

ALGEBRAIC LANGUAGES AND POLYOMINOES ENUMERATION

Marie-Pierre DELEST

Département d'Informatique, Université Bordeaux II, 33076 Bordeaux Cedex, France

Gérard VIENNOT

U.E.R. Mathématiques et Informatique, Université Bordeaux I, 33405 Talence, France

A notre 'bon maître' M.P. Schützenberger

Abstract. In this paper, the use of algebraic languages theory in solving an open problem in combinatorics is shown. By constructing a bijection between convex polyominoes and words of an algebraic language, and by solving the corresponding algebraic system, we prove that the number of convex polyominoes with perimeter $2n + 8$ is $(2n + 1)4^n - 4(2n + 1)\binom{2n}{n}$.

1. Introduction

Let A_n be a class of combinatorial objects enumerated by the integer a_n and suppose that the corresponding generating function $f(t) = \sum_{n \geq 0} a_n t^n$ is *algebraic*. An old idea, dear to M.P. Schützenberger, is to explain this algebraicity by expliciting a bijection between A_n and the words of a certain *algebraic (context-free) language* L defined on the alphabet X by a *non-ambiguous grammar*.

Classically, from the non-ambiguous grammar, one can associate a proper algebraic system of equations in *noncommutative power series*. The unique solution of the system contains the (noncommutative) generating function $L = \sum_{w \in L} w$ of the language L . By sending all variables x of X onto one variable t , the series L becomes $f(t) = \sum_{n \geq 0} a_n t^n$, solution of an algebraic system in one variable t (see Schützenberger [34, 35]).

Usually an explicit formula is known for a_n or $f(t)$ by means of classical calculus techniques used in combinatorics (recurrence relation, Lagrange inversion formula, etc.). The coding with words sheds more light upon the combinatorial comprehension of A_n . Each equation of the noncommutative algebraic system is in fact a combinatorial property of the objects of A_n . The coding with words appears to be a nice intermediate between the combinatorial objects themselves and the generating function in one variable $f(t)$.

Classical examples are those used with enumeration of trees or related objects and can be found in [15, 17, 23, 24]. Deep-going examples are found in the work of Cori and Vauquelin [6, 8], following the numerous formulae enumerating planar maps obtained by Tutte et al. (see, for example, [37]).

In this paper, the method is reversed: no formula is known for a_n or $f(t)$, nor the fact that $f(t)$ is algebraic. We use the algebraic language methodology to prove this fact and obtain an explicit formula. Here, A_n is the set of convex polyominoes with perimeter n .

Unit squares having their vertices at integer points in the Cartesian plane are called *cells*. A *polyomino* is a connected subset of the plane which is a finite union of cells and has no cut set, that is, the interior is also connected. The number of cells is the *area* of the polyomino, the length of the border is the *perimeter*. Polyominoes are defined up to translation. Note that, in the enumeration considered here, symmetries or rotations are forbidden.

Polyominoes are classical objects in combinatorics and have been popularized by Gardner and Golomb [16]. Except for some special class of polyominoes, very few exact formulae are known. Enumeration of (general) polyominoes is a major unsolved problem, also called the *cell growth problem*. Since Read [31], polyominoes are also called (fixed) *animals*.

A huge number of asymptotic results has been given by physicists for whom such objects are important in statistical mechanics (they call *animal* the set of points obtained by taking the center of each cell of a polyomino).

A polyomino P is said to be *column-* (respectively *row-*) convex if the intersection of P with any vertical (respectively horizontal) line is a connected segment. A *convex polyomino* is a polyomino which is both column- and row-convex (see Fig. 1).

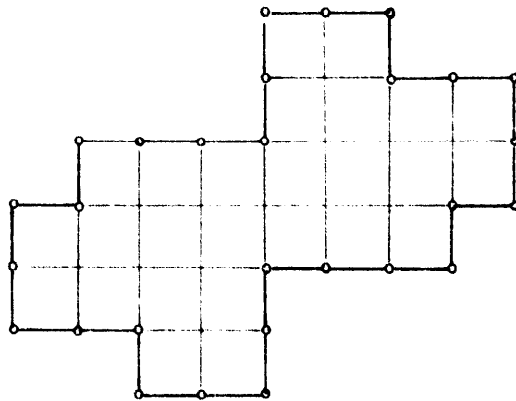


Fig. 1. A convex polyomino.

Klarner [19] gave an explicit expression for the generating function of row-convex polyominoes enumerated according to the area. The generating function is rational and is obtained by using a combinatorial interpretation of Fredholm integral operation [20].

Knuth raised the problem [21] to give some information about the number of convex polyominoes. Klarner and Rivest [21] and Bender [1] gave asymptotic estimates for the number a_n of convex polyominoes having area n . More precisely, $a_n \sim cy^n$ with $\gamma = 2.30914\dots$ and $c = 2.67564\dots$.

We give here an exact formula for the number p_{2n} of convex polyominoes having a perimeter $2n$. Surprisingly the result is very simple.

Theorem 1.1. *The number p_{2n} of convex polyominoes having a perimeter $2n$ is*

$$p_4 = 1, \quad p_6 = 2$$

and for $n \geq 0$,

$$p_{2n+8} = (2n + 1)4^n - 4(2n + 1)\binom{2n}{n}.$$

The method makes use of three steps:

Step (i) Bijection between convex polyominoes and words of an algebraic language. In fact, three different types of polyominoes, and thus three languages, have to be considered.

Step (ii) Solving the three corresponding algebraic systems and obtaining the generating function $p(t) = \sum_{n=0}^{\infty} p_{2n}t^{2n}$. These systems have about 20 to 40 equations each. We are thus led to employ carefully the algebraic language methodology by using auxiliary algebraic languages and some substitution operators. Also, multi-head finite automata can be used in order to encode convex polyominoes of the second type with words of a *rational language* (accepted by a *finite automata*), and thus reduce the computation of the corresponding generating function to a determinant calculus (in our case the matrix has size 16×16). The final solution (especially for polyominoes of the third type) has been made possible using the symbolic manipulation system MACSYMA from MIT.

Step (iii) Expanding the generating function $p(t)$ in order to obtain the formula for p_{2n} .

Remark 1.2. To the knowledge of the authors, no 'classical' proof of Theorem 1.1 has been found yet.

Remark 1.3. The concept of convex polyominoes appears in some algorithmic problems related to *integrated circuit manufacture*. A layer of an integrated circuit is printed on a photographic plate by flashing rectangles and produce an image equal to their superposition. The plate will become a photographic mask in the manufacture of integrated circuit. The image is a (union of) polyominoes. Neglecting some additional technical constraints, the problem is to produce the image using as few rectangles as possible. Masek [26] proved that finding the minimum number of rectangles is NP-complete. Chaiken et al. [9] established a beautiful min-max property about this number in the case of convex polyominoes, for which a polynomial time algorithm can be deduced. Then, Berge et al. [2] looked for possible extensions to vertically convex polyominoes and the so-called 'pataconvex' polyominoes.

For another example of relationships between polyominoes, and VLSI and non-conventional architectures, see Van Leeuwen [38].

This paper has been made self-contained for both ‘combinatorists’ and ‘theoretical computer scientists’.

We recall a few geometric notations used in all this paper.

Notations. The plane $\mathbb{Z} \times \mathbb{Z}$ is denoted by Π . A path ω is a sequence $\omega = (s_0, s_1, \dots, s_n)$ of points of Π . The point s_0 (respectively s_n) is the *starting* (respectively *final*) point. The *length* of ω is the integer n . Each pair (s_i, s_{i+1}) is an *elementary step* of the path. The elementary step (s_i, s_{i+1}) is called *North* (*South*, *East*, *West* respectively) iff $s_i = (x, y)$, $s_{i+1} = (x', y')$ with $x = x'$, $y' = y + 1$ ($x = x'$, $y' = y - 1$; $x' = x + 1$, $y = y'$; $x' = x - 1$, $y = y'$ respectively).

2. Algebraic languages (for ‘pure’ combinatorists)

This section has been introduced for the combinatorists not familiar with the classical concept (in Theoretical Computer Science) of algebraic language. We will not give complete formal definitions for every notion we use, but we will, with examples, give a brief outline of what is necessary. For more details, see the works of Berstel [3], Ginsburgh [14] or Salomaa and Soittola [33].

Notations. Let X be a finite nonempty set called *alphabet*. We denote by X^* the free *monoid* generated by X , that is, the set of words written with letters from X , together with the product defined as the *concatenation* of two words: for $u = u_1 \dots u_p$ and $v = v_1 \dots v_q$, we have $uv = u_1 \dots u_p v_1 \dots v_q$.

The *empty word* is denoted by e . The number of occurrences of the letter x in the word w is denoted by $|w|_x$ and the length of w by $|w| = \sum_{x \in X} |w|_x$. If the word w can be factorized as $w = uvv$, we say that the word u (f , v respectively) is a *left factor* (*factor*, *right factor* respectively) of w .

Let \mathbb{K} be a unitary commutative ring (in fact \mathbb{K} will be \mathbb{Z} or \mathbb{Q}). We denote by $\mathbb{K}\langle\langle X \rangle\rangle$ (respectively $\mathbb{K}[[X]]$) the algebra of noncommutative (respectively commutative) power series with variables from X and coefficients in \mathbb{K} . By commuting variables, one obtains a canonical morphism $\alpha : \mathbb{K}\langle\langle X \rangle\rangle \rightarrow \mathbb{K}[[X]]$.

A *language* is nothing but a subset of X^* . For any language L we associate the generating function $L = \sum_{w \in L} w$, element of $\mathbb{Z}\langle\langle X \rangle\rangle$. Note that for every language A , B , C , the equality $C = A \cdot B$ means that $C = AB$ and that any word w of C has a unique factorization $w = uv$ with $u \in A$, $v \in B$. Also the equality $C = A + B$ means $C = A \cup B$ and $A \cap B = \emptyset$.

An *algebraic* (also called *context-free*) grammar is a 4-tuple $G = (N, X, P, s)$ where N and X are finite disjoint sets, s is an element of N and P is a finite set of pairs (α, β) with $\alpha \in N$ and $\beta \in (N \cup X)^*$. Such a pair is called a *production* and also denoted by $\alpha \rightarrow \beta$.

Starting from s , one defines a set of words of $(N \cup X)^*$ by applying recursively substitutions of the form $u\alpha v \rightarrow u\beta v$ with $(\alpha, \beta) \in P$. The subset of words of X^* is called the language generated by G and denoted by $L(G)$. Such a language is called *algebraic* (or *context-free*). We give some (very elementary) examples that will be used in this paper.

Example 2.1 (Dyck word). Let $G = (N, X, P, s)$ with $N = \{D\}$, $X = \{x, \bar{x}\}$, $s = D$ and P is given by the two productions $D \rightarrow xD\bar{x}D$, $D \rightarrow e$. The language $L(G)$ is the classical 'restricted Dyck language' on two letters. Throughout this paper, this language will be called *Dyck language* for short and will be denoted by D . The words w of D , or *Dyck words*, are characterized by the two following conditions:

- (1) for any left factor u of w , $|u|_x \geq |u|_{\bar{x}}$;
- (2) $|w|_x = |w|_{\bar{x}}$.

Example 2.2 (Motzkin word). The words w of $\{x, \bar{x}, b\}^*$ (respectively $\{x, \bar{x}, b, r\}^*$) satisfying the above conditions (1) and (2) are called *Motzkin words* (respectively *2-colored Motzkin words*). The Motzkin words are generated by the following grammar:

$$N = \{M\}, \quad s = M, \quad X = \{x, \bar{x}, b\},$$

$$M \rightarrow xM\bar{x}M, \quad M \rightarrow bM, \quad M \rightarrow e.$$

By adding the rule $M \rightarrow rM$, one obviously obtains a grammar for the 2-colored Motzkin words.

Example 2.3 (Fibonacci word). Let FB be the set of words of $\{x, a\}^*$ which can be factorized as a product of words reduced to "a" or xx , and having an even length. Such words are generated by the algebraic grammar:

$$N = \{FB, G\}, \quad s = FB, \quad X = \{x, a\}$$

and productions

$$FB \rightarrow aG, \quad F \rightarrow xxFI, \quad FB \rightarrow e, \quad G \rightarrow aFB, \quad G \rightarrow xxG.$$

In this paper, such words will be called *Fibonacci words* (of even length) and will appear in Section 3.

The three above examples are examples of a *non-ambiguous* algebraic grammar G , that is, each word of $L(G)$ can be formed in a 'unique way' using the productions of P . For example, every Motzkin word w is either the empty word (produced by $M \rightarrow e$), or produced by $M \rightarrow bM$, or produced by $M \rightarrow xM\bar{x}M$. Also, in the second (respectively third) case, the factorization $w = bv$, $v \in M$ (respectively $w = xu\bar{x}v$, $u \in M$, $v \in M$) is unique.

Such properties for the above examples are equivalent to state the following equalities, in algebra $\mathbb{Z}\langle\langle X \rangle\rangle$ for the corresponding (noncommutative) generating functions (where \mathbf{CM} denotes the set of 2-colored Motzkin words):

$$(3) \quad \mathbf{D} = 1 + x\mathbf{D}\bar{x}\mathbf{D},$$

$$(4) \quad \mathbf{M} = 1 + b\mathbf{M} + x\mathbf{M}\bar{x}\mathbf{M},$$

$$(5) \quad \mathbf{CM} = 1 + b\mathbf{CM} + r\mathbf{CM} + x\mathbf{CM}\bar{x}\mathbf{CM},$$

$$(6) \quad \mathbf{FB} = 1 + a\mathbf{G} + xx\mathbf{FB}, \quad \mathbf{G} = a\mathbf{FB} + xx\mathbf{G}.$$

The step going from the non-ambiguous algebraic grammar to the corresponding algebraic system in $\mathbb{Z}\langle\langle X \rangle\rangle$ can be done in a general way. Each element α in N corresponds to an equation $\alpha = \sum_i \beta_i$, where the β_i 's are all the words appearing in the production of the form $\alpha \rightarrow \beta_i$. In the 'good' cases the system has a unique solution (in the α 's). For more details, see, for example, the works of Nivat [28] or Salomaa and Soittola [33], and the pioneer papers of Schützenberger [34, 35] and Chomsky and Schützenberger [5].

Now, if one wants to enumerate $L(G) \cap X^n$ (the words of length n of the language $L(G)$), one sends by a morphism θ all variables $x \in X$ onto a single variable t . We obtain an algebraic system for the corresponding (ordinary) generating function $\theta(L) = f(t) = \sum_{n \geq 0} a_n t^n$ (with $a_n = |L(G) \cap X^n|$).

For the above examples we can solve the corresponding one-variable equation or algebraic system and obtain the following equations:

$$(7) \quad \theta(\mathbf{D}) = c(t) = \frac{1 - (1 - 4t^2)^{1/2}}{2t^2} \quad (\text{Dyck words}),$$

$$(8) \quad \theta(\mathbf{M}) = m(t) = \frac{(1-t) - (1-2t-3t^2)^{1/2}}{2t^2} \quad (\text{Motzkin words}),$$

$$(9) \quad \theta(\mathbf{CM}) = \text{cm}(t) = \frac{(1-2t) - (1-4t)^{1/2}}{2t^2} \quad (\text{2-colored Motzkin words}),$$

$$(10) \quad \theta(\mathbf{FB}) = \text{fb}(t) = \frac{1-t^2}{1-3t^2+t^4} \quad (\text{Fibonacci words with even length}).$$

Note that the coefficient of t^{2n} in $c(t)$ is the classical *Catalan number* $C_n = [1/(n+1)]\binom{2n}{n}$ (which can be obtained by expanding in power series $c(t)$). Also, the coefficient of t^n in $m(t)$ is the less classical *Motzkin number* M_n . Note also the relation

$$(11) \quad 1 + t^2 \text{cm}(t^2) = c(t),$$

which implies that the number of 2-colored Motzkin words of length n equals the Catalan number C_{n+1} (see a bijective proof in Section 3).

The coefficient of t^{2n} in $f(t)$ is the *Fibonacci number* F_{2n} (defined by the recurrence $F_{n+1} = F_n + F_{n-1}$, $F_0 = F_1 = 1$).

The language \mathbf{FB} of Example 2.3 is in fact a *rational language*, that is, a language which can be obtained by applying recursively, from the family of finite languages,

the three operations

$$\text{union}(A, B) \rightarrow A \cup B,$$

$$\text{product}(A, B) \rightarrow AB = \{w = uv, u \in A, v \in B\},$$

star operation $A \rightarrow A^*$, submonoid of X^* generated by A .

For example, we can write

$$(12) \quad \text{FB} = \{\{xx\} \cup a(xx)^*a\}^*.$$

Such an expression for a rational language L , using only the three operations union, product and star operation is called a *rational expression* of L . In the same way as above for grammar, one can define a *non-ambiguous* rational expression. From such an expression for L , one immediately deduces an expression for L by replacing the operations (for languages) union, product and star, respectively by sum, product, and 'quasi-inversion' $A^* \rightarrow (1 - A)^{-1}$ (in the algebra $\mathbb{Z}\langle\langle X \rangle\rangle$). Applying morphism θ , one obtains the generating function $f(t)$ for $a_n = |L \cap X^n|$.

Also, it is possible that the grammar defining L leads to a linear system for $f(t)$. The calculus is thus reduced to a determinant calculus.

L can also be defined by a so-called *finite automaton*, that is a 5-tuple $A = (S, s, F, X, \mu)$ where S is a finite set (the *states*), $s \in S$ (*initial state*), $F \subseteq S$ (*final states*), X a finite set (*input alphabet*) and $\mu : S \times X \rightarrow S$ (*transition function*). The action of μ is extended to words of X^* by defining $\mu^* : S \times X^* \rightarrow S$ with $\mu^*(s, ux) = \mu(\mu^*(s, u), x)$. A word w of X^* is *accepted* by the automaton A iff $\mu^*(s, w) \in F$. The well-known Kleene's Theorem states that L is rational iff L is accepted by a finite automata.

Usually a finite automata is visualized by a labeled graph, with vertices the states and labeled arrows corresponding to μ . An automaton accepting the Fibonacci words of FB of Example 2.3 is displayed in Fig. 2.

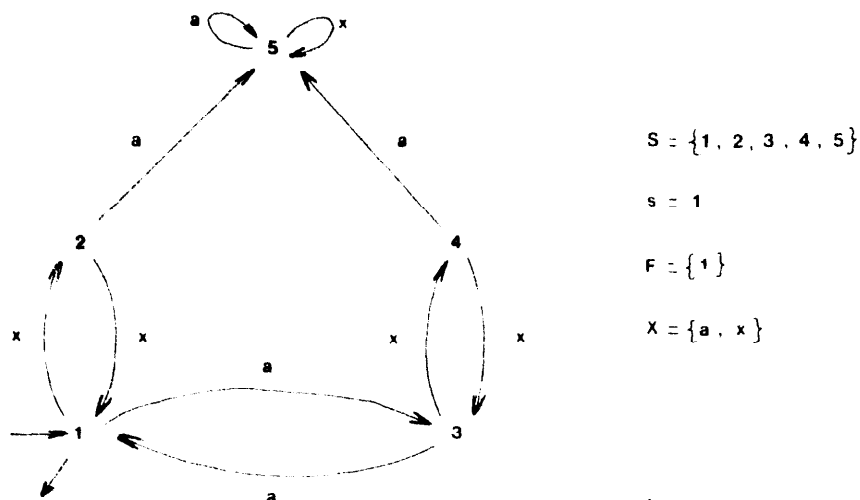


Fig. 2 A finite automaton for the Fibonacci words FB.

Let S be totally ordered, $S = \{s_1, \dots, s_k\}$, and let M be the $k \times k$ matrix $M = (a_{ij})$ where a_{ij} is the formal sum of letters $x \in X$ such that $\mu(s_i, x) = s_j$. Let L_{ij} be the language accepted with initial state $s = s_i$ and final state s_j . Then the (commutative) generating function $\alpha(L_{ij})$ of $\mathbb{Z}[[x]]$ is equal to the (i, j) -coefficient of the matrix $(1 - M)^{-1}$. Thus the calculus is reduced to determinant calculus. Note that a nice expression for the determinant $\det(1 - M)$ is given by

$$(13) \quad \det(1 - M) = \sum_{\gamma_1, \dots, \gamma_r} (-1)^r v(\gamma_1) \dots v(\gamma_r),$$

where $\gamma_1, \dots, \gamma_r$ are two by two disjoint cycles of S (not necessarily covering all the vertices of S), the edges of the cycle are edges of a labeled graph representing the automaton, and the valuation $v(\gamma_i)$ is the product of the letters labeling the edges of γ_i .

An analogous expression exists for the (i, j) -cofactor. This gives an easier way to obtain the generating function $\alpha(L_{ij})$ (see Figs. 2 and 12).

An important notion in algebraic language theory is the concept of *iterative pair* of a word w of an algebraic language L , that is, a factorization of w of the form $w = fugvh$ such that, for every $n \geq 1$, the word $fu^n gv^n h$ is in L . Various 'iterative lemmas' are known (see [3, 29]). We will use here Boasson's lemma. Two iterative pairs of w , $w = f_1 u_1 g_1 v_1 h_1$ and $w = f_2 u_2 g_2 v_2 h_2$ are said to be *overlapping* if one can write $w = \alpha u_1 \beta u_2 \gamma v_1 \delta v_2 \epsilon$ or $w = \alpha u_2 \beta u_1 \gamma v_2 \delta v_1 \epsilon$. Boasson's lemma [4] says that no word of an algebraic language has overlapping iterative pairs.

We show that this lemma implies that there is no hope to encode a convex polyomino P with word of an algebraic language L by just following the border of P .

For example, one can choose a canonical point and encode P by a word with four letters, obtained by following the border (clockwise, for example) and writing the letter corresponding to one of the four possible elementary steps North, South, East or West. It would be possible to characterize words corresponding to convex polyominoes. Another possible coding would be with three letters, corresponding to elementary steps: turn right, turn left or go ahead.

It is easy to see that an iterative pair of $w \in L$ corresponds to define two portions of the border of P , located between the points α, β and γ, δ , such that $(\alpha, \beta, \gamma, \delta)$ is a parallelogram and with some conditions between the first and last step of the portion of paths going from α to β and γ to δ (see Fig. 3). Overlapping iterative pairs are easily found, and thus the language L is not algebraic.

3. Stacks and parallelogram polyominoes

The fundamental idea for the encoding of a convex polyomino P is to split it into three simpler polyominoes. This trisection was introduced by Klarner and Rivest [21] and is defined as follows.

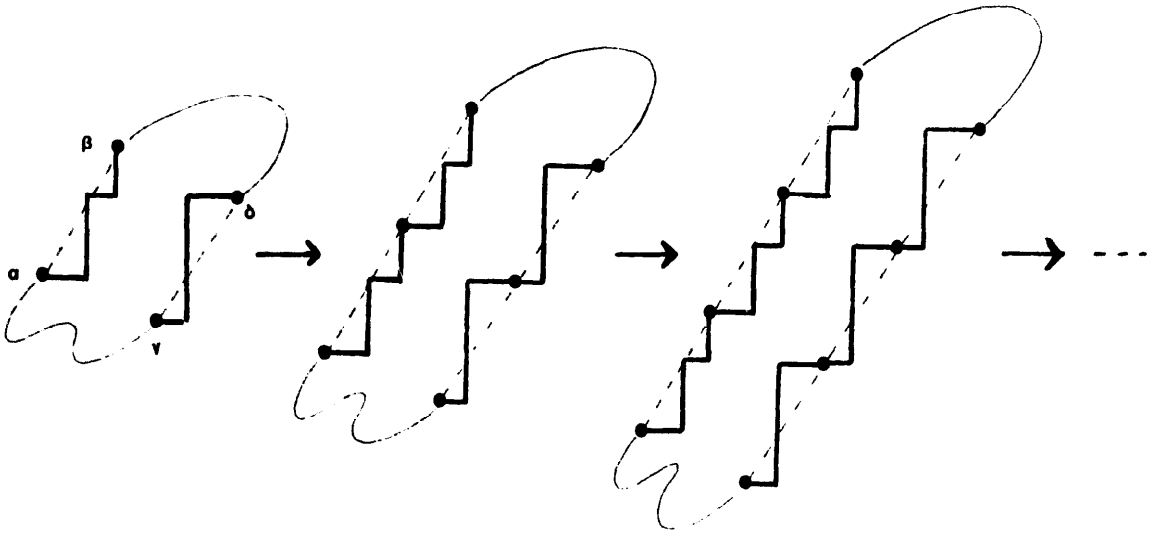


Fig. 3. Iterative pairs.

Notations. Let P be a convex polyomino and $\text{Rect}(P)$ be the smallest *rectangle* (considered as a convex polyomino) containing P . The polyomino P touches the border of $\text{Rect}(P)$ along four connected segments. Each of these segments has two extreme points and thus we introduce 8 points, as shown in Fig. 4. The Westmost (respectively Eastmost) of the points of P on the segment composing the South (respectively North) border of $\text{Rect}(P)$ is denoted by $S(P)$ (respectively $N(P)$).

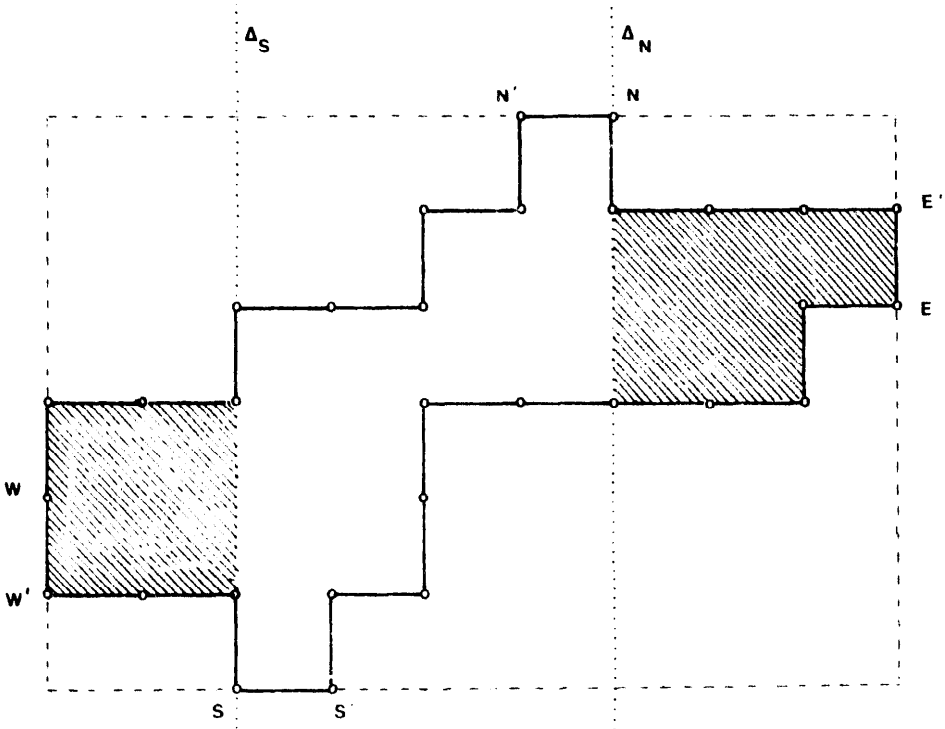


Fig. 4. Trisection of a convex polyomino.

Following counterclockwise the border of P , one meets successively the above 8 canonical points in the order: $S(P)$, $S'(P)$, $E(P)$, $E'(P)$, $N(P)$, $N'(P)$, $W(P)$, $W'(P)$.

Note that some of these points can be identical, that is, possibly $S' = E$, $E' = N$, $N' = W$ or $W' = S$.

We can now define two important subclasses of convex polyominoes. A *parallelogram polyomino* is a convex polyomino P such that $S(P) = W'(P)$ and $N(P) = E'(P)$ (see Fig. 6). A *stack polyomino* is a convex polyomino such that $S(P) = W'(P)$ and $S'(P) = E(P)$ (see Fig. 7).

In other words, a parallelogram polyomino is a polyomino such that the intersection with every line perpendicular to the main diagonal is a connected segment. It can also be characterized with the border formed with two paths, having only East and North elementary steps, having the same initial and final points, and being disjoint except at the extreme points.

For any convex polyomino P , let Δ_S (respectively Δ_N) be the vertical line passing by the points S (respectively N). These two lines split the polyomino P into three (possibly empty) parts. The part between Δ_S and Δ_N is a parallelogram polyomino (or the symmetric, up to a vertical axis, of a parallelogram polyomino). The two extreme parts are, up to 90° rotation, stack polyominoes (see Fig. 4).

In fact, three cases have to be considered according to the fact that Δ_N is at the right of Δ_S (type I polyomino), or $\Delta_N = \Delta_S$ (type II polyomino), or Δ_N is at the left of Δ_S (type III polyomino) (see Fig. 5). Note that the symmetric, up to a vertical axis, of a type III polyomino, is a (special case of) type I polyomino.

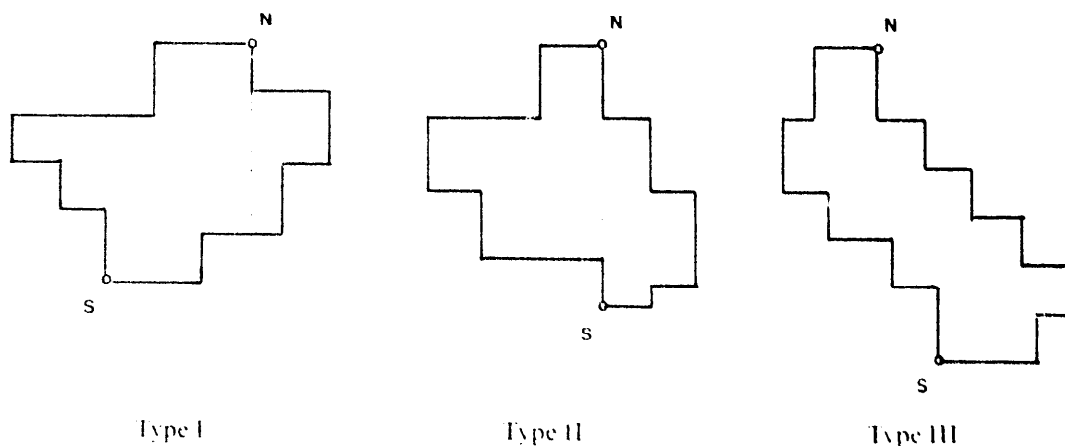


Fig. 5. The three types of convex polyominoes.

The first values of the number of convex polyominoes with the perimeter $2n$ are given in Table 1.

It is easy to enumerate parallelogram and stack polyominoes according to the perimeter.

Table 1. Number of convex polyominoes with perimeter $2n$.

Perimeter $2n$	Type I	Type II	Type III	Total P_{2n}
4	1			1
6	2			2
8	7			7
10	27	1		28
12	110	9	1	120
14	460	55	13	528
16	1948	286	110	2344
18	8296	1362	758	10416
20	35400	6143	4617	46160
22	151056	26729	25895	203680
24	643892	113471	136949	894312

For parallelogram polyominoes, the result and the bijection with Dyck words is now classical (see Polya [30], Gessel and Viennot [13] or Shapiro and Zeilberger [36]). We recall here this bijection.

Let P be a parallelogram polyomino with perimeter $2n+2 \geq 4$. Such a polyomino is defined by two paths ω and η of length $n+1$ starting at $S(P)$ and ending at $N(P)$. The two paths do not intersect, except at the endpoints. We suppose that ω is above (North) path η . To fix the ideas (and the polyomino P), suppose $S(P) = (0, 0)$. For $i \geq 0$, let Δ_i be the line with equation $y = -x + i$. For any i , $1 \leq i \leq n-1$, we look for the kind (North or East) of the elementary steps ω_i (respectively η_i) of the path ω (respectively η), and delimited by the two lines Δ_i and Δ_{i+1} . We thus define a word $w = w_1 \dots w_{n-1}$ of length $n-1$ of $\{x, \bar{x}, b, r\}^*$ by the following condition,

$$(14) \quad \begin{aligned} w_i &= x \text{ if } \omega_i \text{ is a North step and } \eta_i \text{ an East step,} \\ w_i &= \bar{x} \text{ if } \omega_i \text{ is an East step and } \eta_i \text{ a North step,} \\ w_i &= b \text{ if } \omega_i \text{ is a North step and } \eta_i \text{ a North step,} \\ w_i &= r \text{ if } \omega_i \text{ is an East step and } \eta_i \text{ an East step.} \end{aligned}$$

The word w satisfies condition (1) because the two paths ω and η are not intersecting, and condition (2) because the two paths end at the same point. Thus, w is a 2-colored Motzkin word (Example 2.2). Obviously, the map $P \rightarrow w$ is a bijection between parallelogram polyominoes with perimeter $2n+2$ and 2-colored Motzkin words of length $n-1$.

Now for such a word $w = w_1 \dots w_{n-1}$ we define the word $v = xh(w_1) \dots h(w_{n-1})\bar{x}$ where h is the morphism $\{x, \bar{x}, b, r\}^* \rightarrow \{x, \bar{x}\}^*$ defined by the condition

$$(15) \quad h(x) = xx, \quad h(\bar{x}) = \bar{x}\bar{x}, \quad h(b) = x\bar{x}, \quad h(r) = \bar{x}x.$$

It is easily shown that the map $w \rightarrow v = xh(w)\bar{x}$ is a bijection between 2-colored Motzkin words of length $n-1$ and Dyck words of length $2n$. We deduce the following lemma.

Lemma 3.1. *The number of parallelogram polyominoes with perimeter $2n+2$ is the Catalan number $C_n = [1/(n+1)]\binom{2n}{n}$.*

The map $P \rightarrow v = \gamma(P)$ defined above by conditions (14) and (15) is a bijection. An example is displayed in Fig. 6.

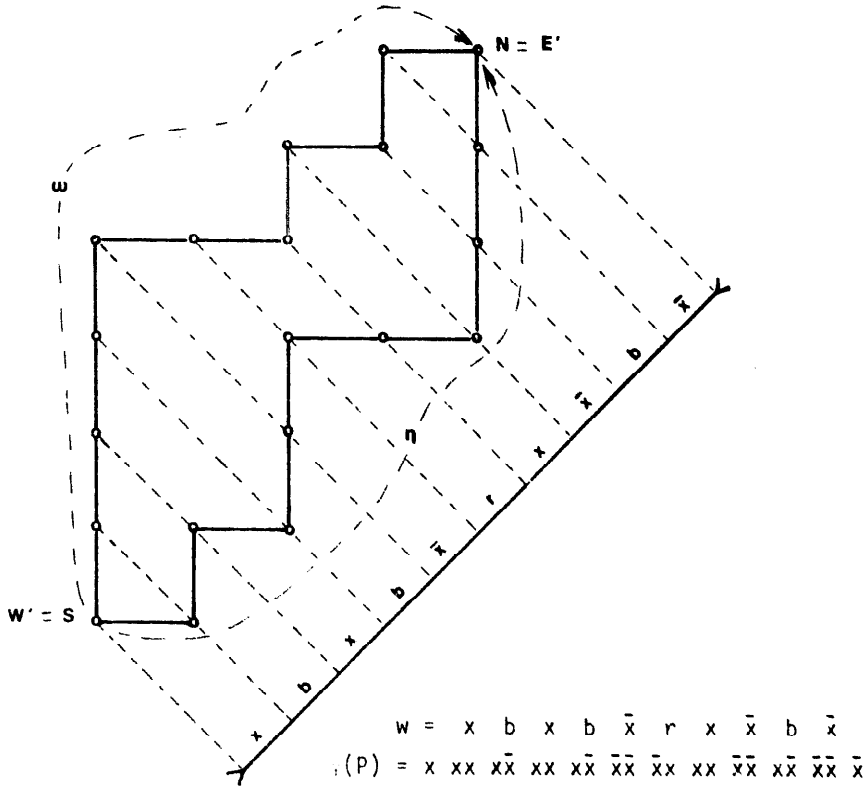


Fig. 6. The (classical) bijection γ between parallelogram polyominoes and Dyck words.

Now let P be a stack polyomino with perimeter $2n+4$. The border of such a polyomino is made with two non-intersecting (except at the endpoints) paths ω and η . Path η is a path going from $S(P)$ to $S'(P) = E(P)$ and composed only with p East steps. Path ω is above η , goes from $S(P)$ to $E(P)$, and is made with p East steps, $n-p+2$ North steps and $n-p+2$ South steps. In this path, the North steps occur before the South steps, and the first (respectively last) step is always a North (respectively South) step. We define word w of length $2n+4$ of $\{a, x\}^*$ by following path ω from $S(P)$ to $E(P)$ and writing the letter "a" (respectively the factor xx) each time one meets a North or South step (respectively an East step). We obtain a word characterized by the three following conditions:

- (16) $w \in \{a, xx\}^* \subseteq \{a, x\}^*$,
- (17) the first (respectively last) letter of w is "a",
- (18) $|w|_a$ is an even number, say $2p$ and after the p th letter "a" there is a factor xx .

If we delete the first and last letter of w , and a factor xx (which always exists) between the p th and $(p+1)$ st a 's, we obtain a word $v = \Phi(P)$ of even length $2n$.

Obviously, the map $P \rightarrow v = \Phi(P)$ is a bijection between stack polyominoes with perimeter $2n+4$ and words of length $2n$ satisfying (16) having an even number of "a", that is nothing but what we have called Fibonacci words in Example 2.3. An example of the bijection Φ is given in Fig. 7.

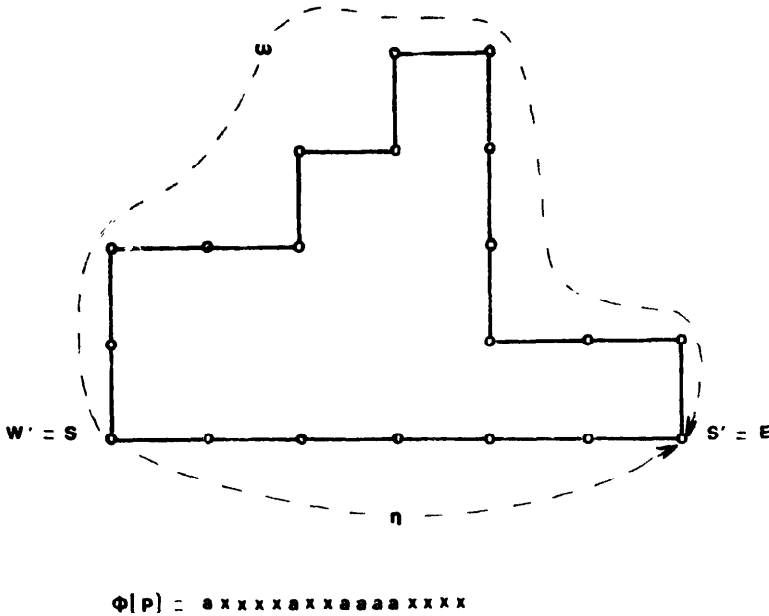


Fig. 7. The bijection Φ between stack polyominoes and Fibonacci words.

From Example 2.3 and equation (10), we deduce the following lemma.

Lemma 3.2. *The number of stack polyominoes with perimeter $2n+4$ is the Fibonacci number F_{2n} with generating function*

$$\sum_{n \geq 0} F_{2n} t^{2n} = \frac{1-t^2}{(1-t-t^2)(1+t-t^2)}$$

Thus each part of the trisection of a convex polyomino can be encoded by a word of an algebraic (or rational) language. The idea is to mix the three codings in a single one. The major problem is to define a 'gluing' process, keeping the algebraic as well as the convex property. Unfortunately, with the above coding γ between parallelogram polyominoes and Dyck words, this 'gluing' process would lead to non-algebraic languages. It would be possible to 'insert' the coding with Fibonacci words into 2-colored Motzkin words, but it seems impossible to obtain words of an algebraic language having a tractable associated algebraic system of equations.

We are going to give a more elaborate bijection between parallelogram polyominoes and Dyck words, which will fit very well with our 'gluing' problem.

4. The bijection β between Dyck words and parallelogram polyominoes

Let w be a Dyck word of length $2n$. A factor $x\bar{x}$ (respectively $\bar{x}x$) is called a *peak* (respectively *trough*) of w . We number the peaks (respectively troughs) from left to right. Let $k \geq 1$ be the number of peaks of w (thus w has $k - 1$ troughs). The *height* of the peak $w = fx\bar{x}g$ is defined by $1 + \delta(f)$ with δ the function

$$(19) \quad \delta(f) = |f|_x - |f|_{\bar{x}}.$$

The *height* of the trough $w = f\bar{x}xg$ is defined as $\delta(f)$.

To w , we associate two sequences $a(w) = (a_1, \dots, a_k)$ and $b(w) = (b_1, \dots, b_{k-1})$ by the condition

(20) for $i, 1 \leq i \leq k$ (respectively $1 \leq i \leq k - 1$), the number a_i (respectively b_i) is the height of the i th peak (respectively trough).

Obviously, these two sequences satisfy the following condition,

(21) for any $i, 1 \leq i \leq k - 1, \quad 1 \leq b_i \leq a_i \quad \text{and} \quad 1 \leq b_i \leq a_{i+1}.$

For any $i, 1 \leq i \leq k$, we consider a vertical strip of i cells. Inside each strip, the cells are ordered down-up. Now we 'glue' these strips together according to the sequence $b(w)$. More precisely, for $i, 1 \leq i \leq k - 1$, the $(i + 1)$ st strip is 'glued' on the right of the i th strip such that the first b_i cells of the $(i + 1)$ st strip are 'glued' to the last b_i cells of the i th strip (see Fig. 8). Formally these two strips form a

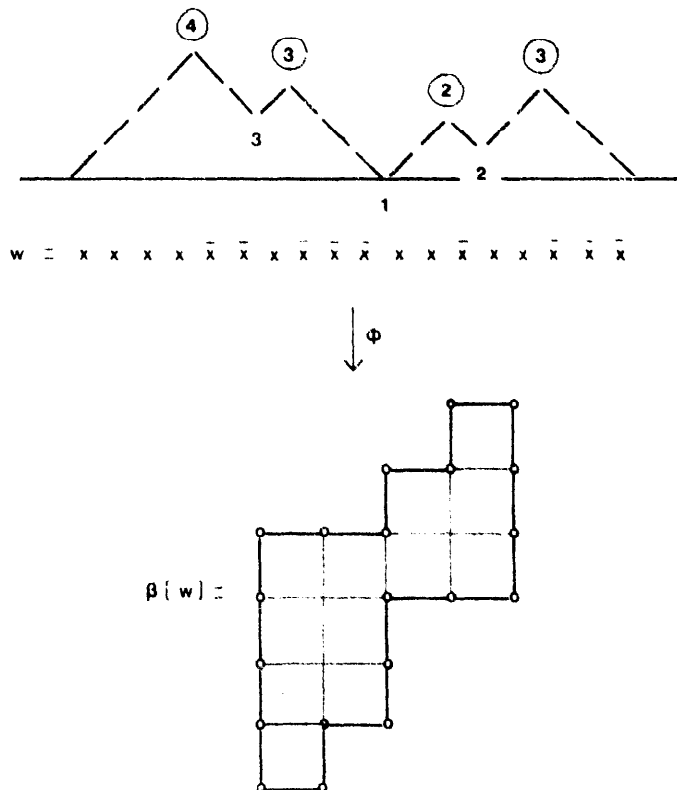


Fig. 8. The bijection β between Dyck words and parallelogram polyominoes.

parallelogram polyomino P_i such that the path $\omega(P_i)$ (respectively $\eta(P_i)$) defined in Section 3 has the following sequence of elementary steps: North a_i times, East, North $(a_{i+1} - b_i)$ times, East (respectively East, North $(a_i - b_i)$ times, East, North a_{i-1} times).

We thus obtain a parallelogram polyomino P denoted by $P = \beta(w)$.

The perimeter p is given by

$$(22) \quad p = \sum_{i=1}^k (2 + 2a_i - b_i - b_{i-1})$$

with the convention $b_0 = b_k = 0$.

The sum (22) is easily seen to be equal to the length of the Dyck word w , increased by 2.

We now describe a map μ which will be the reverse bijection of β .

Let P be a parallelogram polyomino with perimeter $2n + 2$. Let k be the number of East steps of the paths $\omega(P)$ and $\eta(P)$ defined in Section 3. The polyomino P is formed with k vertical strips, which we number from left to right. We define two sequences of integers $a(P) = (a_1, \dots, a_k)$ and $b(P) = (b_1, \dots, b_{k-1})$ such that a_i is the number of cells of the i th strip, and b_i is the length of the segment common to the border of the i th and $(i + 1)$ st strips.

Obviously, such sequences satisfy relations (21) and (22) (with $p = 2n + 2$). Making the convention $b_0 = b_k = 0$, we define the word $w = u_1 v_1 \dots u_k v_k$ by the relation

$$(23) \quad \text{for } 1 \leq i \leq k, \quad u_i = x^{(1+a_i-b_{i-1})} \quad \text{and} \quad v_i = \bar{x}^{(1+a_i-b_i)}.$$

It is easy to check that w is a Dyck word. From relation (22), the length of w is $2n$. Denote by $\mu(P) = w$ the Dyck word thus obtained.

It is easy to see that maps β and μ are the inverses of each other. We thus have the following proposition.

Proposition 4.1. *The map β defined above is a bijection from Dyck words of length $2n$ onto parallelogram polyominoes of perimeter $2n + 2$. The area of P is the sum of the height of the peaks of $\beta(P)$.*

Remark 4.2. For a parallelogram polyomino P , define $l(P)$ (respectively $r(P)$) as to be the distance between the two points (defined in Section 3) $W(P)$ and $W'(P)$ (respectively $E(P)$ and $E'(P)$). For a Dyck word w , let $l(P)$ (respectively $r(P)$) be the maximum length of a left (respectively right) factor of w equal to a power of x (respectively \bar{x}). We have proved that the double distribution (l, r) for parallelogram polyominoes with perimeter $2n + 2$ is the same as the double distribution (l, r) for Dyck words with length $2n$.

Remark 4.3. Using the classical 'André's reflexion principle', it can be proved that the number $a_{n,i}$ of Dyck words w of length $2n$ such that $l(w) = i$ and $r(w) = j$

(defined in Remark 4.2) is given by

$$(24) \quad a_{n,i,j} = \binom{2n-i-j-2}{n-j-1} - \binom{2n-i-j-2}{n-1},$$

with the convention $\binom{a}{b} = 0$ when $b > a$.

Remark 4.4. Define the width of a parallelogram polyomino to be the number of vertical strips composing this polyomino. Applying Gessel and Viennot's [13] methodology interpreting determinants as noncrossing paths, it is easy to prove that the number $b_{n,k}$ of parallelogram polyominoes with perimeter $2n + 2$ and width k is given by the 2×2 determinant

$$(25) \quad b_{n,k} = \begin{vmatrix} \binom{n-1}{k-1} & \binom{n-1}{k} \\ \binom{n}{k-1} & \binom{n}{k} \end{vmatrix}.$$

This determinant is equal to $(1/n)\binom{n}{k}\binom{n}{k-1}$ and using Proposition 4.1, we deduce the well-known formula for the number of Dyck words of length $2n$ having k peaks (see, for example, Kreweras [22]).

Remark 4.5. It is interesting to note that the bijection β can be defined in a completely different way, using binary trees and some new combinatorial properties relating 'prefix order', 'symmetric order' and 'height' of vertices. We suppose here that the reader is familiar with this terminology for binary trees and briefly describe the other version of β . For more details and combinatorial properties, the reader is referred to Viennot [41].

Let $w = w_1 \dots w_{2n}$ be a Dyck word of length $2n$. First we define the Dyck word $\tilde{w} = \tilde{w}_{2n} \dots \tilde{w}_1$ where \tilde{w}_i is x (respectively \bar{x}) if w_i is \bar{x} (respectively x). It is very classical to associate to the Dyck word \tilde{w} a complete binary tree B (with $2n + 1$ vertices) using the prefix order (here the left subtree is traversed after the right subtree). Then deleting the leaves of B , we obtain a binary tree b (with n vertices) which we order according to the symmetric order (called also *inorder*) (see Fig. 9). Now we define a path ω of length $n - 1$ such that, for $1 \leq i \leq n - 1$, the i th elementary step is North (respectively East) if the i th vertex of b has (respectively does not have) a right son. The number of East steps is thus the number of left edges of the binary tree b . Under the j th East step of ω we put a horizontal edge such that the distance between these two edges is equal to the right height of the j th vertex of b having a left son. It is a combinatorial property of binary trees that these horizontal edges are East steps of a (unique) path η having only East and North steps and same endpoints as ω . Sliding 45° downwards path η gives a path η' , and adding two 'corners' we obtain a parallelogram polyomino P (see Fig. 9).

It can be proved that this map $w \rightarrow P$ is a bijection, identical to β . The reverse map can also be described, using the analogue, for binary trees, of Schützenberger's 'jeu de taquin' for Young tableaux (see Viennot [41]).

$w = x x x x x \bar{x} \bar{x} x \bar{x} \bar{x} \bar{x} x x \bar{x} x x \bar{x} \bar{x} \bar{x}$
 $\tilde{w} = x x x x \bar{x} \bar{x} x \bar{x} \bar{x} x x x \bar{x} x x \bar{x} \bar{x} \bar{x}$

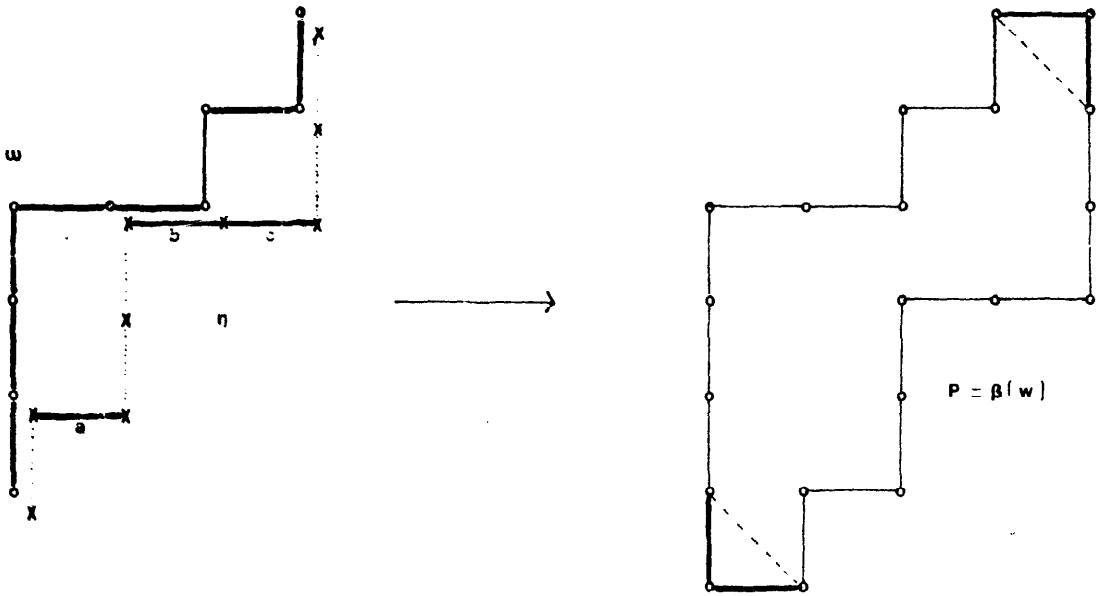
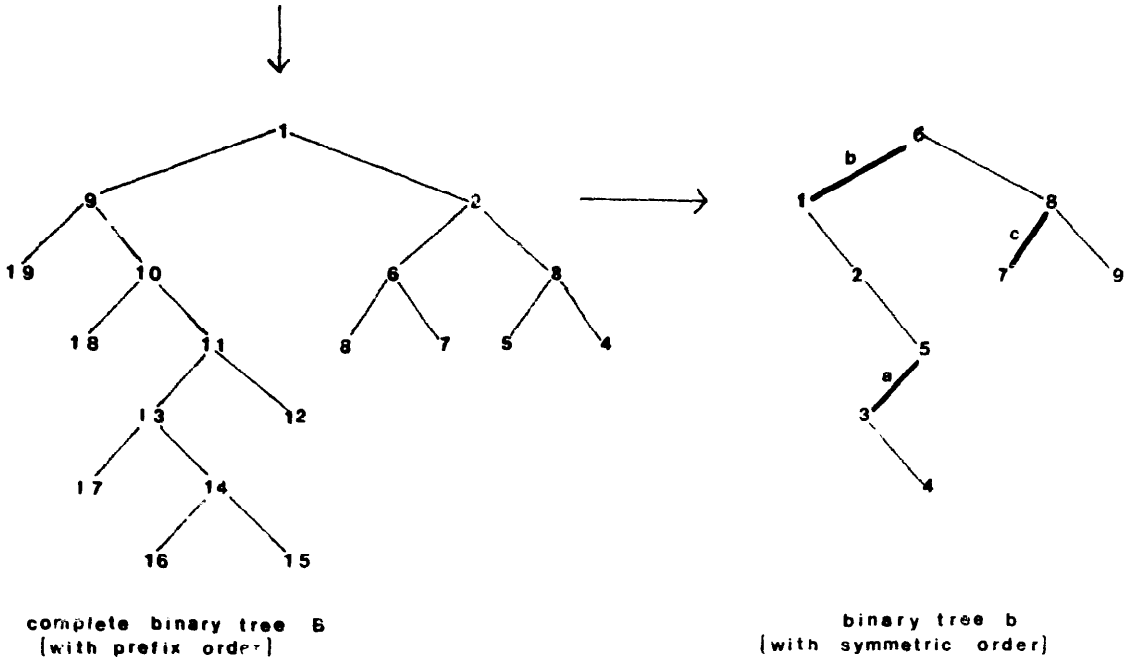


Fig. 9. The bijection β with binary trees.

5. The coding of type I polyominoes

We combine the bijection Φ (between stack polyominoes and Fibonacci words) with the bijection β (between parallelogram polyominoes and Dyck words) in order to give a bijection between type I (convex) polyominoes and some words of an algebraic language.

$$w = \underbrace{x a a x a a x x \bar{x} \bar{x} x \bar{x} \bar{x} \bar{x} x x \bar{x} x x \bar{x} a a \bar{x} a a a a \bar{y}}_{w_1} \quad \underbrace{\hspace{15em}}_{w_3}$$

$$\downarrow \qquad \qquad \qquad \downarrow$$

$$w'_1 = x a a x x a a x \qquad w'_3 = a a \bar{x} a \bar{x} a a a \bar{x}$$

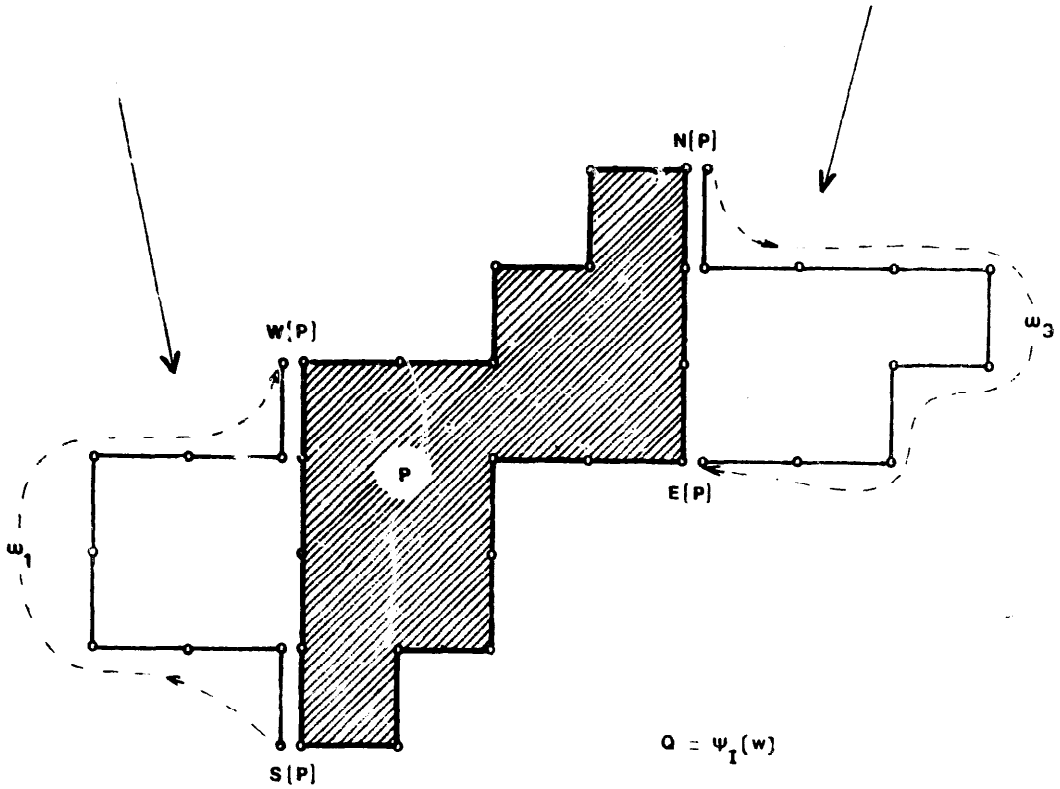


Fig. 10. The bijection Ψ_1 .

The reader familiar with algebraic languages theory will easily construct a 'push-down automaton' accepting the set of pigmented Dyck words. Thus this language is algebraic (see next section).

6. The generating function for type I polyominoes

In this section we introduce auxiliary languages and give the generating function (in one variable) for the pigmented Dyck language.

Notations. We denote by Y the alphabet $\{x, \bar{x}, y, \bar{y}\}$ and by μ the morphism $Y^* \rightarrow \{x, \bar{x}\}^*$ defined by its action on the letters of Y^* :

$$(27) \quad \mu(x) = \mu(y) = x \quad \text{and} \quad \mu(\bar{x}) = \mu(\bar{y}) = \bar{x}.$$

Let H be the set of words w of Y^* satisfying the following two conditions:

(28) $\mu(w)$ is a Dyck word of $\{x, \bar{x}\}^*$,

(29) w has one of the following forms: $w = y^k \bar{y}^k, k \geq 1$, or $w = u \bar{x} w' x v$ with $u \in \{y\}^*$, $v \in \{\bar{y}\}^*$ and $w' \in \{x, \bar{x}\}^*$.

In other words, a word w of H is obtained from a Dyck word w'' of $\{x, \bar{x}\}^*$ by changing each letter x (respectively \bar{x}) of the left (respectively right) leg of w'' into y (respectively \bar{y}).

Let L be the set of words of Y^* satisfying condition (28) and

(29a) $w = e$ or $w = u \bar{x} w'$ with $u \in \{y\}^*$ and $w' \in \{x, \bar{x}\}^*$.

Let R be the set of words of Y^* satisfying condition (28) and

(29b) $w = e$ or $w = w' x v$ with $v \in \{\bar{y}\}^*$ and $w' \in \{x, \bar{x}\}^*$.

The noncommutative generating function $H \in \mathbb{Z}\langle Y \rangle$ of the auxiliary language H satisfies the following algebraic system of equations:

$$\begin{aligned}
 (30) \quad & H = y\bar{y} + yH\bar{y} + yL\bar{x}DxR\bar{y}, \\
 & L = 1 + yL\bar{x}D, \\
 & R = 1 + DxR\bar{y}, \\
 & D = 1 + xD\bar{x}D.
 \end{aligned}$$

The first equation is just a translation of the following fact. For every word $w \in H$, let $\mu(w) = u_1 \dots u_k$ be the unique factorization of the Dyck word w into prime words w_i (Dyck words which are not product of other Dyck words). Then if $k = 1$, w has the form $w = yu\bar{y}$ with either $u = e$, or $u \in H$. If $k \neq 1$, then w has a unique factorization $w = \bar{y}w_1 \bar{x}w'xw_k \bar{y}$ with $w_1 \in L, w' \in D, w_k \in R$ (and in fact $\mu(yw_1 \bar{x}) = u_1$ and $\mu(xw_k \bar{y}) = u_k$).

The second and third equations come from analogous properties for the languages L and R .

Let Z be the alphabet $\{x, \bar{x}, y, \bar{y}, a, b\}$. We define the two substitution operators λ and ρ sending every word $w \in Y^*$ into a set of words of Z^* . The words of $\lambda(w)$ (respectively $\rho(w)$) are the words obtained from w by changing any letter y (respectively \bar{y}) into the factor $a^i x$ (respectively $\bar{x} b^i$) with $i \geq 0$. In language theory, λ and ρ are very simple examples of a so-called *rational transduction*.

Let $S \subseteq \{x, \bar{x}, a, b\}^*$ be the language defined by

(31) $w \in S$ iff there exists $u, v \in Z^*$ such that $y u \bar{y} \in H, v \in \lambda(\rho(u)), w = x v \bar{x}$.

Let $V \subseteq S$ the set of words $w \in S$ such that

(32) $|w|_a$ and $|w|_b$ are even.

The reader can easily prove the following lemma.

Lemma 6.1. *The language obtained by replacing in each word of V the letter “b” by “a” (V defined by (31) and (32)) is the set of pigmented Dyck words defined by (26).*

Remark 6.2 (for ‘language theorists’). The pigmented Dyck is algebraic because V is obtained from the Dyck language D using the operations rational transduction and intersection with a rational language.

Notations. We introduce the following generating functions in commutative variables $x, \bar{x}, y, \bar{y}, a, b$:

$$\begin{aligned} h(x, \bar{x}, y, \bar{y}) &= \alpha(H), & l(x, \bar{x}, y, \bar{y}) &= \alpha(L), \\ r(x, \bar{x}, y, \bar{y}) &= \alpha(R), & s(x, \bar{x}, a, b) &= \alpha(S), \\ v(x, \bar{x}, a, b) &= \alpha(R) \quad \text{and} \quad d(x, \bar{x}) &= \alpha(D) = \frac{1 - (1 - 4x\bar{x})^{1/2}}{2x\bar{x}}. \end{aligned}$$

From Proposition 5.2 and Lemma 6.1, the (ordinary) generating function $\sum_{n \geq 1} p_{2n+2}^1 t^{2n}$ for the number p_{2n+2}^1 of type I polyominoes with perimeter $2n + 2$ is the formal series obtained by sending all variables on t , that is,

$$(33) \quad p_1(t) = \sum_{n \geq 2} p_{2n}^1 t^{2n} = t^2 \theta(v).$$

The substitutions λ and ρ defined on the family of languages of Z^* have analogues in the set of formal series.

We define the morphisms (of commutative algebra) $\bar{\lambda}$ and $\bar{\rho} : \mathbb{Z}[[Y]] \rightarrow \mathbb{Z}[[Z]]$ by the conditions

$$(34a) \quad \bar{\lambda}(y) = \frac{x}{1-a}, \quad \bar{\lambda}(z) = z \quad \text{for } z \in Y, z \neq y,$$

$$(34b) \quad \bar{\rho}(\bar{y}) = \frac{\bar{x}}{1-b}, \quad \bar{\rho}(z) = z \quad \text{for } z \in Y, z \neq \bar{y}.$$

The reader will easily prove that relation (31) can be extended to power series in the following way:

$$(35) \quad s(x, \bar{x}, a, b) = (1-a)(1-b)h\left(x, \bar{x}, \frac{x}{1-a}, \frac{\bar{x}}{1-b}\right).$$

In order to obtain the analogue, for power series, of relation (32), it suffices to take according to the variable ‘‘a’’ the even part of the power series s , and then take again, according to the variable b , the even part of the result. Thus

$$(36) \quad v(x, \bar{x}, a, b) = \frac{1}{4}(s(x, \bar{x}, a, b) + s(x, \bar{x}, -a, b) + s(x, \bar{x}, a, -b) + s(x, \bar{x}, -a, -b)).$$

Noting that s is symmetric in a and b , we deduce, from (33),

$$(37) \quad p_1(t) = \frac{1}{4}t^2(s(t, t, t, t) + 2s(t, t, t, -t) + s(t, t, -t, -t)).$$

The generating functions $a(x, \bar{x}, y, \bar{y})$, $l(x, \bar{x}, y, \bar{y})$ and $r(x, \bar{x}, y, \bar{y})$ satisfy a commutative analogue of the algebraic system (30), which can be solved by linear equations:

$$(38) \quad l = \frac{1}{1 - d\bar{x}y}, \quad r = \frac{1}{1 - dx\bar{y}},$$

$$(39) \quad h(x, \bar{x}, y, \bar{y}) = \frac{y\bar{y}}{1 - y\bar{y}} (1 + x\bar{x}dlr).$$

An explicit expression for $p_l(t)$ can be obtained from relations (35), (37), (38) and (39).

In order to simplify the calculus we denote

$$\Delta(t) = (1 - 4t^2)^{1/2}, \quad c(t) = \frac{1 - \Delta}{2t^2} = d(t, t) \quad \text{and} \quad \bar{c}(t) = \frac{1 + \Delta}{2t^2}.$$

We have

$$(40) \quad c\bar{c} = c + \bar{c} = \frac{1}{t^2} \quad \text{and} \quad c^2t^2 = c - 1.$$

From (38) and (39) we deduce

$$(41) \quad h(x, \bar{x}, y, \bar{y}) = \frac{y\bar{y}}{1 - y\bar{y}} \left(1 + \frac{x\bar{x}d}{(1 - d\bar{x}y)(1 - dx\bar{y})} \right).$$

Using (35) we have

$$(42) \quad s(t, t, a, b) = \frac{t^2(1-a)(1-b)}{(1-a-b+ab-t^2)} \left(1 + \frac{t^2(1-a)(1-b)c}{(1-a-t^2c)(1-b-t^2c)} \right),$$

$$(43) \quad s(t, t, t, t) = \frac{t^2(1-t)^2(1-2t+t^4c)}{(1-2t)^2(1-t^2c)}.$$

Multiplying numerator and denominator by $(1 - t^2\bar{c})$ gives

$$(44) \quad s(t, t, t, t) = \frac{(1-t)^2(1-2t+t^2-2t^4-(1-t)^2\Delta)}{2(1-2t)^2}.$$

Similarly we compute successively

$$(45) \quad s(t, t, t, -t) = \frac{t^2(1-t)(1+t)}{(1-2t^2)} \left(1 + \frac{t^2(1-t^2)c}{(1+t-t^2c)(1-t-t^2c)} \right),$$

$$(46) \quad s(t, t, t, -t) = \frac{t^2(1-t^2)(1-2t^2-t^4c)}{(1-2t^2)(1-2t^2-t^2c)}.$$

Multiplying numerator and denominator by $(1 - 2t^2 - t^2\bar{c})$ gives

$$(47) \quad s(t, t, t, -t) = \frac{(1-t^2)((1-2t)(1+2t)(1-3t^2) - (1-t^2)(1-2t^2)\Delta)}{2(1-2t^2)(2t-1)(2t+1)}.$$

Also from (42) we have

$$(48) \quad s(t, t, -t, -t) = \frac{t^2(1+t)^2}{(1+2t)} \left(1 + \frac{t^2(1+t)^2 c}{(1+t-t^2c)} \right),$$

$$(49) \quad s(t, t, -t, -t) = \frac{t^2(1+t)^2(1+2t+t^4c)}{(1+2t)^2(1-t^2c)}.$$

Multiplying numerator and denominator by $1 - t^2\bar{c}$ gives similarly

$$(50) \quad s(t, t, -t, -t) = \frac{(1+t)^2((1+2t+t^2-2t^4-(1+t)^2\Delta))}{2(1+2t)^2}.$$

Now using (37), and adding (44), (47) and (50) we obtain (using MACSYMA) the following generating function for type I polyominoes.

Theorem 6.3. *The generating function for the number p_{2n}^1 of type I polyominoes having perimeter $2n$ is*

$$\sum_{n \geq 2} p_{2n}^1 t^{2n} = \frac{t^4(1-8t^2+21t^4-19t^6+4t^8)}{(1-2t)^2(1+2t)^2(1-2t^2)} - 2t^8(1-4t^2)^{-3/2}.$$

7. The coding of type II polyominoes

Type II polyominoes (i.e., polyominoes P such that $S(P)$ and $N(P)$ are on the same vertical line) are obtained by ‘gluing’ together two stack polyominoes, as shown in Fig. 11. In this section we construct a bijection Ψ_{II} between type II polyominoes having perimeter $2n+8$ and a certain set B of pairs (u, v) of words, with total length $|u|+|v|=2n$. The construction for Ψ_{II} is a more elaborate version of the construction for Φ introduced in Section 3, and related to Fibonacci numbers. Dyck language is not involved here and the final generating function $p_{II}(t)$ is rational.

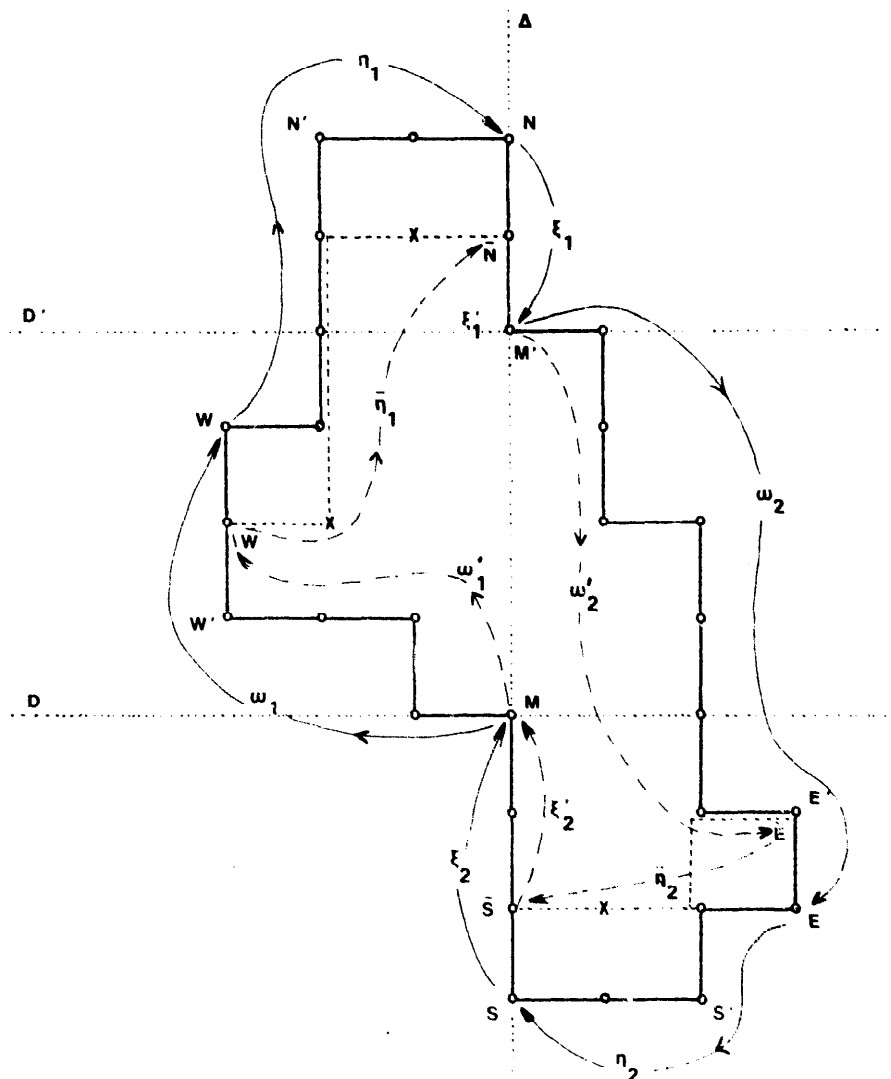
Let B be the set of all pairs of words $u, v \in \{x, z, a\}^*$ satisfying the three following conditions:

- (51a) u and $v \in \{a, z\}^* \cdot \{a, xx\}^*$,
- (51b) $|u|_z = |v|_z \neq 0$,
- (51c) $|u|_a$ and $|v|_a$ are even numbers.

Remark 7.1. For $(u, v) \in B$, $|u|+|v|$ is even ($=2n$) and such a pair is in bijection with the word u/v (of length $2n+1$) of a certain language $\bar{B} \subseteq \{x, z, a, /\}^*$. Language theorists will easily see that \bar{B} is a ‘linear’ language, and thus algebraic.

Proposition 7.2. *There exists a bijection between type II polyominoes with perimeter $2n+8$ and pairs $(u, v) \in B$ (defined by (51)) with $|u|+|v|=2n$.*

Proof. Let P be a type II polyomino with perimeter $2n+8$. Let Δ be the vertical



$$\Psi_{II}(P) : (z a a z a z z x x a , z z a z z x x a a x x a)$$

Fig. 11. The bijection Ψ_{II} .

line where $S(P)$ and $N(P)$ are located. The border of P intersects along two disjoint segments. Let $M = M(P)$ (respectively $M' = M'(P)$) be the upper (respectively lower) extreme point of the segment having $S(P)$ (respectively $N(P)$) as other extreme point. Let D (respectively D') be the horizontal line containing the point M (respectively M'). Starting from M and following clockwise the border of P , we successively meet the following paths:

- ω_1 from M to W having West and North steps,
- η_1 from W to N having East and North steps,
- ξ_1 from N to M' having only South steps,
- ω_2 from M' to E having East and South steps,
- η_2 from E to S having West and South steps,
- ξ_2 from S to M having only North steps.

Remark 7.3. Note that the first (respectively last) step of ω_1 is necessarily West (respectively North). The first (respectively last) step of ω_2 is West (respectively South). Also the last step of η_1 (respectively η_2) is East (respectively West) and ξ_1 and ξ_2 are nonempty.

Proof of Proposition 7.2 (continued). Let T_S (respectively T_N) be the ‘South translation’ $(x, y) \rightarrow (x, y - 1)$ (respectively ‘North translation’ $(x, y) \rightarrow (x, y + 1)$). We define $\bar{W} = T_S(W)$, $\bar{N} = T_S(N)$, $\bar{E} = T_N(E)$ and $\bar{S} = T_N(S)$. The step (W, \bar{W}) (respectively (\bar{E}, E)) is the last step of the path ω_1 (respectively ω_2). Let ω'_1 (respectively ω'_2) be the path obtained from ω_1 (respectively ω_2) by deleting this last step. The step (N, \bar{N}) (respectively (S, \bar{S})) is the first step of the path ξ_1 (respectively ξ_2). Let ξ'_1 (respectively ξ'_2) be the path obtained by deleting this first step.

We define $\bar{\eta}_1 = T_S(\eta_1)$ (path going from \bar{W} to \bar{N}) and $\bar{\eta}_2 = T_N(\eta_2)$ (path going from \bar{E} to \bar{S}).

From Remark 7.3, the polyomino P can be reconstructed from the six paths ω'_1 , ω'_2 , ξ'_1 , ξ'_2 , $\bar{\eta}_1$, $\bar{\eta}_2$. The total length of these paths is $2n + 4$.

We define the word u' by following (from M to \bar{N} via \bar{W}) first the path ω'_1 , and then the path $\bar{\eta}_1$, and by writing a letter “ a ” (respectively “ z ”, a factor xx) each time we make an East or West step (respectively a North step located between the two lines D and D' , a North step above the line D').

Dually, we define the word v' by following (from M' to \bar{S} via \bar{E}) first the path ω'_2 , and then the path $\bar{\eta}_2$, and by writing a letter “ a ” (respectively “ z ”, a factor xx) each time we make an East or West step (respectively a South step located between the two lines D and D' , a South step below the line D).

Example. For the polyomino P (with perimeter 30) displayed in Fig. 11, we have

$$u' = azaazazzxxaa \quad \text{and} \quad v' = azzazzxxaaxxaa.$$

Obviously (cf. Remark 7.3) the first and last letter of u' (respectively v') is “ a ”. We define u (respectively v) to be the word obtained by deleting these two letters, and finally

$$(52) \quad \Psi_{11}(P) = (u, v).$$

It is easy to verify that $|u| + |v| = 2n$ and that (u, v) satisfies the three conditions (51a)-(51c) defining the set B .

Conversely, let (u, v) be a pair of B with $|u| + |v| = 2n$. We define $u' = aua$ and $v' = ava$. The number of “ x ” in u' is an even number, say $2r$ (≥ 0). Also denote $|v'|_a = 2p \geq 2$. Let u'' be the word obtained from u' by replacing each factor xx by “ x ” and by inserting an “ x ” just after the p th letter “ a ”. Also let $2s = |v'|_x \geq 0$, $2q = |v'|_a \geq 2$ and v'' be the word obtained in a similar way. Let $m = |u|_z = |v|_z$. We define the four points $S = (0, -(s + 1))$, $M = (0, 0)$, $M' = (0, m)$, $N = (0, m + r + 1)$. These four points are all distinct.

Reading from left to right the word u'' (respectively v'') and similarly to the process defining Φ^{-1} in Section 3, we construct a path χ_1 (respectively χ_2) going from M

to \bar{N} (respectively M' to \bar{S}). Each letter "x" or "z" corresponds to a North (respectively South) step. The first p (respectively q) letters "a" correspond to a West (respectively East) step. The last p (respectively q) letters "a" correspond to an East (respectively West) step.

Now we respectively define ξ_1 (ξ_2) to be the path going from N to M' (S to M). The path ξ_1 (ξ_2) has $r+1$ ($s+1$) South (North) steps.

It is easy to verify that the four paths $\xi_1, \xi_2, \chi_1, \chi_2$ define the border of a type II polyomino P with perimeter $2n+8$. Also, the reader will verify that the two correspondences defined above are inverse each other. \square

8. The generating function for type II polyominoes

The language \bar{B} defined in Remark 7.1 and coding type II polyominoes is algebraic but not rational. It would be possible to give a length preserving bijection between words of \bar{B} and words of a rational language, using a *multi-head automaton*.

These kinds of automaton are well known in theoretical computer science (see, for example, [27, 32, 43]). A word $w = u/v$ of \bar{B} can be recognized by a two-head finite automaton in the following sense. At the beginning, the heads are pointing at the first letter of u and v . The automaton begins to read u . Each time one of the heads reads a letter z , the automaton will read the next letter with the other head. Each head moves to the right. When the word reaches the final state, each letter has been read once (and only once) by one of the heads. The first head is pointing at the symbol "/", the second is pointing at the last letter of v . The computation of the generating function for the number of words of \bar{B} of length $2n+1$ is thus reduced to a determinant calculus. Relation (13), and the analogous expression for the cofactor, gives an efficient way to compute these determinants. For pedagogical purpose, we give in Fig. 12 a 16-states two-head automaton accepting \bar{B} .

It is perhaps better to apply the same procedure as in Section 6, using substitution operators.

Let K be the set of pairs (w_1, w_2) of words of $\{u, v, x, a, b\}^*$ satisfying the conditions

$$(53a) \quad w = w_1 w_2, \quad w_1 \in u^* \cdot (a + xx)^*, \quad w_2 \in v^* \cdot (b + xx)^*,$$

$$(53b) \quad |w|_u = |w|_v \neq 0.$$

Taking the corresponding noncommutative generating function and applying the morphism α , we can easily write

$$(54) \quad \sum_{(w_1, w_2) \in K} \alpha(w_1)\alpha(w_2) = \frac{uv}{(1-uv)(1-a-x^2)(1-b-x^2)}.$$

As in Section 6, we apply the substitutions defined by

$$(55) \quad u \rightarrow \frac{x}{1-a}, \quad v \rightarrow \frac{x}{1-b}.$$

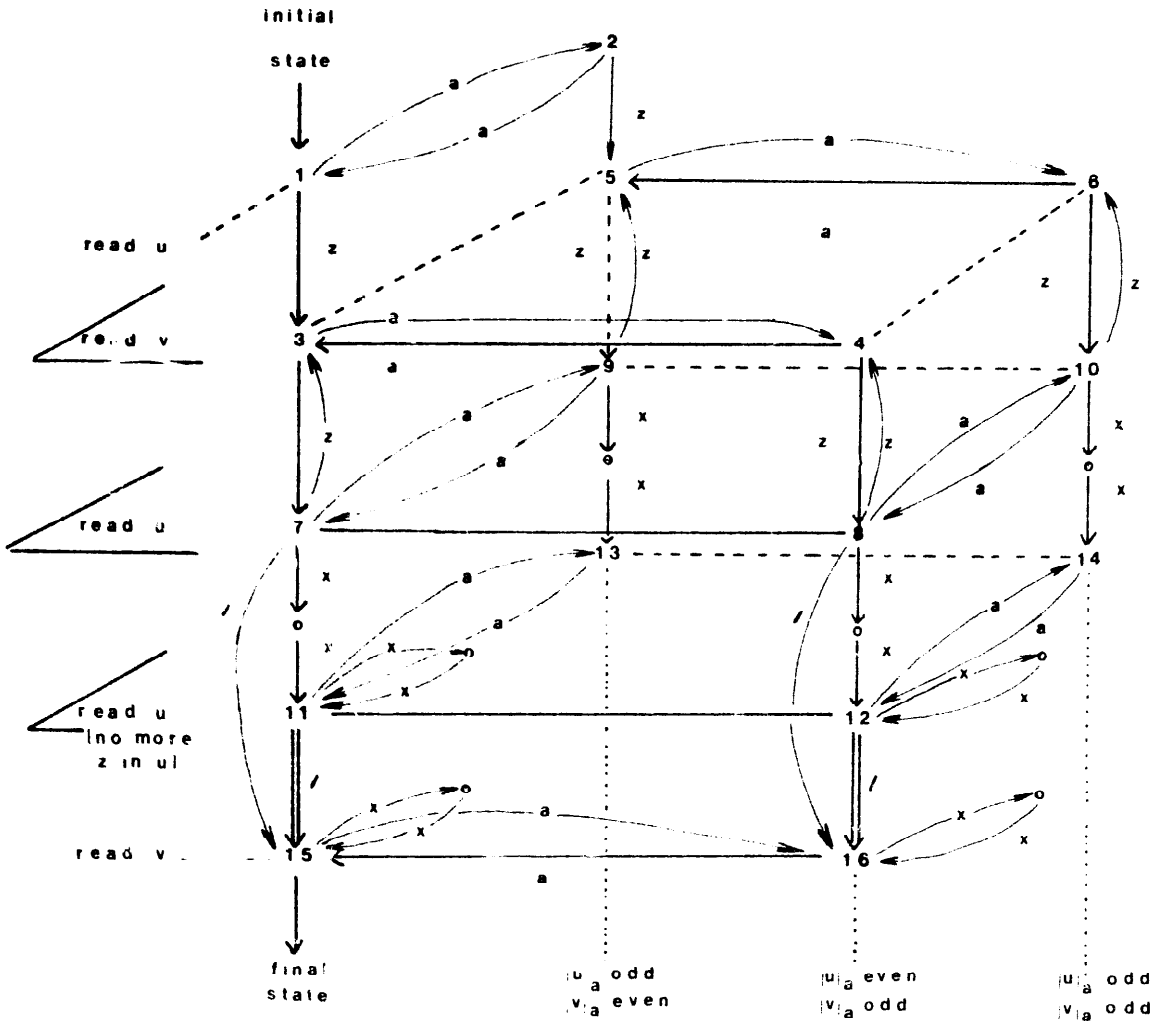


Fig. 12. The finite two-head automaton for type II polyominoes.

Let $k(u, v, x, a, b)$ be the generating function (54) and $b(z, x, a, b)$ be the generating function defined by

$$(56) \quad b(z, x, a, b) = \sum_{(w_1, w_2) \in B} \alpha(w_1)\alpha(w_2).$$

As in Section 6, we obviously have

$$(57) \quad b(t, t, t, t) = \frac{1}{4}(r(t, t, t) + 2r(t, -t, t) + r(t, -t, -t)),$$

where $r(x, a, b)$ is defined by

$$(58) \quad r(x, a, b) = k\left(\frac{x}{1-a}, \frac{x}{1-b}, a, b\right).$$

From (55) and (58) we first compute

$$(59) \quad r(x, a, b) = \frac{x^2}{((1-a)(1-b) - x^2)(1-a-x^2)(1-b-x^2)}.$$

We have successively

$$(60) \quad r(t, t, t) = \frac{t^2}{(1-2t)(1-t-t^2)^2},$$

$$(61) \quad r(t, -t, t) = \frac{t^2}{(1-2t^2)(1-t-t^2)(1+t-t^2)},$$

$$(62) \quad r(t, -t, -t) = \frac{t^2}{(1+2t)(1+t-t^2)^2}.$$

Combining (60), (61), (62) and using (57) and Proposition 7.2, we deduce the following theorem.

Theorem 8.1. *The generating function $p_{11}(t)$ for the number p_{2n}^{11} of type II convex polyominoes with perimeter $2n$ is*

$$(63) \quad \sum_{n \geq 5} p_{2n}^{11} t^{2n} = \frac{t^{10}(1-3t^2+2t^4+t^6)}{(1-2t)(1+2t)(1-2t^2)(1+t-t^2)^2(1-t-t^2)^2}.$$

9. The coding for type III polyominoes

The coding for type III polyominoes is a sort of combination of the coding for type I (Section 5) and for type II (Section 7). The coding of the middle part of the trisection (parallelogram polyomino) is exactly the same as for type I. The coding of the stack polyominoes is analogous to the construction given in Section 7 (a letter “a” encodes a West or East step, a letter “z” or a factor xx encodes a North or South step).

Nevertheless, we need to introduce some slight modifications in the mixing of the three words coding each part of the trisection, in order to obtain tractable algebraic systems. Also we choose a coding such that large parts of the computations for type I can be used again for type III.

Let Q be a type III polyomino with perimeter $2n+2$, i.e. the vertical line Δ_N supporting N is at the left-hand side of the vertical line Δ_S supporting S . The middle part between Δ_N and Δ_S is a *reverse parallelogram polyomino* P , that is, the symmetric, up to a vertical line, of a parallelogram polyomino P . Denote by (I, I') (respectively (J, J')) the segment $P \cap \Delta_S$ (respectively $P \cap \Delta_N$). Also we suppose I is below I' (respectively J above J'). The two stacks polyominoes of the trisection are necessarily nonempty. Respectively denote by SI (SJ) the right (left) stack polyomino. Let respectively I'' (J'') be the intersection of Δ_S (Δ_N) with the upper (lower) border of SI (SJ). The point I'' is between I and I' with $I \neq I''$. The point J'' is between J and J' with $J \neq J''$. Also we must have $I \neq S$ and $J \neq N$ (see Fig. 13).

Let $w = \beta^{-1}(P)$ be the Dyck word coding the parallelogram polyomino P , as defined in Section 4. Respectively denote by x^p (\bar{x}^q) the left leg (right leg) of w and $w = x^p w' x^q$. Recall that the length of the segment (I, I') (respectively (J, J')) is p (respectively q).

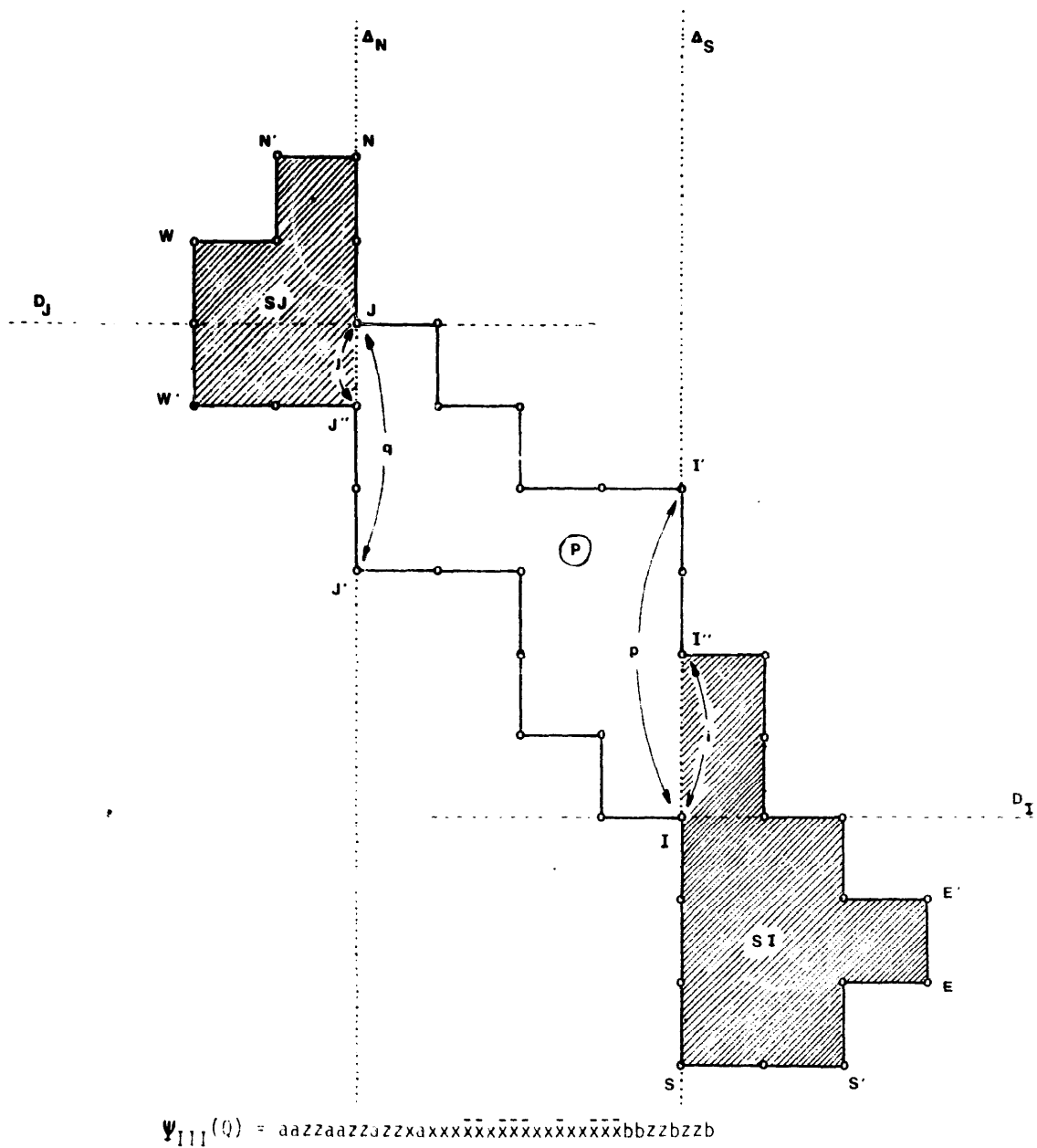


Fig. 13. The bijection Ψ_{II} .

Let D_I (respectively D_J) be the horizontal line supporting I (respectively J).

We define the word u by following counterclockwise the border of the stack polyomino SI , from S to I'' (passing through S', E and E'). Each time we make a horizontal (East or West) step we write a letter “ a ”. Each time we make a North step, we write a letter “ x ” (respectively “ z ”) if the step is above D_I (respectively below D_I).

Analogously, we define the word v by following counterclockwise (and writing v from right to left) the border of stack polyomino SJ from N to J'' , with “ a ” for an horizontal step, and “ \bar{x} ” (respectively “ \bar{z} ”) for a South step below (respectively above) D_J . Necessarily we have

$$(64) \quad |u|_a, |v|_a, |u|_x, |v|_{\bar{x}}, |u|_z, |v|_{\bar{z}} \neq 0.$$

The reader will easily prove the following proposition.

Proposition 9.1. *The map Ψ_{III} defined by (67) is a bijection between type III polyominoes with perimeter $2n+2$ and words $w \in \{x, \bar{x}, z, a, b\}^*$ of length $2n$ satisfying conditions (66a), (66c) and the following condition:*

$$(66b)' \quad w \text{ has a factorization of the form } w = afxg\bar{x}hb \text{ with } f \in \{a, zz\}^* \cdot \{a, x\}^*, \\ |f|_z \neq 0, g \in \{x, \bar{x}\}^*, h \in \{b, \bar{x}\}^* \cdot \{b, zz\}^*, |h|_z \neq 0.$$

10. The generating function for type III polyominoes

We use the same kind of techniques as for types I and II (auxiliary languages and substitution operators). Details of the proof (and MACSYMA computations!) are omitted.

We start again with the auxiliary languages H and S of Section 6, defined by the relations (28), (29) and (31). Their respective (commutative) generating functions $h(x, \bar{x}, y, \bar{y})$ and $s(x, \bar{x}, a, b)$ are given by (41) and (35). We will use the same notations of that section for $d(x, \bar{x})$, $c(t)$, $\bar{c}(t)$ and $\Delta(t)$.

Let \bar{S} be the set of words w of $\{x, \bar{x}, a, b\}^*$ defined by

$$(68) \quad w \in \bar{S} \text{ iff } w \in a^*Sb^*, \quad |w|_a \neq 0, \quad |w|_b \neq 0.$$

Let G be the (algebraic) language of $\{x, \bar{x}, y, \bar{y}, a, b\}^*$ defined by

$$(69) \quad G = a(a+y)^*y\bar{S}\bar{y}(b+\bar{y})^*b.$$

After Sections 6 and 8, the nonspecialist reader is now familiar with the methodology for translating conditions defining languages like (67), (68) and (69), into equations in noncommutative power series. From Proposition 9.1 he (or she) will easily prove that the generating function $p_{III}(t) = \sum_{n \geq 0} p_{2n}^{III} t^{2n}$ for the number p_{2n}^{III} of type III convex polyominoes is given by the following equations (we have denoted by $\bar{s}(x, \bar{x}, a, b)$ and $g(x, \bar{x}, y, \bar{y}, a, b)$ the respective commutative power series $\alpha(\bar{S})$ and $\alpha(L)$):

$$(70) \quad p_{III}(t) = \frac{1}{4}t^2(g(t, t, t^2, t^2, t, t) + 2g(t, t, t^2, t^2, -t, t) + g(t, t, t^2, t^2, -t, -t)),$$

$$(71) \quad g(x, \bar{x}, y, \bar{y}, a, b) = \frac{aby\bar{y}}{(1-a-y)(1-b-\bar{y})} \bar{s}(x, \bar{x}, a, b),$$

$$(72) \quad \bar{s}(x, \bar{x}, a, b) = \frac{1}{(1-a)(1-b)} s(x, \bar{x}, a, b) - \frac{1}{1-a} s(x, \bar{x}, a, 0) \\ - \frac{1}{(1-b)} s(x, \bar{x}, 0, b) + d(x, \bar{x}) - 1.$$

The series $p_{III}(t)$ is computed from equations (35), (41), (70), (71) and (72).

Let $g_1, g_2, g_3, G_1, G_2, G_3$ defined as follows:

$$(73) \quad g_1(x, a, b) = \frac{abx^6}{(1-a)(1-b)(1-a-x^2)(1-b-x^2)} s(x, x, a, b),$$

$$(74) \quad g_2(x, a, b) = \frac{abx^6}{(1-a)(1-a-x^2)(1-b-x^2)} s(x, x, a, 0),$$

$$(75) \quad g_3(x, a, b) = \frac{abx^6(c(x)-1)}{(1-a-x^2)(1-b-x^2)},$$

$$(76) \quad G_i(t) = \frac{1}{4}(g_i(t, t, t) + 2g_i(t, -t, t) + g_i(t, -t, -t)) \quad \text{for } i = 1, 2, 3.$$

From (70), (71) and (72) we have

$$(77) \quad p_{III}(t) = G_1(t) - 2G_2(t) + G_3(t).$$

Now using (35) and (41), we have successively

$$(78) \quad g_1(t, t, t) = \frac{t^8(1-2t+t^2-2t^4-(1-t)^2\Delta)}{2(1-2t)^2(1-t-t^2)^2},$$

$$(79) \quad g_1(t, -t, t) = \frac{t^8(-(1-2t)(1+2t)(1-3t^2)+(1-t)(1+t)(1-2t^2)\Delta)}{2(1-2t)(1+2t)(1-t-t^2)(1+t-t^2)},$$

$$(80) \quad g_1(t, -t, -t) = \frac{t^8(1+2t+t^2-2t^4-(1+t)^2\Delta)}{2(1+2t)^2(1+t-t^2)^2},$$

$$(81) \quad G_1(t) = \frac{t^8(1-8t^2+34t^4-94t^6+126t^8-54t^{10}+8t^{12})}{2(1-2t)^2(1+2t)^2(1-2t^2)(1+t-t^2)^2(1-t-t^2)^2} \\ - \frac{t^8(1-6t^2+16t^4-24t^6+20t^8-8t^{10})\Delta}{2(1-2t)^2(1+2t)^2(1-2t^2)(1+t-t^2)^2(1-t-t^2)^2},$$

$$(82a) \quad g_2(t, t, t) = \frac{t^{10}}{(1-t-t^2)^2(1-t-t^2+t^2(t-1)c)}.$$

Multiplying numerator and denominator by $(1-t-t^2+t^2(t-1)\bar{c})$ gives us

$$(82b) \quad g_2(t, t, t) = \frac{t^7(-1+t+2t^2+(1-t)\Delta)}{2(1-2t)(1-t-t^2)^2},$$

$$(83a) \quad g_2(t, -t, t) = \frac{t^{10}}{(1-t-t^2)(1+t-t^2)(-1-t+t^2+t^2(1+t)c)}.$$

Multiplying numerator and denominator by $(-1-t+t^2+t^2(1+t)\bar{c})$ gives us

$$(83b) \quad g_2(t, -t, t) = \frac{t^7(-1-t+2t^2+(1+t)\Delta)}{2(1+2t)(1+t-t^2)^2(1-t-t^2)}.$$

From

$$(84) \quad g_2(t, -t, -t) = \frac{-1+t+t^2}{1+t-t^2} g_2(t, -t, t),$$

$$(85) \quad g_2(t, t, -t) = \frac{-1+t+t^2}{1+t-t^2} g_2(t, t, t),$$

we have

$$(86a) \quad G_2(t) = \frac{t}{1+t-t^2} (g_2(t, t, t) + g_2(t, -t, t)),$$

$$(86b) \quad G_2(t) = \frac{t^8(-1+4t^2+(1-2t^2+2t^4)\Delta)}{2(1-2t)(1+2t)(1+t-t^2)^2(1-t-t^2)^2}$$

For $G_3(t)$, we have successively

$$(87) \quad g_3(t, t, t) = \frac{t^6(1-2t^2-\Delta)}{2(1-t-t^2)^2},$$

$$(88) \quad g_3(t, -t, t) = \frac{t^6(-1+2t^2+\Delta)}{2(1+t-t^2)(1-t-t^2)},$$

$$(89) \quad g_3(t, -t, -t) = \frac{t^6(1-2t^2-\Delta)}{2(1+t-t^2)^2},$$

$$(90) \quad G_3(t) = \frac{t^8(1-2t^2-\Delta)}{2(1-t-t^2)^2(1+t-t^2)^2}.$$

Combining (81), (86b) and (90) according to (77), we have the following theorem.

Theorem 10.1. *The generating function for the number p_{2n}^{III} of type III convex polyominoes with perimeter $2n$ is*

$$(91) \quad \sum_{n \geq 6} p_{2n}^{III} t^{2n} = \frac{t^8(2-20t^2+75t^4-127t^6+95t^8-27t^{10}+4t^{12})}{(1-2t)^2(1+2t)^2(1-2t^2)(1+t-t^2)^2(1-t-t^2)^2} - 2t^8(1-4t^2)^{-3/2}.$$

11. The number of convex polyominoes

The generating function $p(t)$ for the number p_{2n} of convex polyominoes with perimeter $2n$ is obtained by adding the three generating functions $p_I(t)$, $p_{II}(t)$, $p_{III}(t)$ given in Theorems 6.3, 8.1 and 10.1. We thus restore the initial symmetries of the convex polyomino and dramatic simplifications occur. Using again MACSYMA, we obtain the following theorem.

Theorem 11.1. *The generating function for the number p_{2n} of convex polyominoes with perimeter $2n$ is*

$$(92) \quad \sum_{n \geq 2} p_{2n} t^{2n} = \frac{t^4(1 - 6t^2 + 11t^4 - 4t^6)}{(1 - 4t^2)^2} - 4t^8(1 - 4t^2)^{-3/2}.$$

The surprising fact is that the inverses of the zeros of the denominator of the rational fraction are integers.

We can expand the generating function (92). The rational fraction gives successively

$$(93) \quad \frac{1}{(1 - 4t^2)^2} = \sum_{n \geq 0} (n + 1)4^n t^{2n},$$

$$(94) \quad \frac{t^4(1 - 6t^2 + 11t^4 - 4t^6)}{(1 - 4t^2)^2} = t^4 + 2t^6 + \sum_{n \geq 0} (2n + 1)4^n t^{2n+8}.$$

The algebraic part is expanded as follows,

$$(95) \quad (1 - 4t^2)^{-3/2} = \sum_{n \geq 0} \frac{1}{2^n} \frac{1 \times 3 \times \dots \times (2n + 1)}{n!} 4^n t^{2n} \\ = \sum_{n \geq 0} (2n + 1) \binom{2n}{n} t^{2n}.$$

Subtracting (95) times $4t^8$ from (94) we have proved our formula given in Theorem 1.1:

$$(96) \quad p_{2n+8} = (2n + 1)4^n - 4(2n + 1) \binom{2n}{n}.$$

12. Conclusion and final remarks

(1) We have solved an open problem in enumerative combinatorics by using algebraic languages methodology, while tentatives using 'classical' analytic techniques have not yet succeeded. Although the final formula is surprisingly simple, so far this is the only proof we know.

(2) We believe that one of the interests of this method lies in the confrontation between two different points of view. Combinatorists are looking for enumeration formulae while algebraic language theorists are motivated by Computer Science considerations. In particular, the combinatorist is looking at the languages up to a commutation of the letters, and may consider languages not necessarily algebraic, but being in bijection with an algebraic (or even rational) language. *Multi-head push-down automata* are of special use to produce such bijections.

Such considerations are illustrated by Cori et al. [7] where *shuffle* of two Dyck words $u \in \{x, \bar{x}\}^*$ and $v \in \{y, \bar{y}\}^*$ is changed bijectively into a pair of two Dyck words.

Note that these commutation considerations are different from the theoretical concepts of commutative languages introduced by Latteux [25].

(3) In this paper, there is a constant interplay between the bijections, the languages and the computations (in commutative variables). Other codings (especially for types II and III) would be possible, using the main idea of the two bijections Φ and β for stack and parallelogram polyominoes. A more direct bijection can be explicated for type II, but with algebraic languages that are not linear. The bijection we gave in Section 7 has the advantage to produce a language easy to define and leading to an ultimate coding with words of a rational language. Slight modifications of a bijection can make the language not algebraic.

(4) M.-P. Delest has produced a direct extension of the bijection β for row-convex polyominoes (see Section 1). This bijection gives an algebraic generating function for the number of row-convex polyominoes according to the perimeter [10]. (Remark that conversely to the case of convex polyominoes, the enumeration of row-convex polyominoes according to the perimeter is much more difficult than enumeration according to the area.) It is possible to characterize in this coding polyominoes that also are column-convex. Thus it would be possible to give a unique algebraic language coding all convex polyominoes. Unfortunately, the definition of the language is much more complicated than the one of the three languages introduced here, and the corresponding computations seem to be much longer.

(5) It would be easy to define a unique algebraic language (a slight extension of the pigmented Dyck language) such that the three algebraic languages coding each type of polyominoes are contained in that language.

(6) It would be possible to write down an algebraic grammar defining the languages for types I and III such that the commutative corresponding algebraic systems can be solved by a succession of linear systems of equations, in term of the generating function $c(t)$ of the Dyck language. Roughly speaking, the algebraicity has been 'concentrated' in the Dyck language. The generating functions $p_I(t)$ and $p_{III}(t)$ are rational expressions in term of t and $c(t)$. This fact was crucial for solving the systems. The corresponding languages are in a certain sense (containing the two concepts 'commutativity' and 'rationality') close to the Dyck language. Note that the Motzkin language of Section 3 does not have this property. A theoretical investigation about such languages would be probably of interest.

We now make some final remarks from the combinatorial point of view.

(7) First, the next step for future work would be to give a direct (bijective) proof of the formula for p_{2n} . Things are completely different once an exact formula is known!

Note that a direct classical approach would be to define a convex polyomino as a 4-tuple of paths: take the part of the border going clockwise from S to W' (respectively W to N , N to E' , E to S'). If these 8 points (notice some can be not distinct) are fixed on each side of a rectangle R , then the total number of such 4-tuples is a product of four binomial coefficients. But some paths can intersect and the configuration does not correspond to a polyomino. Taking the difference between two products of four binomial coefficients gives (from the Gessel-Viennot methodology [13]) the right number of polyominoes. Thus p_{2n} is obtained by taking the difference of two sums (for all points, and all rectangles of perimeter $2n$) of products of four binomial coefficients. The difference of two numbers giving p_{2n} is not the same as the difference of the formula of Theorem 1.1. Using the bijection β and formula (24), it would be possible to reduce the sum to products of three binomial coefficients.

(8) A major problem would be to introduce the area of the polyomino in our computation. This has been done for stack polyominoes [42] and parallelogram polyominoes [30, 12, 18]. The area is easily determined from the word coding the polyomino. The problem is to make a q -analog of what we have done.

(9) Notice that since this work was completed, we have solved two other problems using the same methodology: the number of the so-called *directed lattice animals* introduced recently in statistical physics (see [40]) and the number of *secondary structures* of single-stranded nucleic acids (i.e., RNAs, ...) having a given complexity in biology (see [39]). With the coding of Cori and Vauquelin [8] for planar maps, we have four examples of a coding of a planar 'picture' with words of an algebraic language. It is intriguing that the four codings use languages 'close' to the Motzkin and Dyck languages.

(10) In conclusion, convex polyominoes enumeration is an example of a combinatorial problem solved by 'bijective' methods. The irony is that the solution ends with tedious computations.

Acknowledgment

We thank R. Cori for valuable suggestions about operators and auxiliary languages, and H. Cohen and G. Kahn for MACSYMA help.

References

- [1] E. Bender, Convex n -ominoes, *Discrete Math.* **8** (1974) 31-40.
- [2] C. Berge, C.C. Chen, V. Chvátal and C.S. Seow, Combinatorial properties of polyominoes, *Combinatorica* **1** (1981) 217-224.

- [3] J. Berstel, *Transduction and Context-free Languages* (Teubner, Stuttgart, 1979).
- [4] L. Boasson, Langages algébriques, paires itérantes et transductions rationnelles, *Theoret. Comput. Sci.* **2** (1976) 209–223.
- [5] N. Chomsky and M.P. Schützenberger, The algebraic theory of context-free languages, in: P. Braffort and D. Hirschberg, eds., *Computer Programming and Formal Systems* (North-Holland, Amsterdam, 1963).
- [6] R. Cori, *Un Code pour les Graphes Planaires et ses Applications*, Astérisque **27** (Soc. Math. France, 1975).
- [7] R. Cori, S. Dulucq and G. Viennot, Shuffle of parenthesis systems and Baxter permutations, *J. Combinatorial Theory Serie A*, submitted.
- [8] R. Cori and B. Vaquelin, Planar maps are well labeled trees, *Canad. J. Math.* **33** (1981) 1023–1042.
- [9] S. Chaiken, D.J. Kleitman, M. Saks and J. Shearer, Covering regions by rectangles, *SIAM J. Discrete and Algebraic Methods* **2** (1981) 394–410.
- [10] M.-P. Delest, Generating function for column-convex polyominoes, submitted.
- [11] M. Eden, A two-dimensional growth process, *Proc. 4th Berkeley Symp. on Mathematical Statistics and Probability, IV* (Univ. of California Press, Berkeley, CA, 1961) pp. 223–239.
- [12] J. Gessel, A noncommutative generalization and q -analog of the Lagrange inversion formula, *Trans. Amer. Math. Soc.* **257** (1980) 455–482.
- [13] J. Gessel and G. Viennot, Binomial determinants, paths and Hook lengths formulae, Preprint, MIT, 1983.
- [14] S. Ginsburg, *The Mathematical Theory of Context-free Languages* (McGraw-Hill, New York, 1966).
- [15] J. Goldman, Formal languages and enumeration, *J. Combinatorial Theory Serie A* **24** (1978) 318–338.
- [16] S. Golomb, *Polyominoes* (Scribner, New York, 1965).
- [17] M. Gross, Applications géométriques des langages formels, *I.C.C. Bulletin* **5** (1961) 141–168.
- [18] J. Hofbauer and J. Furlinger, q -Catalan numbers, Preprint, 1983.
- [19] D. Klarner, Some results concerning polyominoes, *Fibonacci Quart.* **3** (1965) 9–20.
- [20] D. Klarner, Cell growth problems, *Canad. J. Math.* **19** (1967) 851–863.
- [21] D. Klarner and R. Rivest, Asymptotic bounds for the number of convex n -ominoes, *Discrete Math.* **8** (1974) 31–40.
- [22] G. Kreweras, Sur les éventails de segments, *Cahiers du B.U.R.O.* **15** (1970) 1–41.
- [23] W. Kuich, Enumeration problems and context free languages, in: *Combinatorial Theory and its Applications*, Colloquia Mathematica (Societatis Janos Bolyai, Hongrie, 1975) pp. 729–735.
- [24] W. Kuich, A context-free language and enumeration problems on infinite trees and diagraphs, *J. Combinatorial Theory* **10** (1971) 135–142.
- [25] M. Latteux, Cones rationnels commutatifs, *J. Comput. Systems Sci.* **18** (1979) 307–333.
- [26] W. Masek, Personal communication quoted in [9], 1979.
- [27] S. Miyano, Remark on multihead pushdown automata and multihead stack automata, *J. Comput. System Sci.* **27** (1983) 116–124.
- [28] M. Nivat, Séries formelles algébriques, in: J. Berstel, ed., *Séries Formelles en Variables Non Commutative et Applications*. Actes de la 5ème Journée de Printemps d'Informatique Théorique, Vieux-Boucau les Bains, 1977 (LITP and ENSTA, Paris, 1978) pp. 219–230.
- [29] W. Ogden, A helpful result for proving inherent ambiguity, *Math. Syst. Theory* **2** (1968) 191–194.
- [30] G. Polya, On the number of certain lattice polygons, *J. Combinatorial Theory* **6** (1969) 102–105.
- [31] R.C. Read, Contributions to the cell growth problem, *Canad. J. Math.* **14** (1962) 1–20.
- [32] A.I. Rosenberg, On multihead finite automata, *IBM J. Res. Develop.* **10** (1966) 388–394.
- [33] A. Salomaa and M. Soittola, *Automata-theoretic Aspects of Formal Power Series* (Springer, New York/Berlin, 1978).
- [34] M.P. Schützenberger, Certain elementary families of automata, in: *Proc. Symp. on Mathematical Theory of Automata* (Polytechnic Institute of Brooklyn, 1962) pp. 139–153.
- [35] M.P. Schützenberger, Context-free languages and pushdown automata, *Information and Control* **6** (1963) 246–264.
- [36] L.W. Shapiro and D. Zeilberger, A Markov chain occurring in enzyme kinetics, *J. Math. Biology*, submitted.
- [37] W.T. Tutte, A census of planar maps, *Canad. J. Math.* **15** (1963) 249–271.
- [38] J. van Leeuwen, Periodic storage schemes with minimum number of memory banks, in: *Proc. "Workshop on Graph-theoretic Concepts in Computer Science"*, Haus Ohrbeck, 1983 (Hauser, München, to be published).

- [39] M. Vauchaussade de Chaumont and G. Viennot, Enumeration of RNAs by complexity, *Internat. Conf. on Medicine and Biology*, Bari, Italie, 1983.
- [40] G. Viennot, Problèmes combinatoires posés par la physique statistique, Exposé No. 626, in: *Séminaire Bourbaki 1983/1984; Astérisque*, to appear.
- [41] G. Viennot, Up-down sequence of trees and parallelogram polyominoes, In preparation.
- [42] E.M. Wright, Stacks, *Quart. J. Math. Oxford (2)* **19** (1968) 313-320.
- [43] A.C. Yao and R.L. Rivest, $k + 1$ heads are better than k , *J. Assoc. Comput. Mach.* **25** (1980) 337-340.