

Primitive Dyck words as an invariant class for pattern avoidance

S.Bilotta E. Grazzini E.Pergola

Dipartimento di Sistemi e Informatica
Università degli Studi di Firenze

GASCom 2012

Table of contents

- 1 Definitions and notations
 - Binary words avoiding a pattern p
 - Primitive Dyck words
 - Succession rules
- 2 Motivations
- 3 Main Result
 - The constructive algorithm
 - Proof: idea
- 4 Enumeration
- 5 Generalization
- 6 Further developments

Binary words avoiding a pattern p

Let $F \subset \{0, 1\}^*$ be the class of binary words ω such that $|\omega|_0 \leq |\omega|_1$ for any $\omega \in F$, $|\omega|_0$ and $|\omega|_1$ are the number of zeroes and ones in ω , respectively.

We are interested in studying the subclass $F^{[p]} \subset F$ of binary words excluding a given pattern $p = p_0 \dots p_{h-1} \in \{0, 1\}^h$, i.e. the words $\omega \in F^{[p]}$ that do not admit a sequence of consecutive indices $i, i+1, \dots, i+h-1$ such that

$$\omega_i \omega_{i+1} \dots \omega_{i+h-1} = p_0 p_1 \dots p_{h-1}.$$

- R. Sedgewick & P. Flajolet (1996): *An Introduction to the Analysis of Algorithms*. Addison-Wesley, Reading, MA.

Binary words avoiding a pattern p

Let $F \subset \{0, 1\}^*$ be the class of binary words ω such that $|\omega|_0 \leq |\omega|_1$ for any $\omega \in F$, $|\omega|_0$ and $|\omega|_1$ are the number of zeroes and ones in ω , respectively.

We are interested in studying the subclass $F^{[p]} \subset F$ of binary words excluding a given pattern $p = p_0 \dots p_{h-1} \in \{0, 1\}^h$, i.e. the words $\omega \in F^{[p]}$ that do not admit a sequence of consecutive indices $i, i+1, \dots, i+h-1$ such that

$$\omega_i \omega_{i+1} \dots \omega_{i+h-1} = p_0 p_1 \dots p_{h-1}.$$

- R. Sedgewick & P. Flajolet (1996): *An Introduction to the Analysis of Algorithms*. Addison-Wesley, Reading, MA.

Binary words avoiding a pattern p

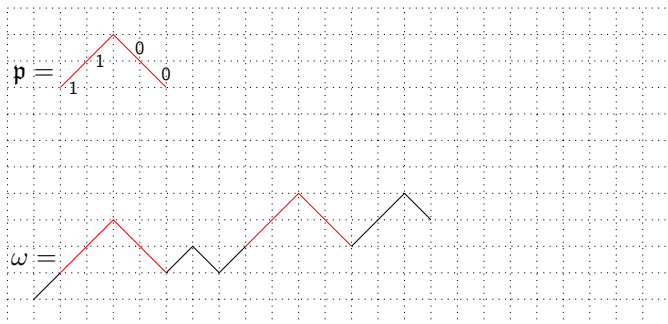
Let $F \subset \{0, 1\}^*$ be the class of binary words ω such that $|\omega|_0 \leq |\omega|_1$ for any $\omega \in F$, $|\omega|_0$ and $|\omega|_1$ are the number of zeroes and ones in ω , respectively.

We are interested in studying the subclass $F^{[p]} \subset F$ of binary words excluding a given pattern $p = p_0 \dots p_{h-1} \in \{0, 1\}^h$, i.e. the words $\omega \in F^{[p]}$ that do not admit a sequence of consecutive indices $i, i+1, \dots, i+h-1$ such that

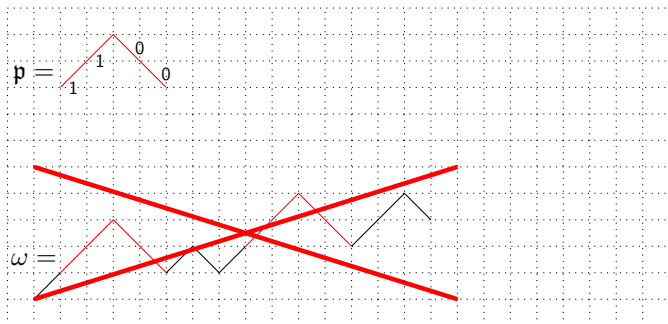
$$\omega_i \omega_{i+1} \dots \omega_{i+h-1} = p_0 p_1 \dots p_{h-1}.$$

- R. Sedgewick & P. Flajolet (1996): *An Introduction to the Analysis of Algorithms*. Addison-Wesley, Reading, MA.

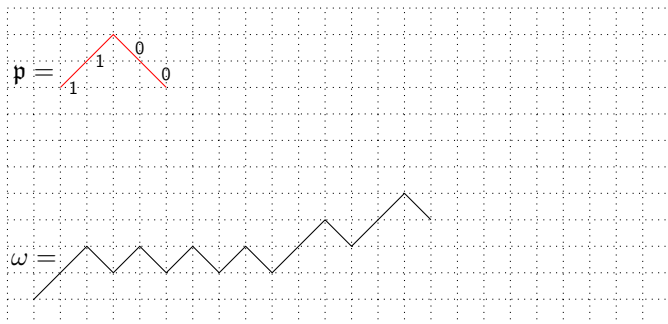
Example



Example



Example



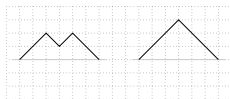
Primitive Dyck words

Let us denote by \mathcal{D}_j^p the set of $2j$ -length primitive Dyck words. We study the class $F^{[p]}$ where $\mathfrak{p} \in \mathcal{D}_j^p$, for any fixed $j \geq 1$.

It is well known that a Dyck word δ is a binary string such that $|\delta|_0 = |\delta|_1$ and no prefix of δ has more zeroes than ones. A path associated to a Dyck word (that is a *Dyck path*) lies above, but may intersect, the x -axis.



In this work, we consider the so called *primitive* Dyck paths, that is Dyck paths which intersect the x -axis at the beginning and at the end, exclusively.



Succession rules

A *succession rule* Ω is a system constituted by an *axiom* (a) , with $a \in \mathbb{N}$, and a set of *productions* of the form:

$$(k) \rightsquigarrow (e_1(k))(e_2(k)) \dots (e_k(k)), \quad k \in \mathbb{N}, \quad e_j : \mathbb{N} \rightarrow \mathbb{N}.$$

A production constructs, for any given label (k) , its *successors* $(e_1(k)), (e_2(k)), \dots, (e_k(k))$.

Compact notation:

$$\left\{ \begin{array}{l} (a) \\ (k) \rightsquigarrow (e_1(k))(e_2(k)) \dots (e_k(k)) \end{array} \right.$$

- F. R. K. Chung, R. L. Graham, V. E. Hoggatt & M. Kleimann (1978): *The number of Baxter permutations*. *Journal of Combinatorial Theory, Series A* 24, pp. 382-394.

Succession rules

A *succession rule* Ω is a system constituted by an *axiom* (a) , with $a \in \mathbb{N}$, and a set of *productions* of the form:

$$(k) \rightsquigarrow (e_1(k))(e_2(k)) \dots (e_k(k)), \quad k \in \mathbb{N}, \quad e_j : \mathbb{N} \rightarrow \mathbb{N}.$$

A production constructs, for any given label (k) , its *successors* $(e_1(k)), (e_2(k)), \dots, (e_k(k))$.

Compact notation:

$$\left\{ \begin{array}{l} (a) \\ (k) \rightsquigarrow (e_1(k))(e_2(k)) \dots (e_k(k)) \end{array} \right.$$

- F. R. K. Chung, R. L. Graham, V. E. Hoggatt & M. Kleimann (1978): *The number of Baxter permutations*. *Journal of Combinatorial Theory, Series A* 24, pp. 382-394.

Succession rules

A *succession rule* Ω is a system constituted by an *axiom* (a) , with $a \in \mathbb{N}$, and a set of *productions* of the form:

$$(k) \rightsquigarrow (e_1(k))(e_2(k)) \dots (e_k(k)), \quad k \in \mathbb{N}, \quad e_j : \mathbb{N} \rightarrow \mathbb{N}.$$

A production constructs, for any given label (k) , its *successors* $(e_1(k)), (e_2(k)), \dots, (e_k(k))$.

Compact notation:

$$\left\{ \begin{array}{l} (a) \\ (k) \rightsquigarrow (e_1(k))(e_2(k)) \dots (e_k(k)) \end{array} \right.$$

- F. R. K. Chung, R. L. Graham, V. E. Hoggatt & M. Kleimann (1978): *The number of Baxter permutations*. Journal of Combinatorial Theory, Series A 24, pp. 382-394.

Generating trees

The rule Ω can be represented by means of a *generating tree*, that is a rooted tree whose vertices are the labels of Ω ; where (a) is the label of the root and each node labelled (k) has k sons labelled $(e_1(k)), (e_2(k)), \dots, (e_k(k))$, respectively. As usual, the root lies at level 0, and a node lies at level n if its parent lies at level $n - 1$.

If a succession rule describes the growth of a class of combinatorial objects, then a given object can be coded by the sequence of labels met from the root of the generating tree to the object itself.

Generating trees

The rule Ω can be represented by means of a *generating tree*, that is a rooted tree whose vertices are the labels of Ω ; where (a) is the label of the root and each node labelled (k) has k sons labelled $(e_1(k)), (e_2(k)), \dots, (e_k(k))$, respectively. As usual, the root lies at level 0, and a node lies at level n if its parent lies at level $n - 1$.

If a succession rule describes the growth of a class of combinatorial objects, then a given object can be coded by the sequence of labels met from the root of the generating tree to the object itself.

Jumping succession rules

A *jumping succession rule* is a set of productions acting on the objects of a class and producing sons at different levels.

Compact notation:

$$\left\{ \begin{array}{l} (a) \\ (k) \xrightarrow{1} (e_1(k))(e_2(k)) \dots (e_k(k)), \\ (k) \xrightarrow{j} (d_1(k))(d_2(k)) \dots (d_k(k)). \end{array} \right.$$

- L. Ferrari, E. Pergola, R. Pinzani & S. Rinaldi (2003): *Jumping succession rules and their generating functions*. Discrete Mathematics 271, pp. 29-50.

Jumping succession rules

A *jumping succession rule* is a set of productions acting on the objects of a class and producing sons at different levels.

Compact notation:

$$\left\{ \begin{array}{l} (a) \\ (k) \xrightarrow{1} (e_1(k))(e_2(k)) \dots (e_k(k)), \\ (k) \xrightarrow{j} (d_1(k))(d_2(k)) \dots (d_k(k)). \end{array} \right.$$

- L. Ferrari, E. Pergola, R. Pinzani & S. Rinaldi (2003): *Jumping succession rules and their generating functions*. Discrete Mathematics 271, pp. 29-50.

Jumping succession rules

A *jumping succession rule* is a set of productions acting on the objects of a class and producing sons at different levels.

Compact notation:

$$\left\{ \begin{array}{l} (a) \\ (k) \xrightarrow{1} (e_1(k))(e_2(k)) \dots (e_k(k)), \\ (k) \xrightarrow{j} (d_1(k))(d_2(k)) \dots (d_k(k)). \end{array} \right.$$

- L. Ferrari, E. Pergola, R. Pinzani & S. Rinaldi (2003): *Jumping succession rules and their generating functions*. Discrete Mathematics 271, pp. 29-50.

Marked jumping succession rules

A *marked jumping succession rule* is a jumping succession rule where *marked* labels are considered together with usual ones. In this way a generating tree can support negative values if we consider a node labelled (\bar{k}) as opposed to a node labelled (k) lying on the same level.

Compact notation:

$$\left\{ \begin{array}{l} (a) \\ (k) \xrightarrow{1} (e_1(k))(e_2(k)) \dots (e_k(k)), \\ (k) \xrightarrow{j} (\overline{d_1(k)})(\overline{d_2(k)}) \dots (\overline{d_k(k)}). \end{array} \right.$$

Marked jumping succession rules

A *marked jumping succession rule* is a jumping succession rule where *marked* labels are considered together with usual ones. In this way a generating tree can support negative values if we consider a node labelled (\bar{k}) as opposed to a node labelled (k) lying on the same level.

Compact notation:

$$\left\{ \begin{array}{l} (a) \\ (k) \xrightarrow{1} (e_1(k))(e_2(k)) \dots (e_k(k)), \\ (k) \xrightarrow{j} (\overline{d_1(k)})(\overline{d_2(k)}) \dots (\overline{d_k(k)}). \end{array} \right.$$

Marked jumping succession rules

Compact notation:

$$\left\{ \begin{array}{l} (a) \\ (k) \xrightarrow{1} (e_1(k))(e_2(k)) \dots (e_k(k)), \\ (k) \xrightarrow{j} (\overline{d_1(k)})(\overline{d_2(k)}) \dots (\overline{d_k(k)}). \end{array} \right.$$

This describes also the rules for (\overline{k}) :

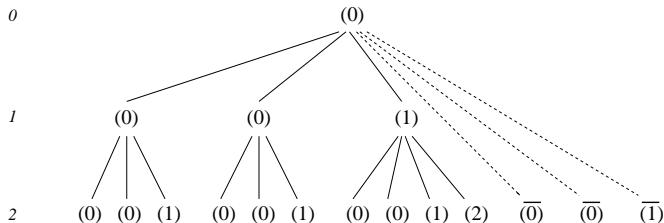
$$\left\{ \begin{array}{l} (\overline{k}) \xrightarrow{1} (\overline{e_1(k)})(\overline{e_2(k)}) \dots (\overline{e_k(k)}), \\ (\overline{k}) \xrightarrow{j} (d_1(k))(d_2(k)) \dots (d_k(k)). \end{array} \right.$$

We remark that $(\overline{\overline{k}}) = (k)$.

Example

$$\left\{ \begin{array}{l} (0) \\ (k) \xrightarrow{1} (k+1)(k) \cdots (1)(0)(0) \quad k \geq 0 \\ (k) \xrightarrow{2} \overline{(k+1)}(\overline{k}) \cdots (\overline{1})(\overline{0})(\overline{0}) \quad k \geq 0 \end{array} \right.$$

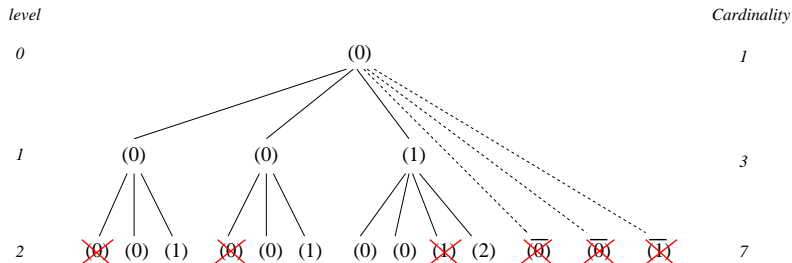
level



$$\overline{\overline{(k)}} = (k)$$

Example

$$\left\{ \begin{array}{l} (0) \\ (k) \xrightarrow{1} (k+1)(k)\cdots(1)(0)(0) \quad k \geq 0 \\ (k) \xrightarrow{2} \overline{(k+1)}\overline{(k)}\cdots\overline{(1)}\overline{(0)}\overline{(0)} \quad k \geq 0 \end{array} \right.$$



$$\overline{\overline{(k)}} = (k)$$

Motivations

Let $F \subset \{0, 1\}^*$ be the class of binary words w such that $|w|_0 \leq |w|_1$ for any $w \in F$.

For example, in order to generate the language $F^{[p]}$ an “ad hoc” grammar (depending on the forbidden pattern p) should be defined. Consequently, for each pattern p a different generating function enumerating the words in $F^{[p]}$ must to be computed.

Our aim is to determine a constructive algorithm proposing a more unified approach. The algorithm bases on a succession rule which allows us to obtain a parametric generating function for the enumeration of each classes $F^{[p]}$, according to the number of ones. Such enumeration does not depend on the shape of the forbidden pattern p .

Theorem

Theorem

The generating tree of the paths in $F^{[p]}$, where $p \in \mathcal{D}_j^p$, according to the number of rise steps, is isomorphic to the tree having its root labelled (0) and recursively defined by the succession rule:

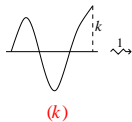
$$\left\{ \begin{array}{l} (k) \xrightarrow{1} (k+1)(k) \dots (1)(0_2)(0_1) \quad k \geq 0 \\ (0) \xrightarrow{j} (\overline{0_2}) \\ (k) \xrightarrow{j} (\overline{k})(\overline{k-1}) \dots (\overline{1})(\overline{0_2})(\overline{0_1}) \quad k \geq 1 \end{array} \right.$$

The constructive algorithm: First step

$$(k) \rightsquigarrow (k+1) (k) \dots (1) (0_2) (0_1)$$

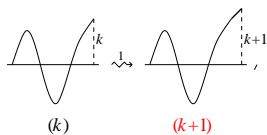
The constructive algorithm: First step

$$(k) \rightsquigarrow (k+1)(k) \dots (1)(0_2)(0_1)$$



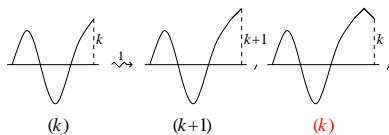
The constructive algorithm: First step

$$(k) \rightsquigarrow (k+1) (k) \dots (1) (0_2) (0_1)$$



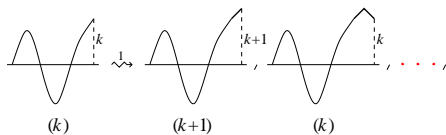
The constructive algorithm: First step

$$(k) \rightsquigarrow (k+1) (k) \dots (1) (0_2) (0_1)$$



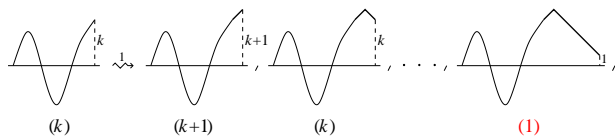
The constructive algorithm: First step

$$(k) \rightsquigarrow (k+1) (k) \dots (1) (0_2) (0_1)$$



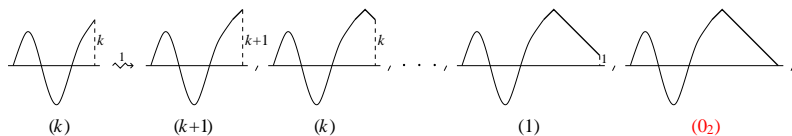
The constructive algorithm: First step

$$(k) \rightsquigarrow (k+1) (k) \dots (1) (0_2) (0_1)$$



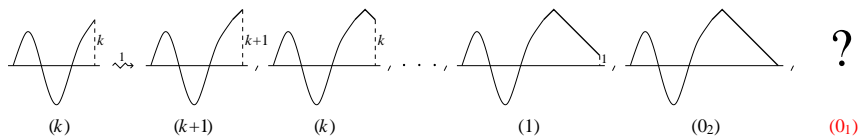
The constructive algorithm: First step

$$(k) \rightsquigarrow (k+1)(k) \dots (1) (0_2)(0_1)$$



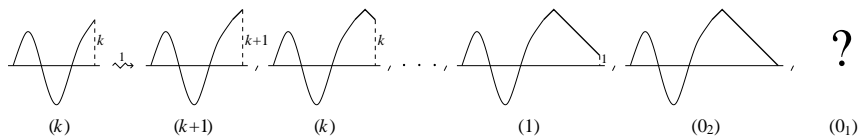
The constructive algorithm: First step

$$(k) \rightsquigarrow (k+1)(k) \dots (1)(0_2)(0_1)$$



The constructive algorithm: First step

$$(k) \rightsquigarrow (k+1) (k) \dots (1) (0_2) (0_1)$$

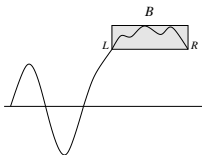


The constructive algorithm: a marked forbidden pattern

We define a *marked forbidden pattern* \mathfrak{p} as a pattern $\mathfrak{p} \in \mathcal{D}_j^P$ whose steps can not be divided, they must lie always in that defined sequence.

We say that a point is strictly contained in a marked forbidden pattern if it is between two steps of the forbidden pattern itself.

We denote a marked forbidden pattern \mathfrak{p} by drawing its minimal bounding rectangle B .

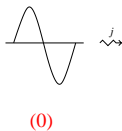


The constructive algorithm: First step

$$(0) \rightsquigarrow^j \overline{(0_2)}$$

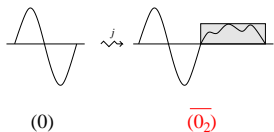
The constructive algorithm: First step

$$(0) \rightsquigarrow^j \overline{(0_2)}$$



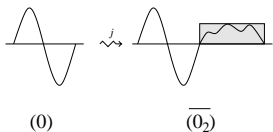
The constructive algorithm: First step

$$(0) \rightsquigarrow^j \overline{(0_2)}$$



The constructive algorithm: First step

$$(0) \rightsquigarrow^j \overline{(0_2)}$$

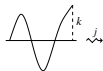


The constructive algorithm: First step

$$(k) \rightsquigarrow (\overline{k}) \overline{(k-1)} \cdots \overline{(1)} \overline{(0_2)} \overline{(0_1)}$$

The constructive algorithm: First step

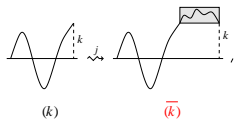
$$(k) \rightsquigarrow (\overline{k}) (\overline{k-1}) \cdots (\overline{1}) (\overline{0_2}) (\overline{0_1})$$



(k)

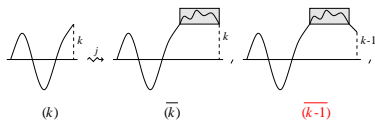
The constructive algorithm: First step

$$(k) \rightsquigarrow \overline{(k)} \overline{(k-1)} \cdots \overline{(1)} \overline{(0_2)} \overline{(0_1)}$$



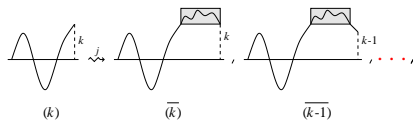
The constructive algorithm: First step

$$(k) \rightsquigarrow \overline{(k)} \overline{(k-1)} \cdots \overline{(1)} \overline{(0_2)} \overline{(0_1)}$$



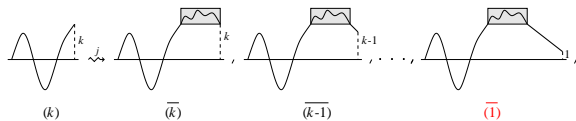
The constructive algorithm: First step

$$(k) \rightsquigarrow (\overline{k}) (\overline{k-1}) \cdots (\overline{1}) (\overline{0_2}) (\overline{0_1})$$



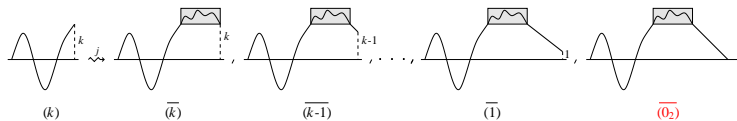
The constructive algorithm: First step

$$(k) \rightsquigarrow (\bar{k}) (\overline{k-1}) \cdots (\bar{1}) (\overline{0_2}) (\overline{0_1})$$



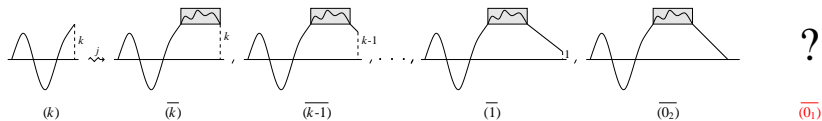
The constructive algorithm: First step

$$(k) \rightsquigarrow (\bar{k}) (\overline{k-1}) \cdots (\bar{1}) (\overline{0_2}) (\overline{0_1})$$



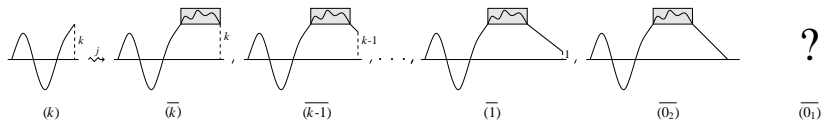
The constructive algorithm: First step

$$(k) \rightsquigarrow (\bar{k}) (\overline{k-1}) \cdots (\bar{1}) (\overline{0_2}) (\overline{0_1})$$



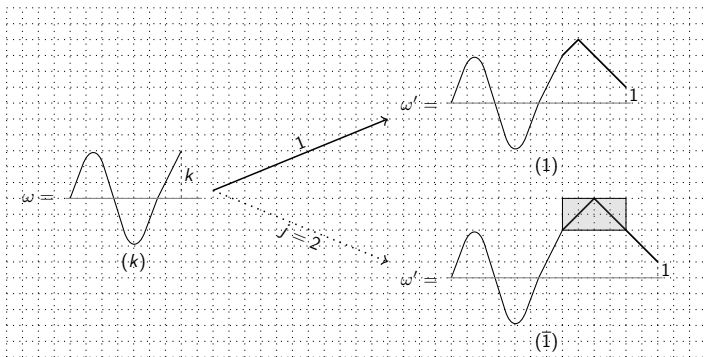
The constructive algorithm: First step

$$(k) \rightsquigarrow (\bar{k}) (\overline{k-1}) \cdots (\bar{1}) (\overline{0_2}) (\overline{0_1})$$



The constructive algorithm: Second step

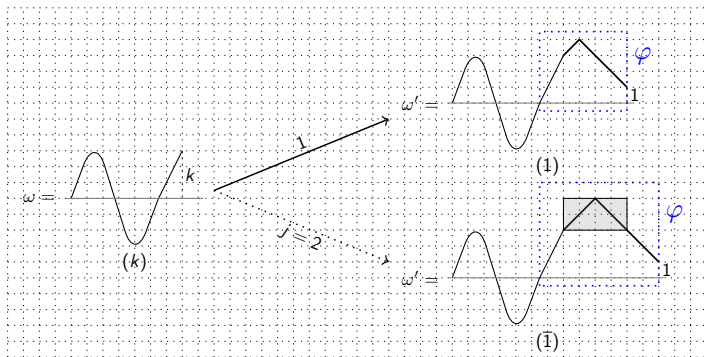
First action in order to obtain the label (0_1)



$\omega' = v\varphi$, being φ is the rightmost suffix in ω' beginning from the x -axis and strictly remaining above the x -axis.

The constructive algorithm: Second step

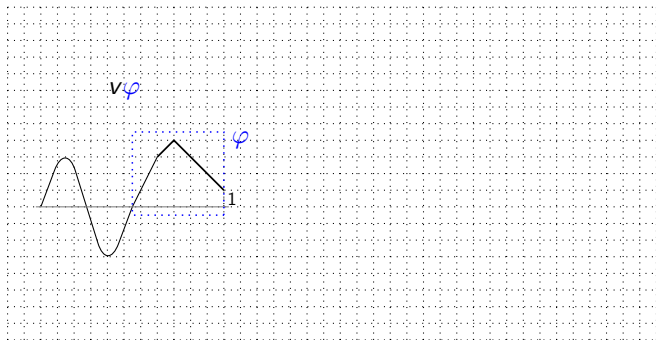
First action in order to obtain the label (0_1)



$\omega' = v\varphi$, being φ is the rightmost suffix in ω' beginning from the x -axis and strictly remaining above the x -axis.

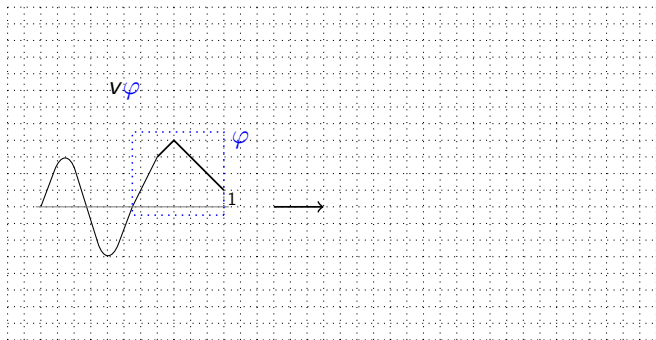
The constructive algorithm: Second step - case 1

φ does not contain any marked forbidden pattern



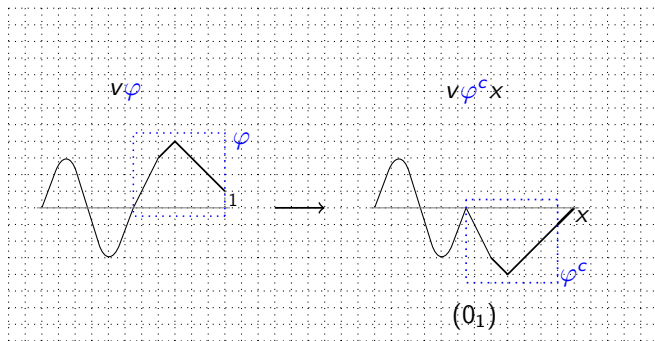
The constructive algorithm: Second step - case 1

φ does not contain any marked forbidden pattern



The constructive algorithm: Second step - case 1

φ does not contain any marked forbidden pattern



The constructive algorithm: Second step - case 2

φ contains at least one marked forbidden pattern

- Find z the leftmost point in φ having highest ordinate and not strictly contained in a marked forbidden pattern.
- Apply the *cut and paste* operation.

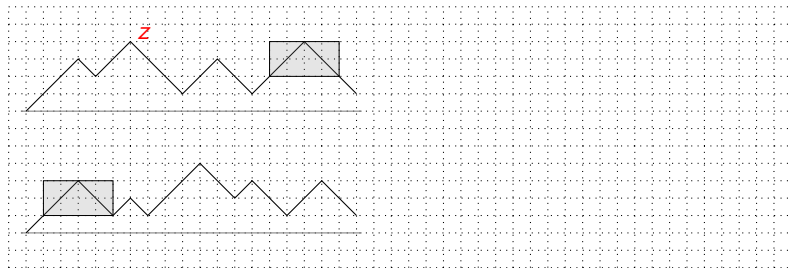
The constructive algorithm: Second step - case 2

φ contains at least one marked forbidden pattern

- Find z the leftmost point in φ having highest ordinate and not strictly contained in a marked forbidden pattern.
- Apply the *cut and paste* operation.

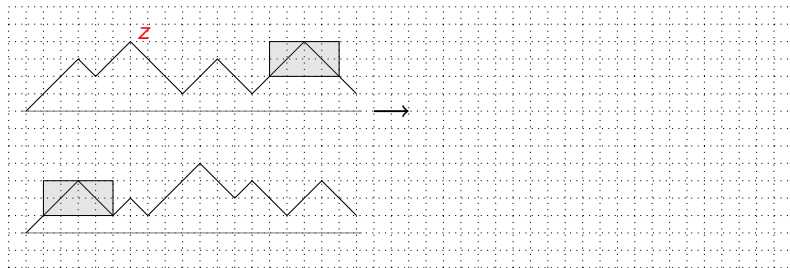
The constructive algorithm: Second step - case 2

φ contains at least one marked forbidden pattern



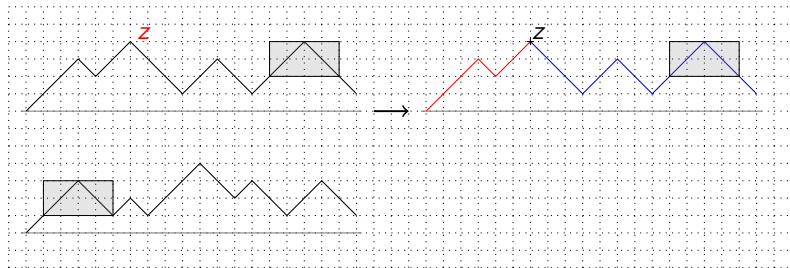
The constructive algorithm: Second step - case 2

φ contains at least one marked forbidden pattern



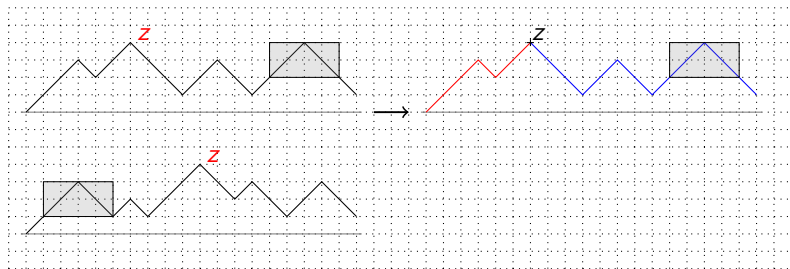
The constructive algorithm: Second step - case 2

φ contains at least one marked forbidden pattern



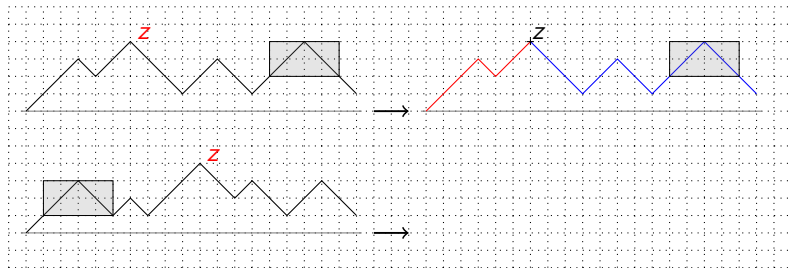
The constructive algorithm: Second step - case 2

φ contains at least one marked forbidden pattern



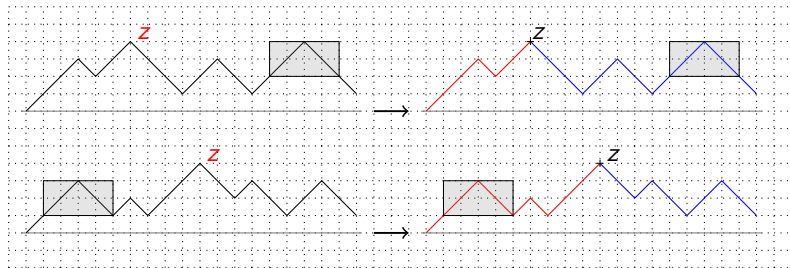
The constructive algorithm: Second step - case 2

φ contains at least one marked forbidden pattern



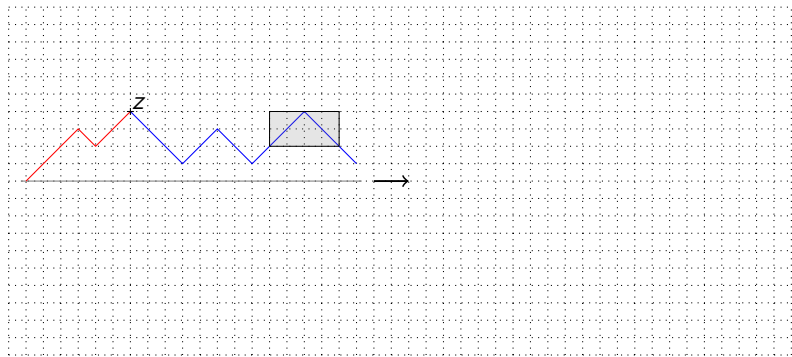
The constructive algorithm: Second step - case 2

φ contains at least one marked forbidden pattern



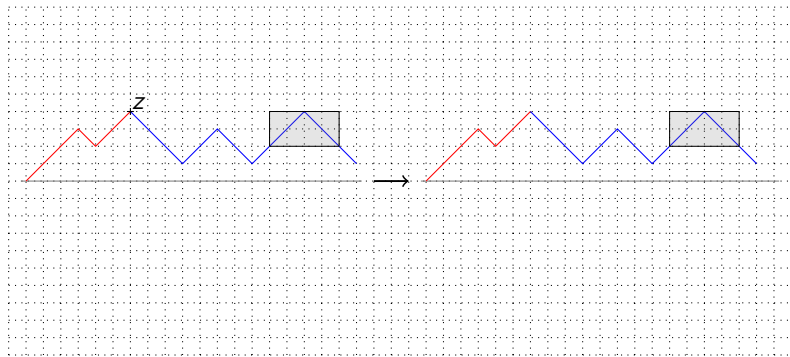
The constructive algorithm: Second step - case 2

Cut and paste



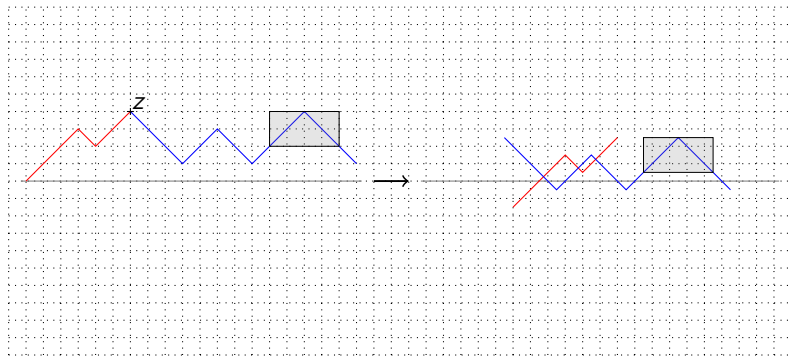
The constructive algorithm: Second step - case 2

Cut and paste



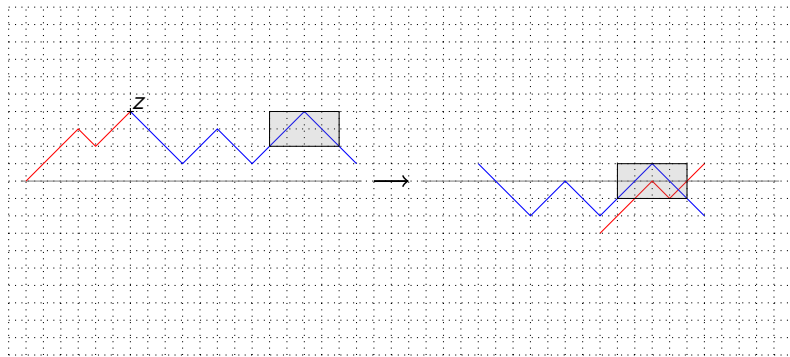
The constructive algorithm: Second step - case 2

Cut and paste



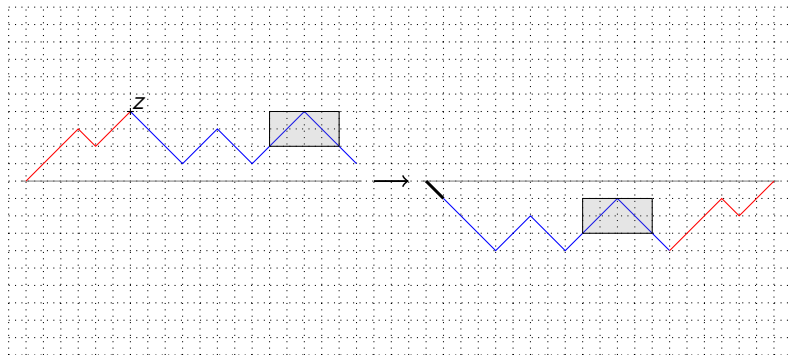
The constructive algorithm: Second step - case 2

Cut and paste



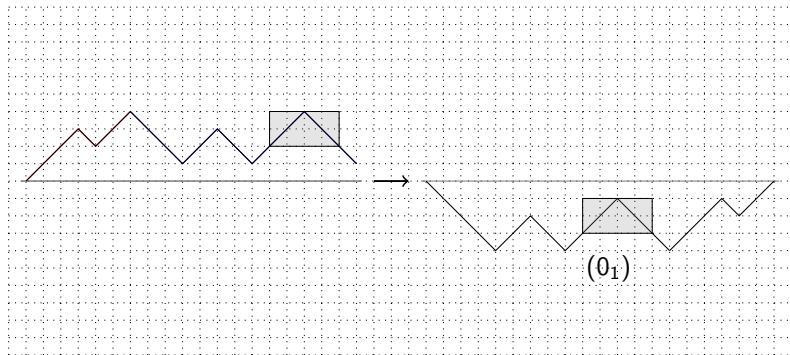
The constructive algorithm: Second step - case 2

Cut and paste



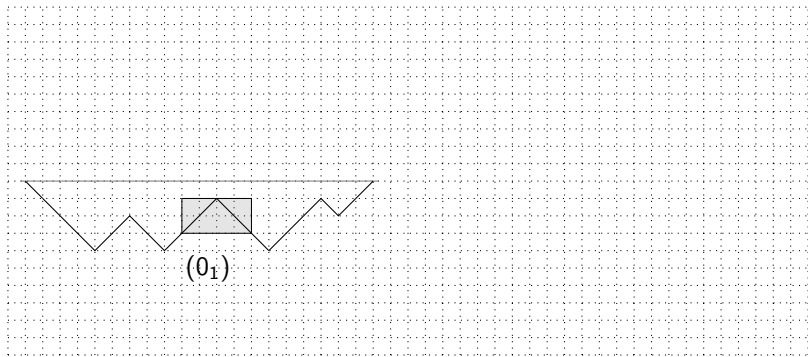
The constructive algorithm: Second step - case 2

Cut and paste



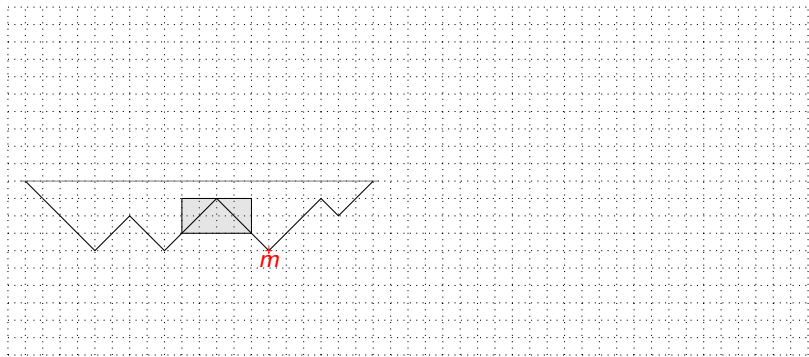
The constructive algorithm: Second step - case 2

Reverse of cut and paste



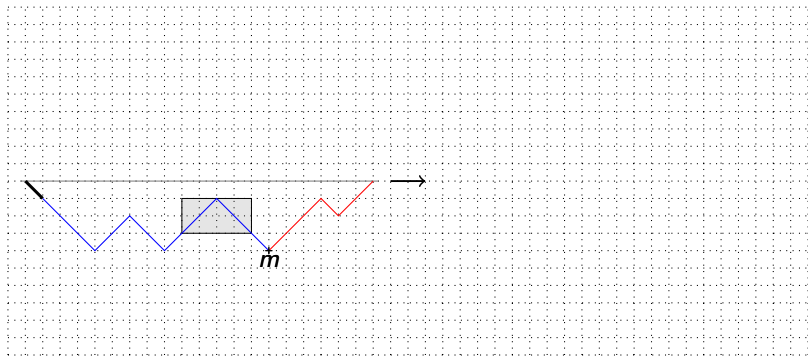
The constructive algorithm: Second step - case 2

Reverse of cut and paste



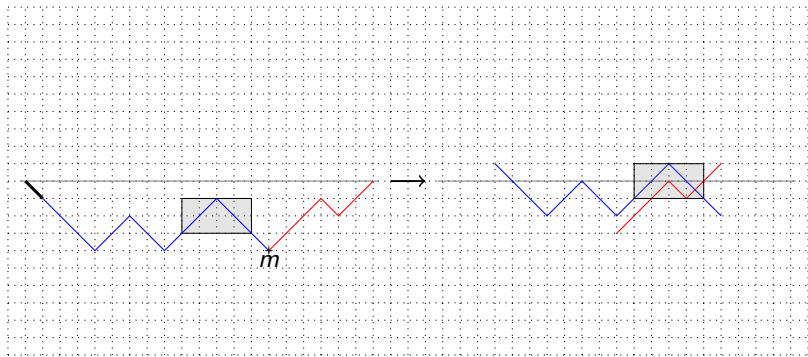
The constructive algorithm: Second step - case 2

Reverse of cut and paste



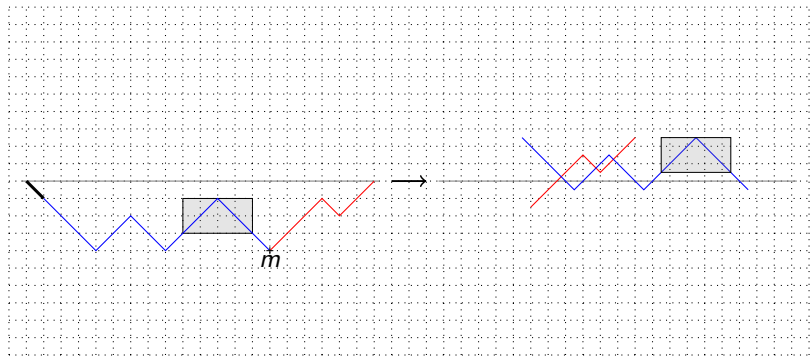
The constructive algorithm: Second step - case 2

Reverse of cut and paste



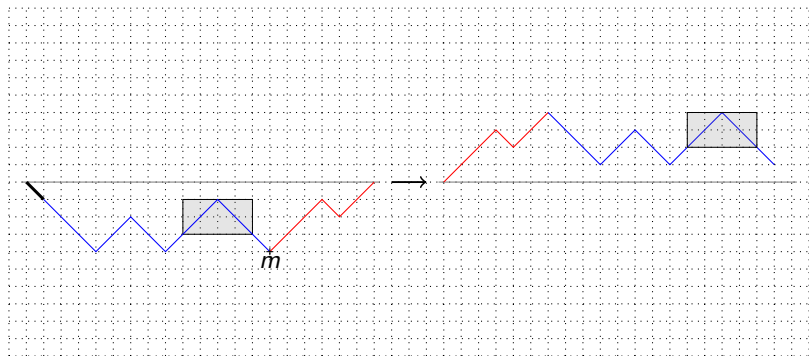
The constructive algorithm: Second step - case 2

Reverse of cut and paste

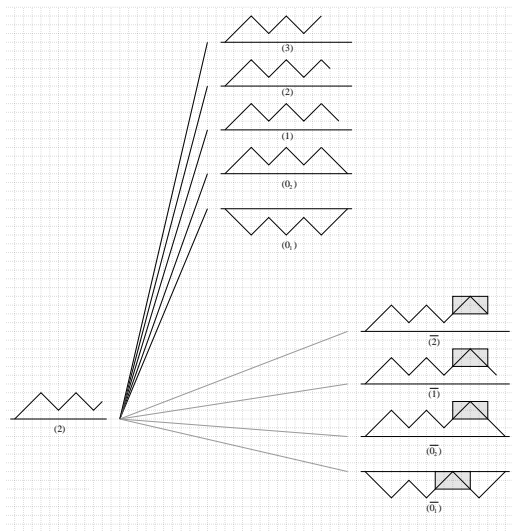


The constructive algorithm: Second step - case 2

Reverse of cut and paste

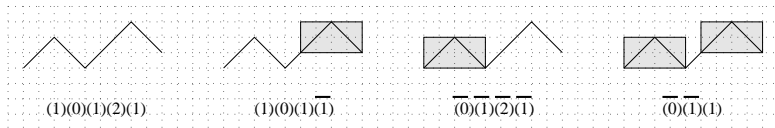


Example of the complete algorithm



Proof: idea

The described algorithm is a construction for the set $F^{[p]}$ according to the number of rise steps. This means that all the paths in F with n rise steps are obtained. Moreover, for each obtained path ψ in $F \setminus F^{[p]}$, having C forbidden patterns, with n rise steps and (k) as last label of the associated code, a path ψ' in $F \setminus F^{[p]}$ having C forbidden patterns, with n rise steps, and (\bar{k}) as last label of the associated code is also generated having the same shape as ψ but such that the last forbidden pattern is marked if it is not in ψ and vice-versa.



Enumeration

Let Z be the set of paths whose instances are coded by a sequence of labels in the generating tree ending by a non-marked zero, S be the set of paths whose instances are coded by a sequence of labels ending by a marked zero, N be the set of paths whose instances are coded by a sequence of labels ending by a non-marked $k \geq 1$ and M be the set of paths whose instances are coded by a sequence of labels ending by a marked $k \geq 1$.

Then $F^{[p]} = (Z \setminus S) \cup (N \setminus M)$.

$$Z(x, 1) = \sum_{\omega \in Z} x^{n(\omega)} y^0 = 1 + 2xZ(x, 1) + 2xN(x, 1) + x^j S(x, 1) + 2x^j M(x, 1),$$

$$S(x, 1) = \sum_{\omega \in S} x^{n(\omega)} y^0 = 2xS(x, 1) + 2xM(x, 1) + x^j Z(x, 1) + 2x^j N(x, 1),$$

$$N(x, y) = \sum_{\omega \in N} x^{n(\omega)} y^{h(\omega)} = xyZ(x, 1) + \sum_{\omega \in N} \sum_{i=1}^{h(\omega)+1} x^{n(\omega)+1} y^i + \sum_{\omega \in M} \sum_{i=1}^{h(\omega)} x^{n(\omega)+j} y^i,$$

$$M(x, y) = \sum_{\omega \in M} x^{n(\omega)} y^{h(\omega)} = xyS(x, 1) + \sum_{\omega \in M} \sum_{i=1}^{h(\omega)+1} x^{n(\omega)+1} y^i + \sum_{\omega \in N} \sum_{i=1}^{h(\omega)} x^{n(\omega)+j} y^i.$$

Enumeration

We obtain the generating function $F_j(x)$, $j > 1$, for the words $\omega \in F^{[p]}$ according to the number of ones:

$$F_j(x) = \frac{y_0(x) - x^{j-1}}{(1 - x^{j-1})(1 + x^j - 2xy_0(x))},$$

where

$$y_0(x) = \frac{x^j + 1 - \sqrt{(x^j + 1)^2 - 4x}}{2x}.$$

Enumeration

Let us remark that the generating function $F_j(x)$ depends only on the number of ones in the forbidden pattern. So, primitive Dyck words represent an *invariant class* for the enumeration of the words which avoid anyone of such a shape path. This means that all the sets in F with n ones avoiding a forbidden pattern $p \in \mathcal{D}_j^p$ have the same cardinality, independently on the shape of p .

Example

For any pattern p in D_4^p , the first numbers of the sequence enumerating the binary words in $F^{[p]}$, according to the number of ones, are:
 1, 3, 10, 35, 125, 454, 1671, 6211, 23261, \dots being

$$F_4(x) = \frac{1 - x^4 - \sqrt{x^8 + 2x^4 + 1 - 4x}}{2x(1 - x^3)\sqrt{x^8 + 2x^4 + 1 - 4x}}$$

the associated generating function.

Generalization

Given a set \mathcal{Q} of forbidden patterns, we say that the pattern p_ℓ is *redundant* in \mathcal{Q} if there is a pattern p_h in \mathcal{Q} that is a sub-path of p_ℓ .

Let $\mathcal{Q} = \{p_1, p_2, \dots, p_m\}$ be a set of non redundant forbidden patterns where each pattern p_i belongs to $\mathcal{D}_{j_i}^p$, that is a primitive Dyck path with j_i rise steps.

$$\left\{ \begin{array}{l} (0) \\ (k) \xrightarrow{1} (k+1)(k) \cdots (1)(0_2)(0_1) \quad k \geq 0 \\ (0) \xrightarrow{j_1} (\overline{0_2}) \\ (k) \xrightarrow{j_1} (\overline{k})(\overline{k-1}) \cdots (\overline{1})(\overline{0_2})(\overline{0_1}) \quad k \geq 1 \\ (0) \xrightarrow{j_2} (\overline{0_2}) \\ (k) \xrightarrow{j_2} (\overline{k})(\overline{k-1}) \cdots (\overline{1})(\overline{0_2})(\overline{0_1}) \quad k \geq 1 \\ \vdots \\ (0) \xrightarrow{j_m} (\overline{0_2}) \\ (k) \xrightarrow{j_m} (\overline{k})(\overline{k-1}) \cdots (\overline{1})(\overline{0_2})(\overline{0_1}) \quad k \geq 1 \end{array} \right.$$

Generalization

By using the previous notation, we have:

$$Z(x, 1) = 1 + 2xZ(x, 1) + 2xN(x, 1) + (x^{j_1} + \dots + x^{j_m})S(x, 1) + 2(x^{j_1} + \dots + x^{j_m})M(x, 1),$$

$$S(x, 1) = 2xS(x, 1) + 2xM(x, 1) + (x^{j_1} + \dots + x^{j_m})Z(x, 1) + 2(x^{j_1} + \dots + x^{j_m})N(x, 1),$$

$$N(x, y) = xyZ(x, 1) + \sum_{\omega \in N} \sum_{i=1}^{h(\omega)+1} x^{n(\omega)+1} y^i + \sum_{\omega \in M} \sum_{i=1}^{h(\omega)} x^{n(\omega)+j_1} y^i + \dots + \sum_{\omega \in M} \sum_{i=1}^{h(\omega)} x^{n(\omega)+j_m} y^i,$$

$$M(x, y) = xyS(x, 1) + \sum_{\omega \in M} \sum_{i=1}^{h(\omega)+1} x^{n(\omega)+1} y^i + \sum_{\omega \in N} \sum_{i=1}^{h(\omega)} x^{n(\omega)+j_1} y^i + \dots + \sum_{\omega \in N} \sum_{i=1}^{h(\omega)} x^{n(\omega)+j_m} y^i.$$

Generalization

We obtain the generating function $F_{j_1, \dots, j_m}(x)$ for the words $\omega \in F^{[\mathbb{Q}]}$ according to the number of ones:

$$F_{j_1, \dots, j_m}(x) = \frac{y_0(x) - x^{j_1-1} - \dots - x^{j_m-1}}{(1 - x^{j_1-1} - \dots - x^{j_m-1})(1 + x^{j_1} + \dots + x^{j_m} - 2xy_0(x))},$$

where

$$y_0(x) = \frac{1 + x^{j_1} + \dots + x^{j_m} - \sqrt{(x^{j_1} + \dots + x^{j_m} + 1)^2 - 4x}}{2x}.$$

Further developments

1. Study other forbidden patterns p
2. Expand the alphabet $\{0, 1\}$
3. Find a unified approach for pattern avoidance construction and enumeration

Further developments

1. Study other forbidden patterns p
2. Expand the alphabet $\{0, 1\}$
3. Find a unified approach for pattern avoidance construction and enumeration

Further developments

1. Study other forbidden patterns p
2. Expand the alphabet $\{0, 1\}$
3. Find a unified approach for pattern avoidance construction and enumeration

Thanks for your attention