to derive simple characterizations and exact countings for such classes of permutominoes. Figure 6 shows all n -vertex convex permutominoes for $n = 4, 6, 8$, each one embedded on a grid. Only the two topmost convex vertices can be active for INFLATE-PASTE (so, L and R stand for left or right). Crossed vertices are inactive in the following transformation steps: “u” means that the vertex would be discarded in PERMUTOMINOENUM as well (due to uniqueness conditions) and “c” means that the resulting permutomino would not be convex. The sequence of $\{0, 1, 2\}^*$ displayed on the grid top row is the *expansion key* of the corresponding permutomino. Each element of the key gives the number of active convex vertices that see a certain grid cell (in Fig. 6, each counter is in its cell). Here, *see* means that the cell belongs to the free neighborhood of the vertex, already restricted to account for the convexity condition. For all the remaining empty cells, the counter is 0 and, thus, we omitted it. If we add up the elements of the expansion key of a given convex permutomino, we get the number of convex permutominoes that it yields immediately in PERMUTOMINOENUM. In this way, the expansion keys provide an exact encoding of the structural features that are relevant for counting convex permutominoes according to the number of vertices. Actually, it is the key as a whole that matters but not the particular cells associated to each counter. By analysing INFLATE-PASTE in the scope of PERMUTOMINOENUM, we

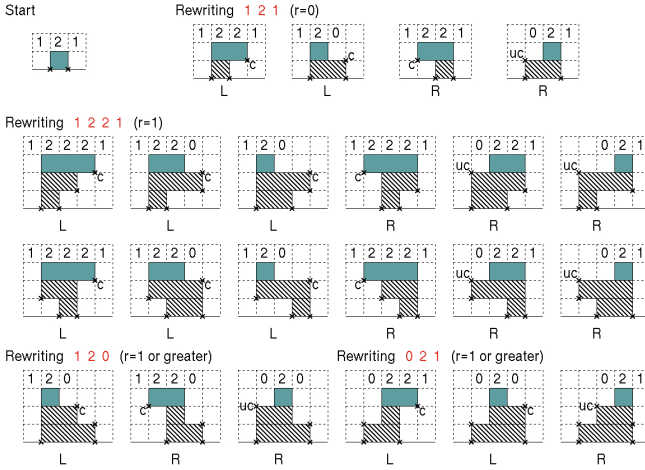


Fig. 6. Enumerating convex permutominoes by size.

may conclude that the *expansion key* of any convex permutomino with $r \geq 0$ reflex vertices must be of one of the following forms:

$$\begin{aligned}
 &1 2^{r+1} 1 \\
 &1 2^j 0, \quad \text{for } 1 \leq j \leq r-1, \\
 &0 2^j 1, \quad \text{for } 1 \leq j \leq r-1, \text{ and} \\
 &0 2^j 0, \quad \text{for } 1 \leq j \leq r-2.
 \end{aligned}$$

INFLATE-PASTE operations acting on convex permutominoes can be seen as rewrite rules. Each rule rewrites the key of a convex permutomino with $r - 1$ reflex vertices to the key of one of the convex permutominoes derived from it, having one more reflex vertex, for $r \geq 1$. The rewrite rules are:

$$\begin{aligned}
 12^r 1 &\xrightarrow{L,R} 12^{r+1} 1 \\
 12^r 1 &\xrightarrow{L} 12^j 0, && \text{for } 1 \leq j \leq r \\
 12^r 1 &\xrightarrow{R} 02^j 1, && \text{for } 1 \leq j \leq r \\
 12^{j'} 0 &\xrightarrow{L} 12^j 0, && \text{for } 1 \leq j \leq j' \leq r-1 \\
 12^{j'} 0 &\xrightarrow{R} 12^{j'+1} 0, && \text{for } 1 \leq j' \leq r-1 \\
 12^{j'} 0 &\xrightarrow{R} 02^j 0, && \text{for } 1 \leq j \leq j' \leq r-1 \\
 02^{j'} 1 &\xrightarrow{R} 02^j 1, && \text{for } 1 \leq j \leq j' \leq r-1 \\
 02^{j'} 1 &\xrightarrow{L} 02^{j'+1} 1, && \text{for } 1 \leq j' \leq r-1 \\
 02^{j'} 1 &\xrightarrow{L} 02^j 0, && \text{for } 1 \leq j \leq j' \leq r-1 \\
 02^{j'} 0 &\xrightarrow{L,R} 02^j 0, && \text{for } 1 \leq j \leq j' \leq r-2
 \end{aligned}$$

where L (left) and R (right) identify the topmost vertex selected. For rules with annotation “ L, R ”, both vertices can be selected, one at a time. Figures 7, 8 and 9 illustrate the idea underlying these rules. The correctness and completeness of this rewrite system can be checked easily by case-analysis, taking into account the conditions on convexity.

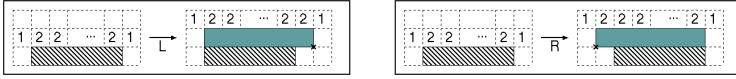


Fig. 7. Rewriting $12^r 1$ using the rewrite rules $12^r 1 \rightarrow^L 12^{r+1} 1$ and $12^r 1 \rightarrow^R 12^{r+1} 1$.

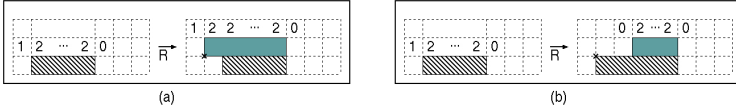


Fig. 8. Rewriting $12^{j'} 0$ using (a) $12^{j'} 0 \rightarrow^R 12^{j'+1} 0$ and (b) a rule $12^{j'} 0 \rightarrow^R 02^j 0$, for $1 \leq j \leq j'$.

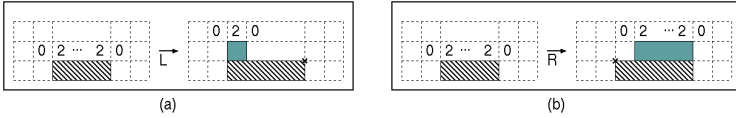


Fig. 9. Rewriting $02^{j'} 0$ using (a) $02^{j'} 0 \rightarrow^L 02^j 0$ and (b) $02^{j'} 0 \rightarrow^R 02^j 0$, for some $1 \leq j \leq j'$.

Proposition 1. Let $C_{\alpha,j,\beta}^{(r)}$ be the number of convex permutominoes of the class $\alpha 2^j \beta$ with r reflex vertices, for $\alpha, \beta \in \{0, 1\}$, $1 \leq j \leq r + 1$ and $r \geq 0$. Then, $C_{0,j,1}^{(r)} = C_{1,j,0}^{(r)}$, for all r and j (symmetry by reflection w.r.t. V -axis) and $C_{\alpha,j,\beta}^{(r)}$ is inductively defined as follows.

$$\begin{aligned}
 C_{1,1,1}^{(0)} &= 1 \\
 C_{1,r+1,1}^{(r)} &= 2C_{1,r,1}^{(r-1)}, \text{ for } r \geq 1 \\
 C_{1,j,0}^{(r)} &= C_{1,r,1}^{(r-1)} + \sum_{j'=\max(1,j-1)}^{r-1} C_{1,j',0}^{(r-1)}, \text{ for } 1 \leq j \leq r \\
 C_{0,j,0}^{(r)} &= 2 \sum_{j'=j}^{r-1} C_{1,j',0}^{(r-1)} + 2 \sum_{j'=j}^{r-2} C_{0,j',0}^{(r-1)}, \text{ for } 1 \leq j \leq r-1
 \end{aligned}$$

The number of convex permutominoes with r reflex vertices (size $r + 1$) is given by $C(r) = C_{1,r+1,1}^{(r)} + 2 \sum_{j=1}^r C_{1,j,0}^{(r)} + \sum_{j=1}^{r-1} C_{0,j,0}^{(r)}$, for $r \geq 0$.

Therefore, using this recurrence, $C(r)$ can be evaluated efficiently using dynamic programming, at least for small values of r , since $C(r)$ grows exponentially. Even for very small values of r , we need to handle big integers (either explicitly or by means of some clever representation). Nevertheless, from [5,9], we know the following closed form for $C(r)$.

$$C(r) = 2(r + 4)4^{r-1} - \frac{r + 1}{2} \binom{2(r + 1)}{r + 1}$$

The first terms of the sequence are listed in [21] (ref. A126020): 1, 4, 18, 84, 394, 1836, 8468, 38632, 174426, 780156, ... In a similar way, we can deduce a recurrence for counting the row-convex permutominoes. In both case, the rewrite rules were useful for deducing the recurrences for counting the polygons.

5.2 Row-Convex Permutominoes

The possible forms of the expansion keys of row-convex permutominoes with r reflex vertices are $1^a 2^b 1^c$, with $a + b + c = r + 3$, and $a, b, c \geq 1$ (see Fig. 10).

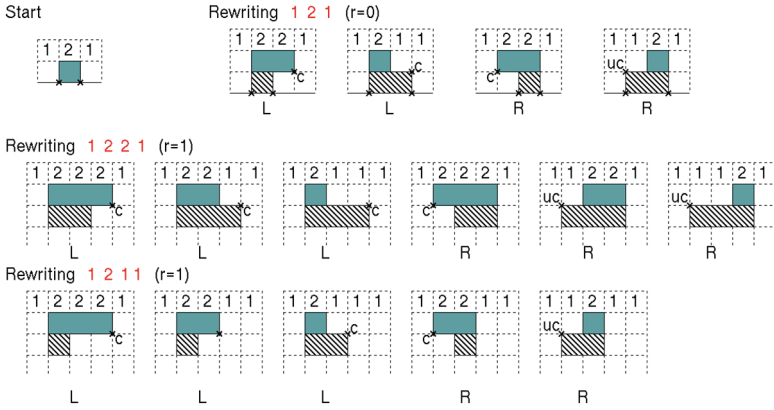


Fig. 10. Creating row-convex permutominoes. To focus on the distinguishing features, for polygons resulting from 1221 and 1211, only the two top rows are shown.

Each rule rewrites the key of a row-convex permutomino with $r - 1$ reflex vertices to the key of one of the row-convex permutominoes derived from it, having one more reflex vertex, for $r \geq 1$. The rewrite rules are:

$$\begin{aligned}
 1^a 2^b 1^c &\rightarrow_L 1^a 2^{b'} 1^{c'} \\
 1^c 2^b 1^a &\rightarrow_R 1^{c'} 2^{b'} 1^a
 \end{aligned}$$

for all $a, b, c \geq 1$, such that $a + b + c = r + 2$, and for all $b', c' \geq 1$ such that $a + b' + c' = r + 3$.

Let $B_{r,p,k}$ be the number of row-convex permutominoes with r reflex vertices and expansion key $1^p 2^{r+3-(p+k)} 1^k$, for $p, k \geq 1$, $p + k \leq r + 2$. By symmetry, we have $B_{r,p,k} = B_{r,k,p}$. For the recurrence, it interesting to aggregate further. Let $R_{r,p} = \sum_{k=1}^{r+1} B_{r,p,k}$ count the instances whose expansion key starts by 1^p . Then, $R_{0,1} = 1$ and, for $r \geq 1$, and we have

$$R_{r,p} = (r + 2 - p)R_{r-1,p} + \sum_{k=1}^{r+2-p} R_{r-1,k}$$

with $R_{r-1,r+1} = 0$. The first term results from the L -rule and the second from the R -rule (exploiting symmetry). The number of row-convex permutominoes with r vertices is $R(r) = \sum_{p=1}^{r+1} R_{r,p}$, for $r \geq 1$, with $R(0) = 1$. It is not difficult to see that

$$R(r) = 4R(r-1) + 2 \sum_{k=1}^{r-1} (r-k)R_{r-1,k}.$$

Again, we can use these recurrences to compute $R(r)$, by dynamic programming (with big integers), for small values of r . In [4], the authors conjecture that $R(r)$ can be defined asymptotically by $R(r) \sim k(r+2)!h^{r+1}$, with $k = 0.3419111$ and $h = 1.385933$, but the conjecture remains open.

6 Conclusion

In this paper we proposed a direct enumeration algorithm for generic permutominoes, based on the INFLATE-PASTE construction [23]. We developed tailored versions of the method to generate convex and row-convex permutominoes, from which we derived simple recurrences for counting these subclasses. It is worth noting that some of the constructions proposed by other authors for convex and row-convex permutominoes can be seen as instances of the INFLATE-PASTE method.

Acknowledgments. This paper is an extended version of the work presented at the XV Spanish Meeting on Computational Geometry (EGC 2013). The author would like to thank anonymous reviewers for insightful comments.

References

1. Auer, T., Held, M.: Heuristics for the generation of random polygons. In: Proceedings CCCG 1996, pp. 38–43 (1996)
2. Barcucci, E., Del Lungo, A., Pergola, E., Pinzani, R.: ECO: a methodology for the enumeration of combinatorial objects. *J. Differ. Equ. Appl.* **5**, 435–490 (1999)
3. Barequet, G., Moffie, M.: On the complexity of Jensen’s algorithm for counting fixed polyominoes. *J. Discrete Algorithms* **5**, 348–355 (2007)
4. Beaton, N., Disanto, F., Guttman, A.J., Rinaldi, S.: On the enumeration of column-convex permutominoes. In: Proceedings of FPSAC 2011, Iceland (2011)
5. Boldi, B., Lonati, V., Radicioni, R., Santini, M.: The number of convex permutominoes. *Inf. Comput.* **206**, 1074–1083 (2008)
6. Bousquet-Mélou, M.: Bijection of convex polyominoes and equations for enumerating them according to area. *Discrete Appl. Math.* **48**, 21–43 (1994)
7. Damian, M., Flatland, R., ORourke, J., Ramaswami, S.: Connecting polygonizations via stretches and twangs. *Theor. Comp. Syst.* **47**, 674–695 (2010)
8. Deutsch, E.: Enumerating symmetric directed convex polyominoes. *Discrete Math.* **280**, 225–231 (2004)
9. Disanto, F., Frosini, A., Pinzani, R., Rinaldi, S.: A closed formula for the number of convex permutominoes. *Electron. J. Combin.* **14**, R57 (2007)

10. Golomb, S.: *Polyominoes*. Princeton U. Press, Princeton (1994)
11. Grazzini, E., Pergola, E., Poneti, M.: On the exhaustive generation of convex permutominoes. *Pure Math. Appl.* **19**, 93–104 (2008)
12. Hickerson, D.: Counting horizontally convex polyominoes. *J. Integer Sequences* **2**, Article 99.1.8 (1999)
13. Insitti, F.: Permutation diagrams, fixed points and Kazhdan-Lusztig R-polynomials. *Ann. Comb.* **10**, 369–387 (2006)
14. Jensen, I.: Enumerations of lattice animals and trees. *J. Stat. Phys.* **102**, 865–881 (2001)
15. Jensen, I.: Counting polyominoes: a parallel implementation for cluster computing. In: Sloot, P.M.A., Abramson, D., Bogdanov, A.V., Gorbachev, Y.E., Dongarra, J., Zomaya, A.Y. (eds.) *ICCS 2003, Part III*. LNCS, vol. 2659, pp. 203–212. Springer, Heidelberg (2003)
16. Del Lungo, A., Duchi, E., Frosini, A., Rinaldi, S.: On the generation and enumeration of some classes of convex polyominoes. *Electron. J. Combin.* **11**, R60 (2004)
17. Martins, A.M., Bajuelos, A.: Vertex guards in a subclass of orthogonal polygons. *Int. J. Comput. Sci. Netw. Secur.* **6**, 102–108 (2006)
18. Overmars, M., Wood, D.: On rectangular visibility. *J. Algorithms* **9**(3), 372–390 (1998)
19. O’Rourke, J.: An alternate proof of the rectilinear art gallery theorem. *J. Geom.* **21**, 118–130 (1983)
20. Rinaldi, S., Socci, S.: About half permutations. *Electr. J. Comb.* **21**(1), P1.35 (2014)
21. Sloane, N.J.A.: The On-Line encyclopedia of integer sequences. OEIS Foundation. <http://oeis.org/>
22. Sohler, C.: Generating random star-shaped polygons. In: *Proceedings CCCG 1999* (1999)
23. Tomás, A.P., Bajuelos, A.: Quadratic-time linear-space algorithms for generating orthogonal polygons with a given number of vertices. In: Laganá, A., Gavriloa, M.L., Kumar, V., Mun, Y., Tan, C.J.K., Gervasi, O. (eds.) *ICCSA 2004*. LNCS, vol. 3045, pp. 117–126. Springer, Heidelberg (2004). doi:[10.1007/978-3-540-24767-8_13](https://doi.org/10.1007/978-3-540-24767-8_13)
24. Zhu, C., Sundaram, G., Snoeyink, J., Mitchell, J.S.B.: Generating random polygons with given vertices. *Comput. Geom.* **6**, 277–290 (1996)