POLYLAB
A package for the study of polyominoes

Francesco Bertoli, Alberto Del Lungo, Francesco Ulivi

Dipartimento di Sistemi e Informatica,
Via Lombroso 6/17, 50134 Firenze, Italy,
e-mail: alberto@cellini.ing.unifi.it

Abstract

We give a brief description of the main features of the mathematical package POLYLAB. POLYLAB randomly generates several types of polyominoes, it draws pictures of them and performs statistics regarding some parameters. Moreover, we represent some bijections between polyominoes and some other combinatorial objects graphically.

Résumé

Nous donnons une description succinte des applications du logiciel POLYLAB. POLYLAB est un logiciel pour la génération aléatoire de plusieurs classes de polyominos. Il réalise les dessins des polyominos générés de façon aléatoire et il effectue des statistiques sur leur principals paramètres. En plus, POLYLAB illustre quelques bijections entre les polyominos et d'autres objects combinatoires.

1 General aim

Our aim is to create a laboratory for polyominoes, able to give us a lot of experimental results about them. In order to do this, we collected some algorithms regarding the random generation of polyominoes and implemented these algorithms in a single package, called POLYLAB. This software product is completely written in the C language so that we obtain a program that can be used on any kind of computer. It has user-friendly graphic interface (see fig. 1). POLYLAB has two main features:

1. It randomly generates a single polyomino belonging to one of the classes implemented in POLYLAB. It gives us a drawing of it and a table of its significant parameter values.

2. It produces some statistics regarding generated polyominoes by generating a polyomino of a given class with respect to one of its parameters as often as the user wants. It also furnishes us with the average values of the other parameters which can be illustrated graphically.

The first feature allows us to check some conjectures about the properties of a given class of polyominoes and verify the bijections or injections between different classes of polyominoes and other combinatorial objects in an experimental way.
The second feature gives us a double advantage: it both allows us to check (or reject) the conjectures and gives us some information about the parameters we have no theoretical results for.

2 Classes of implemented polyominoes

In this section, we give a brief description of the algorithms collected.

- *Directed Polyominoes* (rgda.exe)
  We perform three steps to obtain them: 1) We generate a Motzkin left factor. These words are in bijection with a particular structure called skew tree [14]. 2) We create a skew tree. 3) We obtain a directed polyomino from the skew tree. This construction generates a directed polyomino with respect to its area and works in linear time (see [6]). We then use this algorithm to check the Dhar's result [11], that is: *the average height of directed polyominoes with area n is asymptotically:*

  $$h_n \sim \alpha\, n^{\beta},$$

  with $1.23 \le \alpha \le 1.26$, $0.816 \le \beta \le 0.821$. Moreover, we make the following conjecture:

  Conjecture 2.1 *The average bond perimeter of directed polyominoes with area n is asymptotically:*

  $$P_n \sim \sqrt{\pi}\, n.$$

- *Convex Polyominoes* (cvxrc.exe, cvxper.exe)
  The first algorithm generates a convex polyomino having perimeter n and does this with an asymptotic probability of 0.5 (see [12]). We choose a number in $[0..n/2]$ and generate a V-H string of length n. Then we check to see if the V-H sequence maps a convex polyomino: if it does, we draw it; otherwise, we repeat the process.
  The second algorithm draws a polyomino with respect to its number of rows and columns and is obtained by slightly changing the first algorithm. The second algorithm is used for testing the reconstruction of convex polyominoes by starting out from their projections (see [4]).

- *Directed Column-Convex Polyominoes* (dcc.exe)
  By using a recurrence relation for the number of dcc-polyominoes having area n and k cells in their first column, we obtain a very simple linear-time random generation algorithm for this kind of polyomino (see [5]). We use this algorithm to check the following result obtained by Barcucci *et al* [2]:

  Theorem 2.2 *The average height of dcc-polyominoes having area n is:*

  $$h_n \sim \frac{\phi}{\sqrt{5}}\, n + 0.8765.$$

- *"Deco" Polyominoes* (deco.exe)
  The deco polyominoes are directed column-convex polyominoes having height k and which reach the height k only in their last column. We use the bijection between deco polyominoes and the set of permutations of the first k integers to generate the polyominoes of this class (see [3]). Moreover, we examine the deco polyominoes having triangular and hexagonal cells and we generate them in linear time by means of algorithms illustrated in [3].

- *Stair Polyominoes* (escal.exe)
  Dyck words are in bijection with stair polyominoes (see [9, 10]). By using the *Random Bracket Sequence* algorithm (see [1]), we create a Dyck word and subsequently obtain the corrisponding stair polyomino.

- *Parallelogram Polyominoes* (paral.exe)
  In this algorithm, we use the Delest-Viennot bijection between Dyck paths and parallelogram polyominoes (see [8]). The *Random Bracket Sequence* algorithm is used to generate a Dyck path which we use to obtain the corrisponding parallelogram polyomino.

- *Directed Convex Polyominoes* (cvxdir.exe)
  A Grand Dyck word is a sequence of $\{x, y\}^*$ in which the number of x is equal to the number of y. These words of length 2n are in bijection with directed column-convex polyominoes having perimeter 2n+4 (see [7]). We generate a Grand Dyck word by using the *Selection Sampling Tecnique* algorithm (see [1]) and we get the corresponding directed convex polyomino.

- *Wall Polyominoes* (wall.exe)
  A wall polyomino is a column-convex polyomino whose columns are joined so that their lower borders are aligned. By generating a sequence of random numbers, we draw the wall polyominoes with respect to their rows and columns.

By means of the previous algorithms, we develop some modules that show some bijections between various classes of polyominoes and some other combinatorial objects. For example, we illustrate the bijections between stair polyominoes, parallelogram polyominoes, Dyck words (see fig. 2), plane trees, binary trees and 2-coloured Dyck words.

3 Conclusions

Each algorithm is implemented in a stand-alone runnig module. All the modules are managed by POLYLAB, a supervisor program that makes modules run and shows, saves, loads and prints results in a graphic desktop. It therefore allows us to create an open package able to accept any other algorithms without modifying any line of its program's code. Presently, we are developing some new modules.

4 Hardware requirements

Polylab is a package that runs on any Personal Computer supporting DOS with a VGA adapter.

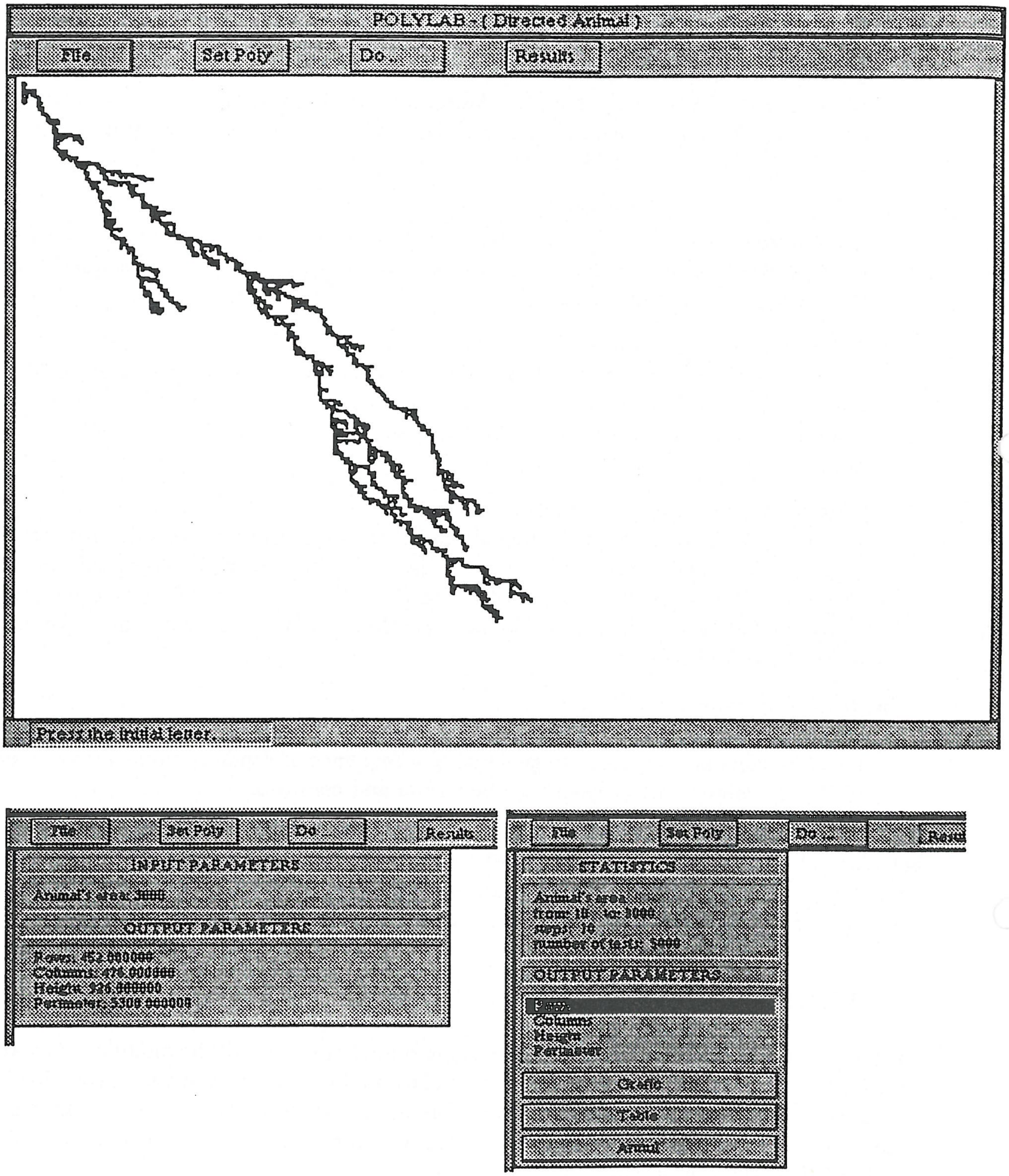

Figure 1: A directed polyomino

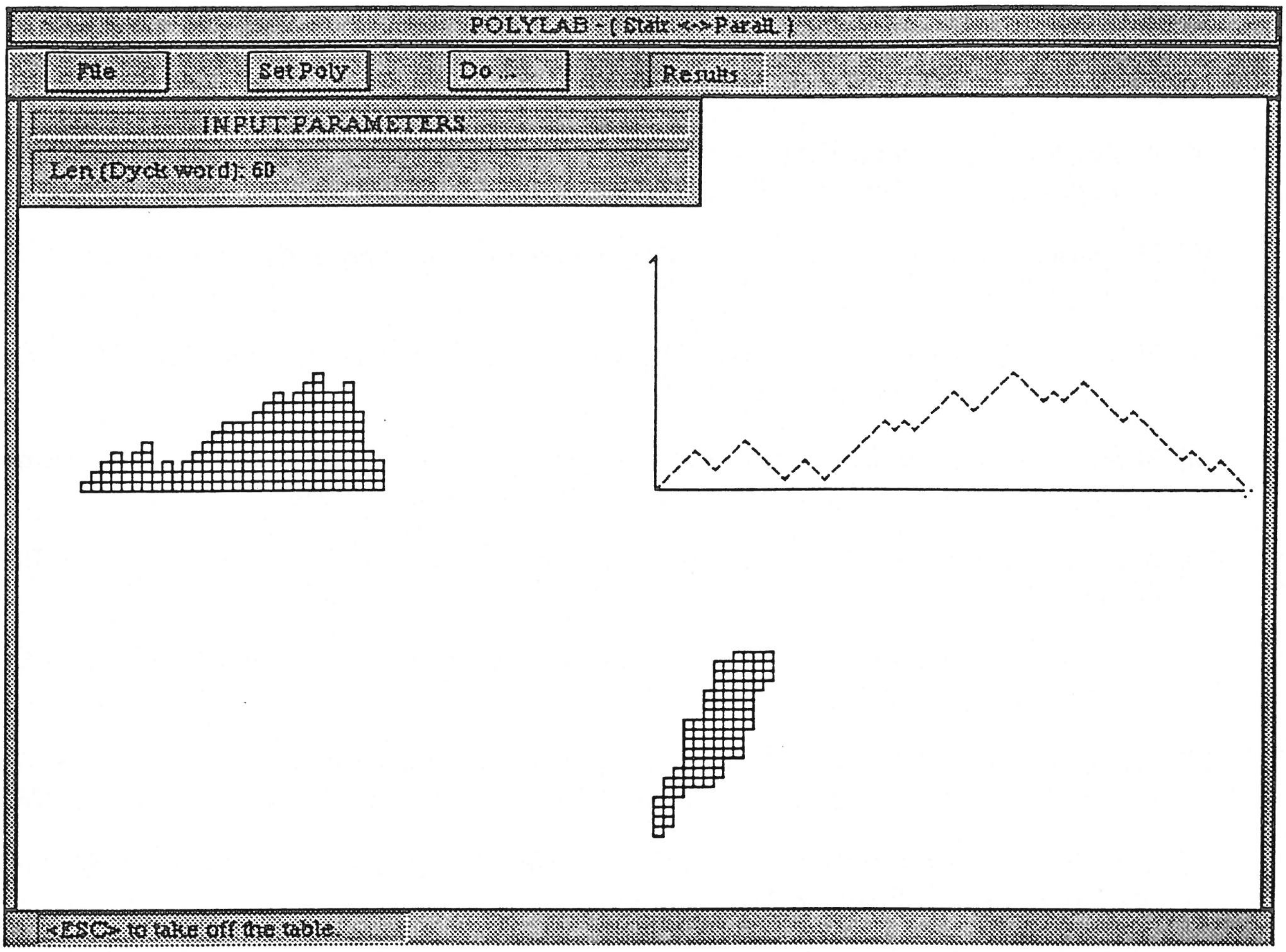

Figure 2: A bijection between combinatorial objects

References

[1] M.D.Atkinson, J.R.Sack, Generating binary trees at random, *Information Processing Letters* 41 (1992), 21-23.

[2] E. Barcucci, A. Del Lungo, R. Pinzani and R. Sprugnoli, La hauteur des polyominos dirigés verticalement convexes, *in Actes* 31ème *Séminaire Lotharingien de Combinatoire*, Université Strasbourg I (1993).

[3] E.Barcucci, A.Del Lungo, R.Pinzani, "Deco" polyominoes, permutations and random generation, *technical report DSI, 16/93*, Università di Firenze (1994).

[4] E. Barcucci, A. Del Lungo, M. Nivat and R. Pinzani, Reconstructing convex polyominoes from their horizontal and vertical projections, *technical report DSI, RT 12/94*, Università di Firenze (1994).

[5] E.Barcucci, R.Pinzani, R.Sprugnoli, Directed Column-Convex Polyominoes by Recurrence Relations, TAPSOFT'93, eds. G.Goos and J.Hartmanis, *Lecture Notes in Computer Science*, 668 (1993) 282-298.

[6] E.Barcucci, R.Pinzani, R.Sprugnoli, The random generation of directed animals, *Theoretical Computer Science*, 127 (1994) 333-350.

[7] M.Bousquet-Mélou, q-énumeration de polyominos convexes, *Publication du LaCIM 9*, Montreal (1991).

[8] M.Delest, X.G.Viennot, Algebraic languages and polyominoes enumeration, *Theoretical Computer Science* 34 (1984) 169-206.

[9] M.Delest, S.Dulucq, Enumeration of directed column convex animal with given perimeters and area, *Report LaBRI 86-15*, Université Bordeaux I (1986).

[10] A.Denise, Etude des q-grammaires, *Mémoire de DEA Informatique*, Université Bordeaux I (1991).

[11] D. Dhar, Longitudinal size exponent for two dimensional directed animals, *J. Phys. A:Math. Gen.*, 21 (1988) L893-L897.

[12] W.Hochstutter, M.Loebl, C.Moll, Generating convex polyominoes at random, *Proceeding of the 5th FPSAC*, eds. A.Barlotti, M.Delest, R.Pinzani, Firenze (1993), 277-278.

[13] D.E. Knuth, *Semi-numerical Algorithms, The art of computer programming*, Vol.2 (Addision-Wesley, Reading, MA, 2nd ed. 1981).

[14] J.G. Penaud, Une novelle bijection pour les animaux dirigés, *Actes du 22ème Séminaire Lotharingien de Combinatorire*, Université Strasbourg I-IRMA.