

On the enumeration of permutominoes

Ana Paula Tomás*

DCC & CMUP, Faculdade de Ciências
Universidade do Porto, Portugal

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* [12] by size, or, equivalently, for the enumeration of *grid orthogonal polygons* [19]. We show how the construction technique allows us to derive a simple characterization of the class of convex permutominoes, which has been extensively investigated recently [4]. The approach extends to some of its subclasses, namely to the row convex and the directed convex permutominoes.

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, 6] or restrict to certain classes of polygons, e.g., monotone, convex or star-shaped polygons [18, 20]. Numerous related problems have also been extensively investigated, as the exact counting or enumeration of polyominoes [9]. 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* (see Fig. 1). 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 [13] – an indirect

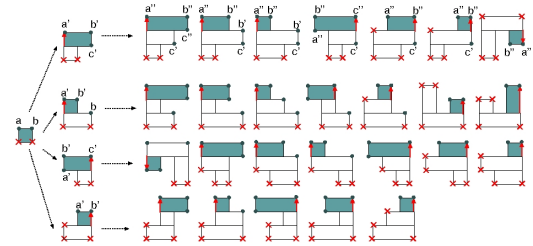


Figure 1: Permutominoes of size 1, 2, and 3 (i.e., with 4, 6, and 8 vertices) and their horizontal partitions.

method is currently the most powerful algorithm for counting fixed polyominoes. Exact counts are known for polyominoes that have up to 56 cells [3, 14]. As far as we can see, Jensen’s algorithm cannot be easily adapted for counting permutominoes.

1 Background

A polygon is called orthogonal if its edges meet at right angles. If r is the number of reflex vertices of an n -vertex orthogonal polygon, then $n = 2r + 4$ (e.g. [16]). Grid orthogonal polygons (*grid ogons*) were introduced in [19] as a relevant class for generation. A *grid ogon* is an orthogonal polygon without collinear edges, embedded in a regular square grid, and having exactly one edge in each line of its minimal bounding square. They were addressed more recently under the name of *permutominoes* [4, 8, 12], because they correspond to polyominoes that are determined by a suitable pair of permutations having the same size. All polyominoes we will consider are simply-connected and, similarly, all polygons are simple and without holes. A permutomino that is given by permutations of $\{1, 2, \dots, r + 2\}$, for $r \geq 0$, is said to have size $r + 1$. The size is the width of its 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 $n = 2r + 4$ vertices in total.

In this paper, we give an algorithm for the enumeration of all permutominoes by size, based on the INFLATE-PASTE¹ technique introduced in [19]. Every

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

*Email: apt@dcc.fc.up.pt. Research partially supported by the European Regional Development Fund through the programme COMPETE and by the Portuguese Government through the FCT – Fundação para a Ciência e Tecnologia under the project PEst C/MAT/UI0144/2011, and project JEDI (PTDC/EIA/66924/2006).

grid n -gon P results from a unit square by applying INFLATE-PASTE r times, where $r = (n - 4)/2$ is the number of reflex vertices of P . 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, say v , must be contained in a region that we called *the free neighbourhood of v* (Fig. 2), and is fixed to v . This region, denoted by $FSN(v)$, consists of the

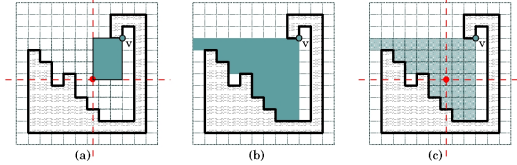


Figure 2: INFLATE-PASTE: (a) gluing a rectangle to v (b) $FSN(v)$ is the shaded region (c) the rectangle is defined by v and the center of a cell of $FSN(v)$.

external points that are rectangularly visible from v and belong to the quadrant with origin v that contains the horizontal edge incident to v (two points a and b are *rectangularly visible* if the axis-aligned rectangle that has a and b as opposite corners does not intersect the interior of the polygon). The INFLATE operation keeps the grid regular: the grid lines are shifted, if needed, to insert two new horizontal and vertical lines for the new edges (Fig. 3).

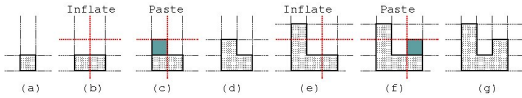


Figure 3: Two applications of INFLATE-PASTE: the center of the *inflated cell* becomes a convex vertex of the polygon created at each step.

In [19], $FSN(v)$ was called the free staircase neighborhood of v . Actually, $FSN(v)$ is a Ferrers diagram, with origin at v , and hence it can be given by a sequence of integers, each integer representing the number of cells that form each row of the diagram. Any cell in $FSN(v)$ could be used for growing the polygon using v . Therefore, for the example given in Fig. 2, we could produce $9 + 7 + 6 + 4 + 4 + 3 + 2 = 35$ distinct grid ogons using the selected vertex.

2 Direct Enumeration

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 grid ogons (i.e., permutominoes) using INFLATE-PASTE. It is based on the existence of a unique depth-first generating tree for each n -gon, once we fix the order for visiting the horizontal partition. One possibility is to define it as the order induced by a clockwise walk around the polygon, starting at its southwest vertex, as in Fig. 4. The bottom

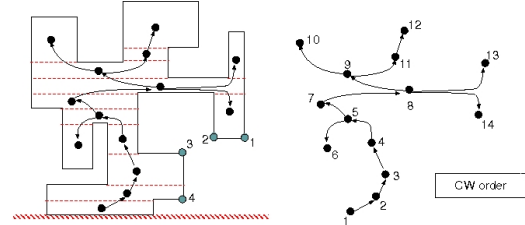


Figure 4: The unique generating tree for the represented grid ogon. Vertices 1, 2, 3 and 4 in the polygon are the ones that could still be expanded in our enumeration method, and should be visited in this order.

line (i.e., the horizontal line that contains the SW-vertex) is never moved upwards, but polygons can move freely along it. Fig. 1 shows how permutominoes can be generated by our method. The vertices marked with a cross would be no longer available for expansion.

```

PERMUTOMINOENUM( $P, S, G, n_0, r$ )
  if  $r = 0$  then return fi
  MAKEEMPTY( $TrS$ )
  while not ISEMPTY( $S$ ) do
     $v := \text{POP}(S)$  /*  $v$  is  $(v_x, v_y)$  */
     $e_H(v) :=$  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( $p, q, G$ )
         $w_1 := (p + 1, v_y)$ ;  $w_2 := (p + 1, q + 1)$ ;
         $w_3 := (v_x, q + 1)$ 
        PASTERECTANGLE( $v, [w_1, w_2, w_3], P$ )
        if  $w_1 \in \text{INTERIOR}(e_H(v))$  then
          PUSH( $\{w_2, w_3\}, S, P$ )
        else PUSH( $\{w_1, w_2, w_3\}, S, P$ ) fi
        OUTPUTPOLYGON( $P, n_0 + 2$ )
        PERMUTOMINOENUM( $P, S, G, n_0 + 2, r - 1$ )
        CUTRECTANGLE( $v, [w_1, w_2, w_3], P$ )
        DEFLATEGRID( $p + 1, q + 1, G$ )
      done
    fi
    PUSH( $v, TrS$ );
  done
  while not ISEMPTY( $TrS$ ) do
    PUSH( $\text{POP}(TrS), S$ )
  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 and count all grid ogons that have up to $2r + 4$ vertices.

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).

An algorithm for enumerating the *convex* permutominoes by size was published recently [10]. 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.

3 Convex Permutominoes

Although the exact counting and enumeration of polyominoes remain challenging open problems, several positive results were achieved for special classes of polyominoes [5, 7, 15], 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 [11]. 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 said to be *convex* if it is both row-convex and column-convex. 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 [5]. The corresponding classes of permutominoes have been addressed recently [4, 8].

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. Fig. 5 shows all n -vertex convex permutominoes for $n = 4, 6, 8$, each one embedded on a grid. Only the two topmost convex vertices may 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

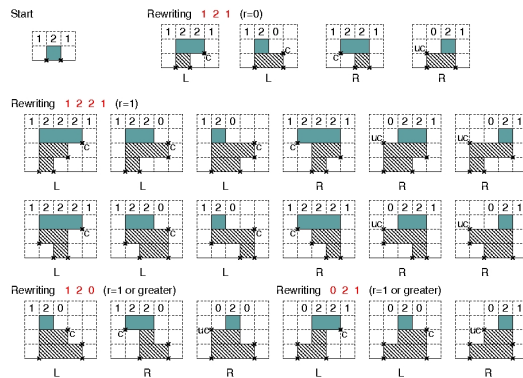


Figure 5: Enumerating convex permutominoes by size.

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. 5, each counter is in its cell). Here, *see* means that the cell belongs to the free neighborhood of the vertex. 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 PERMUTAMINOENUM. 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 PERMUTAMINOENUM, we may conclude that the *expansion key* of any convex permutomino with $r \geq 0$ reflex vertices must be of one of the following forms:

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

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:

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

$$\begin{aligned}
 02^{j'}1 &\xrightarrow{R} 02^j1, & \text{for } 1 \leq j \leq j' \leq r-1 \\
 02^{j'}1 &\xrightarrow{L} 02^{j'+1}1, & \text{for } 1 \leq j' \leq r-1 \\
 02^{j'}1 &\xrightarrow{L} 02^j0, & \text{for } 1 \leq j \leq j' \leq r-1 \\
 02^{j'}0 &\xrightarrow{L,R} 02^j0, & \text{for } 1 \leq j \leq j' \leq r-2
 \end{aligned}$$

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. Figs. 6–8 illustrate the idea underlying these rules.



Figure 6: Rewriting $12^r 1$ using the rewrite rules $12^r 1 \xrightarrow{L} 12^{r+1} 1$ and $12^r 1 \xrightarrow{R} 12^{r+1} 1$.

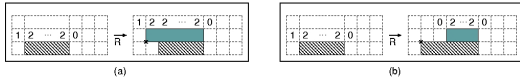


Figure 7: Rewriting $12^{j'} 0$ using (a) $12^{j'} 0 \xrightarrow{L} 12^{j'+1} 0$ and (b) a rule $12^{j'} 0 \xrightarrow{R} 02^j 0$, for $1 \leq j \leq j'$.

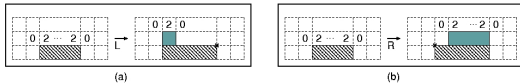


Figure 8: Rewriting $02^{j'} 0$ using (a) $02^{j'} 0 \xrightarrow{L} 02^j 0$ and (b) $02^{j'} 0 \xrightarrow{R} 02^j 0$, for some $1 \leq j \leq j'$.

The correctness and completeness of this rewrite system can be checked easily by case-analysis, taking into account the conditions on convexity.

Proposition 1 Let $A_{\alpha,j,\beta}^{(r)}$ 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, $A_{0,j,1}^{(r)} = A_{1,j,0}^{(r)}$, for all r and j (symmetry by reflection w.r.t. V -axis) and $A_{\alpha,j,\beta}^{(r)}$ is inductively defined as follows.

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

The number of convex permutominoes of size n is given by sequence A126020, in [17]. A closed formula for this number is given in [8]. In a similar way, we may deduce the recurrences for row-convex permutominoes and the simpler classes of bargraphs, stacks, and Ferrers diagrams.

References

- [1] T. Auer and M. Held, Heuristics for the generation of random polygons, in: *Proc. CCCG'96*, 1996, 38–43.
- [2] E. Barucci, A. Del Lungo, E. Pergola, and R. Pinzani, ECO: a methodology for the enumeration of combinatorial objects, *J. Differ. Equ. Appl.* **5** (1999), 435–490.
- [3] G. Barequet and M. Moffie, On the complexity of Jensen’s algorithm for counting fixed polyominoes, *J. Discrete Algorithms* **5** (2007), 348–355.
- [4] P. Boldi, V. Lonati, R. Radicioni, and M. Santini, The number of convex permutominoes, *Information and Computation* **206** (2008), 1074–1083.
- [5] M. Bousquet Mélou, Bijection of convex polyominoes and equations for enumerating them according to area, *Discrete Appl. Math.* **48** (1994), 21–43.
- [6] M. Damian, R. Flatland, J. O’Rourke, and S. Ramaswami, Connecting polygonizations via stretches and twangs, *Theory of Computing Systems* **47** (2010), 674–695.
- [7] E. Deutsch, Enumerating symmetric directed convex polyominoes, *Discrete Math.* **280** (2004), 225–231.
- [8] F. Disanto, A. Frosini, R. Pinzani, and S. Rinaldi, A closed formula for the number of convex permutominoes, *Electron. J. Combin.* **14** (2007), #R57.
- [9] S. Golomb, *Polyominoes*, Princeton U. Press, 1994.
- [10] E. Grazzini, E. Pergola, and M. Poneti, On the exhaustive generation of convex permutominoes, *Pure Mathematics and Applications* **19** (2008), 93–104.
- [11] D. Hickerson, Counting horizontally convex polyominoes, *J. Integer Sequences* **2** (1999), Article 99.1.8.
- [12] F. Insitti, Permutation diagrams, fixed points and Kazhdan Lusztig R polynomials, *Annals of Combinatorics* **10** (2006), 369–387.
- [13] I. Jensen, Enumerations of lattice animals and trees, *J. Stat. Phys.* **102** (2001), 865–881.
- [14] I. Jensen, Counting polyominoes: a parallel implementation for cluster computing, in: *Proc. ICCS'03*, LNCS, vol. 2659, Springer, 2003, 203–212.
- [15] A. Del Lungo, E. Duchi, A. Frosini, and S. Rinaldi, On the generation and enumeration of some classes of convex polyominoes, *Electron. J. Combin.* **11** (2004), #R60.
- [16] J. O’Rourke, An alternate proof of the rectilinear art gallery theorem, *J. Geometry* **21** (1983), 118–130.
- [17] N. J. A. Sloane, The On Line encyclopedia of integer sequences, OEIS Foundation, <http://oeis.org/>.
- [18] C. Sohler, Generating random star shaped polygons, in: *Proc. CCCG'99*, 1999.
- [19] A. P. Tomás and A. Bajuelos, Quadratic time linear space algorithms for generating orthogonal polygons with a given number of vertices, in: *Proc. CGA'04–ICCSA'04*, LNCS, vol. 3045, Springer, 2004, 117–126.
- [20] C. Zhu, G. Sundaram, J. Snoeyink, and J. S. B. Mitchell, Generating random polygons with given vertices, *Comput. Geom.* **6** (1996), 277–290.