PU.M.A. Vol. **17** (2006), No. 1–2, pp. 19–37

ECO-Generation for p-generalized Fibonacci and Lucas permutations

Jean-Luc Baril
LE2I UMR-CNRS 5158, Université de Bourgogne
B.P. 47 870, 21078 DIJON-Cedex France
e-mail: barjl@u-bourgogne.fr

and

Phan-Thuan Do
LE2I UMR-CNRS 5158, Université de Bourgogne
B.P. 47 870, 21078 DIJON-Cedex France
e-mail: phan-thuan.do@u-bourgogne.fr

(Received: October 31, 2006)

Abstract. Using the ECO method, we give succession rules for p-generalized Fibonacci and Lucas sequences. We provide sets of pattern-avoiding permutations which are enumerated by the p-generalized Fibonacci and Lucas sequences. Finally we give generating algorithms in Constant Amortized Time for these sets.

Mathematics Subject Classifications (2000). 05A05, 05A15, 69R05

Keywords: Fibonacci and Lucas permutations, ECO method, pattern-avoiding permutation.

1 Introduction

Barcucci, Del Lungo, Pergola and Pinzani have developed an Enumerating Combinatorial Method (ECO) [6] for the enumeration and the recursive construction of classes of combinatorial objects. The method is used for *enumeration* [4, 5, 6], *algebraic characterization* such as operations on succession rules [9, 13, 14] or production matrices [10], *generating functions* associated with an ECO operator [2, 11], *random and exhaustive generation* [1, 3, 18]. The ECO method consists in producing succession rules to describe certain combinatorial object classes. A *succession rule* (Ω) is a system consisting of an axiom (b), $b \in \mathbb{N}^+$, and a set of productions:

$$\{(k) \rightsquigarrow (e_1(k))(e_2(k))\dots(e_k(k)), k \in \mathbb{N}\},$$

where $e_i : \mathbb{N}^+ \longrightarrow \mathbb{N}^+$, which explains how to derive the successors $(e_1(k))$, $(e_2(k))$,..., $(e_k(k))$ of any given (k), $k \in \mathbb{N}^+$. The positive integers (b), (k), $(e_i(k))$ are called *labels* of (Ω). The root can be represented by means of a *generating tree* where (b) is the label of the root and each node labeled (k) has k sons labeled $(e_1(k))$, $(e_2(k))$,..., $(e_k(k))$. A succession rule (Ω) induces a sequence of positive integers $(a_n)_{n\geq 0}$ where a_n is the number of nodes at level n in the generating tree.

The succession rules for traditional Fibonacci sequences are given by Banderier et al. [2]. In this paper, we extend this result to p-generalized Fibonacci and Lucas sequences. Moreover, we are interested in families of pattern-avoiding permutations which can be enumerated in terms of p-generalized Fibonacci or Lucas sequences. In this regard, a number of papers concerning Fibonacci sequences have been published [7, 12, 17, 19, 20, 21] but none have been forthcoming on Lucas. For instance, Mansour described general results (see Theorem 3 in [19]) such that these families are particular cases. However, in [8], an efficient algorithm is given to generate generalized Lucas binary strings in Gray code order.

This paper is organized as follows. In the next section, we provide the succession rules for p-generalized Fibonacci sequences and their connections with pattern-avoiding permutations. Then we investigate p-generalized Lucas sequences by giving succession rules and two sets of pattern-avoiding permutations which are counted by these sequences. In the last section, we produce constant amortized time algorithms to generate these different classes of p-generalized Fibonacci and Lucas permutations.

2 p-generalized Fibonacci sequences

In this part of the study we only consider the results obtained from p-generalized Fibonacci sequences. Some of the results in this section have already been presented in [7]. Recall that the p-generalized Fibonacci sequences verify $f_{p,n} = \sum_{i=1}^{p} f_{p,n-i}$ anchored by $f_{p,n} = 0$ if $n \leq 0$, $f_{p,1} = 1$ and let us denote $F_{p,n}$ be the set of n-length binary strings without p consecutive ones. Obviously $|F_{p,n}| = f_{p,n+2}$.

2.1 Succession rules

In order to construct generating trees for the p-generalized Fibonacci sequences we use the ECO method, providing the label of its root and a set of succession rules which describe, for each node, the label set of its successors. There are many ways to label fields of succession rules. Here, each node has at most two successors. So we label each node by (2_i) or (1) to encode information concerning the number of successors (2 or 1); the index i $(0 < i < p)$ allows us to distinguish nodes having two successors.

For example, the succession rules for the well-known Fibonacci sequences $(p = 2)$, which appear in [2], can be written as:

$$\begin{cases} (2_1) \\ (2_1) \rightsquigarrow (2_1)(1) \\ (1) \rightsquigarrow (2_1). \end{cases}$$

The following theorem extends this finding to p-generalized Fibonacci sequences.

Theorem 1 *For $p \geq 2$, a system of succession rules (Ω_p) for p-generalized*

Fibonacci sequences is given by:

$$(\Omega_p)\begin{cases}(2_{p-1})\\ (2_{p-1}) \rightsquigarrow (2_{p-1})(2_{p-2})\\ (2_{p-2}) \rightsquigarrow (2_{p-1})(2_{p-3})\\ \dots\\ (2_1) \rightsquigarrow (2_{p-1})(1)\\ (1) \rightsquigarrow (2_{p-1}).\end{cases}$$

Proof. By translating the concept of succession rules into matrix notation [10], we obtain the *production matrix* of size $p \times p$, as shown below:

$$M_p = \begin{pmatrix} 1 & 1 & 0 & \dots & 0 & 0\\ 1 & 0 & 1 & \dots & 0 & 0\\ \dots & \dots & \dots & \ddots & \dots & \dots\\ 1 & 0 & 0 & \dots & 1 & 0\\ 1 & 0 & 0 & \dots & 0 & 1\\ 1 & 0 & 0 & \dots & 0 & 0\end{pmatrix}.$$

Here the i-th row (or column) of the production matrix corresponds to the label (2_{p-i}) for $1 \leq i \leq p-1$ and (1) for $i = p$. For example, the entry in the first row and the second column is the number of labels (2_{p-2}) produced from the label (2_{p-1}).

Let d_p be the determinant of $(x \cdot I - M_p)$ where x is a variable and I the unitary matrix of size $p \times p$. It is easy to prove that d_p verifies the recurrence relation: $d_p = x \cdot d_{p-1} - 1$ anchored by $d_1 = x - 1$. So we obtain the characteristic polynomial $d_p = x^p - x^{p-1} - x^{p-2} - \dots - x - 1$.

Using the Hamilton–Cayley theorem we replace x by M and:

$$M_p^p - M_p^{p-1} - M_p^{p-2} - \dots - M_p - I = 0.$$

Since M_p is invertible, it is equivalent to

$$M_p^n = \sum_{i=n-p}^{n-1} M_p^i, \qquad \forall n \geq p.$$

The succession rules (Ω_p) induce a sequence $(a_n)_{n \geq 0}$ where a_n is the number of nodes at level n ($a_0 = 1$ since the tree has only one root). By [10], $a_n = u^T \cdot M_p^n \cdot e$ where u^T is the row vector $(1\ 0\ 0\ \dots)$ of length p and e is the column vector $(1\ 1\ 1\ \dots)^T$ of length p. So the previous equality implies:

$$a_n = \sum_{i=n-p}^{n-1} a_i, \quad \forall n \geq p.$$

Moreover each node on level $0 \leq n \leq p-2$ has exactly two successors. Thus the number of nodes on level $n \leq p-1$ is obviously 2^n.

So

$$a_n = \begin{cases} 2^n & \text{if } 0 \leq n \leq p-1 \\ \sum_{i=n-p}^{n-1} a_i & \text{if } p \leq n, \end{cases}$$

which implies that a_n is equal to the well-known p-generalized Fibonacci number $f_{p,n+2}$. □

By means of simple calculations, we produce the formula of the well-known generating function of p-generalized Fibonacci [12]:

$$f_p(z) = \frac{1 + z + z^2 + \ldots + z^{p-1}}{1 - z - z^2 - \ldots - z^p} = \frac{1}{z} \cdot \left(\frac{1}{1 - \sum_{i=1}^{p} z^i} - 1 \right).$$

Since $f_p(z) \underset{p\to\infty}{\longrightarrow} \frac{1}{1-2z}$, these succession rules produce a kind of discrete continuity between the p-generalized Fibonacci sequences ($p \geq 2$) and the sequences of binary strings.

The following proposition demonstrates how the p-generalized Fibonacci strings can be attached to the generating tree of (Ω_p).

Proposition 1 *The generating tree of (Ω_p) can be encoded by the p-generalized Fibonacci binary strings. An n-length binary string is obtained from one of length $(n-1)$ by inserting 0 or 1 into its last position (see Figure 1).*

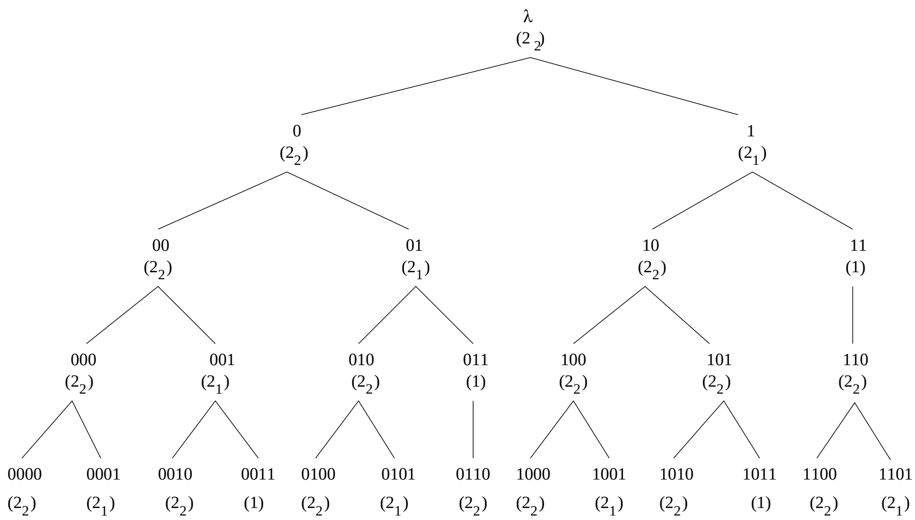

Figure 1: The first five levels of the generating tree encoded by the 3-generalized Fibonacci binary strings.

Proof. We will prove this by recurrence. Obviously, this is true for the root which is encoded by the empty binary string λ (for convenience let the level of the root be 0). Assume that this holds until the level k, i.e. each level $i \leq k$

is encoded by the elements of $F_{p,i}$. We will show that this is also true for the level $k+1$.

Let τ be the $(k+1)$-length binary string of one node at the level $k+1$ and let δ be its predecessor on the level k.

Firstly, if τ is obtained from δ by inserting 0 on the right then τ also belongs to $F_{p,k+1}$. Secondly, if τ is obtained by inserting 1 on the right of δ then δ has two children, thus its label is (2_i), $(1 \leq i \leq p-1)$.

We distinguish between two cases.

- δ is labeled (2_{p-1}). Here we obtain $\delta = \delta'0$ and $\tau = \delta'01$ with $\delta' \in F_{p,k-1}$ via the recurrence hypothesis. Thus τ belongs to $F_{p,k+1}$.
- δ is labeled (2_i), $1 \leq i \leq p-2$. Its predecessor δ_1 is labeled (2_{i+1}). We repeat the process with δ_1 until we find a label (2_{p-1}) (notice that the process always finishes). We have thus constructed a path $\delta_1, \ldots, \delta_{p-1-i}$ where δ_j is the successor of δ_{j+1} for $(1 \leq j \leq p-2-i)$ and δ_j is labeled (2_{i+j}). In particular δ_{p-1-i} is labeled (2_{p-1}). So we deduce that δ is of the form $\delta = \delta'01^{p-i-1}$ and $\tau = \delta'01^{p-i}$ where $\delta' \in F_{p,k-p+i}$ by the recurrence hypothesis. Since the length $p-1-i$ of the path verifies $p-1-i \leq p-2$, we conclude that $\tau \in F_{p,k+1}$.

Moreover, using Theorem 1, each level n of the tree contains exactly $f_{p,n+2} = |F_{p,n}|$ nodes, which shows that the tree is encoded by the p-generalized Fibonacci binary strings. □

2.2 Generating trees using pattern avoiding permutations

Let S_n be the set of permutations on $[n] = \{1, 2, \ldots, n\}$. We represent a permutation $\pi \in S_n$ in online notation: i.e. $\pi = \pi(1)\pi(2)\ldots\pi(n)$. A permutation $\pi \in S_n$ contains the pattern $\tau \in S_k$ if and only if a sequence of indices $1 \leq i_1 < i_2 < \ldots < i_k \leq n$ exists such that $\pi(i_1)\pi(i_2)\ldots\pi(i_k)$ is ordered as τ. We denote by $S_n(\tau)$ the set of permutations of S_n avoiding the pattern τ.

In this subsection, we encode the generating tree of (Ω_p) by permutations. The root is encoded by the identity of length one. Let π be an n-length permutation in the generating tree; each successor is obtained from π by inserting $n+1$ into certain positions also known as the *active sites* of π.

THEOREM 2 *The generating tree of (Ω_p) can be encoded by the permutations π in $S_n(321, 312, 234\ldots(p+1)1)$. The active sites of π are the two last positions if the label of π is (2_i); otherwise only the last position is active (see Figure 2).*

Proof. We will prove this by recurrence. Obviously, this is true for the root which is encoded by the permutation (1) (for convenience let the level of the root take the value 1). Assume that this holds until the level $k \geq 1$, i.e. each level $i \leq k$ is encoded by the permutations in $S_i(321, 312, 234\ldots(p+1)1)$. We will show that this also holds for the level $k+1$.

Let τ be the $(k+1)$-length permutation of one node of the level $k+1$ and let δ be its predecessor on the level k.

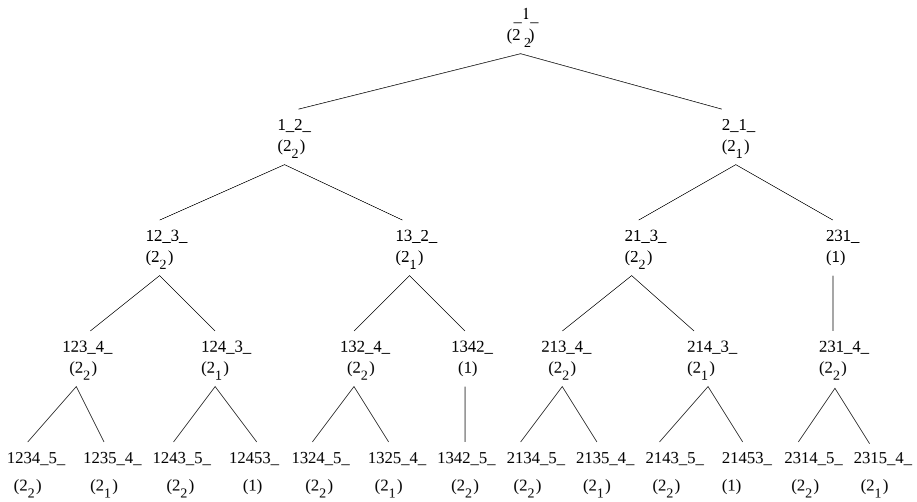

Figure 2: The first five levels of the generating tree of (Ω_3) are encoded by permutations in $S_n(321, 312, 2341)$. The active sites are represented by underscores.

Firstly, if τ is obtained from δ by inserting $k+1$ into the last position, then τ does not contain any occurrence of the forbidden subsequences 321, 312 and $234\ldots(p+1)1$.

Secondly, if τ is obtained by inserting $k+1$ into the site (k) of δ (i.e. the site just before the last entry of δ) then τ does not contain any occurrence of the forbidden subsequences 321, 312. We will now show that τ also belongs to $S_{k+1}(234\ldots(p+1)1)$. Notice that in this case δ has two children, thus its label is (2_i), $(1 \leq i \leq p-1)$.

We can distinguish two cases.

- δ is labeled (2_{p-1}). Here we obtain $\delta = \delta' k$ and $\tau = \delta'(k+1)k$ with $\delta' \in S_{k-1}(321, 312, 234\ldots(p+1)1)$ by the recurrence hypothesis. Thus τ belongs to $S_{k+1}(321, 312, 234\ldots(p+1)1)$.

- δ is labeled (2_i), $1 \leq i \leq p-2$. Its predecessor δ_1 is labeled (2_{i+1}). We repeat the process with δ_1 until we find a label (2_{p-1}). Remark that the process always finishes. So we have constructed a path $\delta_1, \ldots, \delta_{p-1-i}$ where δ_j is successor of δ_{j+1} for $(1 \leq j \leq p-2-i)$ and δ_j is labeled (2_{i+j}). In particular δ_{p-1-i} is labeled (2_{p-1}). So we deduce that δ is of the form $\delta = \delta'(k-p+i+2)\ldots(k-1)k(k-p+i+1)$ and $\tau = \delta'(k-p+i+2)\ldots(k-1)k(k+1)(k-p+i+1)$ where $\delta' \in S_{k-p+i}(321, 312, 234\ldots(p+1)1)$ via the recurrence hypothesis. Since the length $p-1-i$ of the path verifies $p-1-i \leq p-2$, we conclude that τ does not contain any occurrence of the forbidden subsequence $23\ldots(p+1)1$. Hence $\tau \in S_{k+1}(321, 312, 234\ldots(p+1)1)$.

Moreover, Mansour (Theorem 3, [19]) found the cardinality of $S_{k+1}(321, 312, \tau)$ for any $\tau \in S_{p+1}(321, 312)$. In particular, he obtains that the cardinality of $S_{k+1}(321, 312, 234...(p+1)1)$ is equal to that $F_{p,k}$. □

By considering other active sites, we obtain an other set of pattern-avoiding permutations which is associated with the same generating tree.

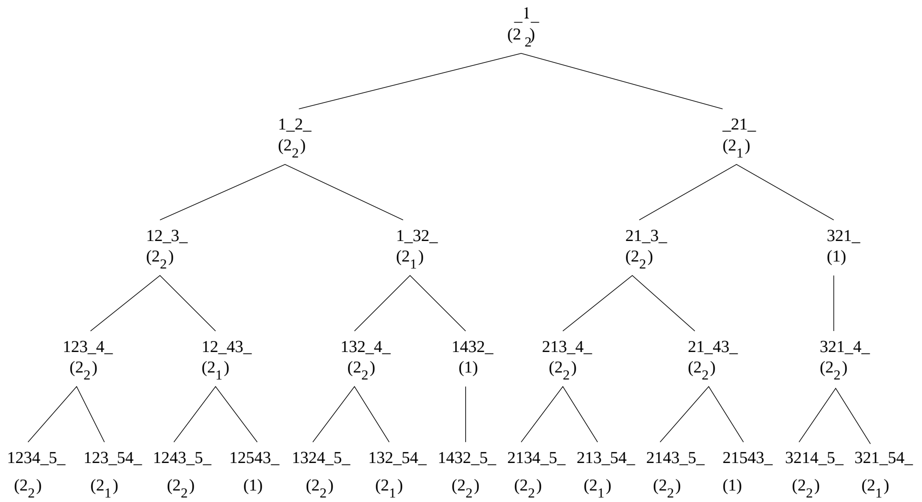

Figure 3: The first five levels of the generating tree of 3-generalized Fibonacci sequences encoded by permutations $S_n(231, 312, 4321)$. The active sites are represented by underscores.

THEOREM 3 *The generating tree of* (Ω_p) *can be encoded by the permutations* π *in* $S_n(231, 312, (p+1)p \ldots 321)$. *A permutation* π *always has an active site in the last position; moreover, in the case where the label is* (2_i)*, there is another one just before the maximal entry of* π *(see Figure 3).*

Proof. In fact, an insertion into the last active site or into active site just before the site of the maximal entry does not produce any patterns of the form 231 and 312. The rest of the proof is similar to the demonstration of Theorem 2. □

3 p-generalized Lucas sequences

In this section, we consider p-generalized Lucas sequences. Recall that the p-generalized Lucas sequences verify: $\ell_{p,n} = \sum_{i=1}^{p} \ell_{p,n-i}$ anchored by $\ell_{p,n} = 2^n - 1$ if $0 < n \leq p$, and let $L_{p,n}$ be the set of n-length binary strings without p consecutive ones by considering the strings circularly. Notice that for convenience we assume here that the string 1^n $(n < p)$ belongs to $L_{p,n}$. So $|L_{p,n}| = \ell_{p,n} + 1$

if $n < p$ and $|L_{p,n}| = \ell_{p,n}$ otherwise. We thus exhibit succession rules for p-generalized Lucas sequences and provide a generating tree for two new associated classes of pattern-avoiding permutations.

3.1 Succession rules

In Section 2 each node was labeled (1) or (2_i) where $1 \leq i \leq p-1$. In the following we consider the label (1) or $(1')$ for the nodes having one successor and the labels (2_i) or $(2'_i)$ $(1 \leq i \leq p-1)$ for the nodes having two successors.

Theorem 4 *For $p \geq 2$, a system of succession rules (Φ_p) for p-generalized Lucas sequences is expressed by:*

$$(\Phi_p)\left\{\begin{array}{l}(2'_{p-1})\\(2'_{p-1}) \rightsquigarrow (2_{p-1})(2'_{p-2})\\(2'_{p-2}) \rightsquigarrow (2_{p-2})(2'_{p-3})\\\dots\\(2'_1) \rightsquigarrow (2_1)(1')\\(1') \rightsquigarrow (1)\\(2_{p-1}) \rightsquigarrow (2_{p-1})(2_{p-2})\\(2_{p-2}) \rightsquigarrow (2_{p-1})(2_{p-3})\\\dots\\(2_1) \rightsquigarrow (2_{p-1})(1)\\(1) \rightsquigarrow (2_{p-1}),\end{array}\right.$$

which means that (Φ_p) produces $\ell_{p,n}$ if $n > p$ and $\ell_{p,n} + 1 = 2^n$ otherwise.

Proof. By translating the concept of succession rules into matrix notation [10], we obtain the following production matrix of size $2p \times 2p$. The i-th column (or row) corresponds to the labels $(2'_i)$ for $i \leq p-1$, $(1')$ for $i = p$, (2_i) for $p+1 \leq i \leq 2p-1$ and (1) for $i = 2p$.

$$M_p = \left(\begin{array}{ccc} A_p & | & B_p \\ --- & + & --- \\ C_p & | & D_p \end{array}\right),$$

$$A_p = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \ddots & \dots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix}, \qquad B_p = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \ddots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix},$$

$$C_p = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 \end{pmatrix}, \qquad D_p = \begin{pmatrix} 1 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \ddots & \dots \\ 1 & 0 & 0 & \dots & 1 \\ 1 & 0 & 0 & \dots & 0 \end{pmatrix}.$$

It is easy to see that the number a_n of nodes on a level $n \leq p-1$ (the root being at the zero level) verifies: $u^T \cdot M_p^n \cdot e = 2^n$ (see Section 2.1 for the definition of u^T and e) and the number of nodes on the level p is $u^T \cdot M_p^p \cdot e = 2^p - 1$. Moreover by [10] the distribution on the level $p+1$ of the labels $(2'_i)$ for $1 \leq i \leq p-1$, $(1')$, (2_i) for $1 \leq i \leq p-1$ and (1) is the row vector

$$r_{p+1} = u^T \cdot M_p^{p+1} = (0, 0, \ldots, 0, 2^p - 1, 2^{p-1} - 1, \ldots, 2^2 - 1, 1).$$

Thus

$$r_{p+1} = (0, 0, \ldots, 0, \ell_{p,p}, \ell_{p,p-1}, \ldots, \ell_{p,1})$$

and

$$u^T \cdot M_p^{p+1} \cdot e = \ell_{p,p+1}.$$

By recurrence on $i \geq 1$,

$$\begin{aligned} r_{p+i+1} &= u^T \cdot M_p^{p+i} \cdot M_p \\ &= (0, 0, \ldots, 0, \ell_{p,p+i-1}, \ell_{p,p+i-2}, \ldots, \ell_{p,i}) \cdot M_p \\ &= (0, 0, \ldots, 0, \sum_{j=i}^{p+i-1} \ell_{p,j}, \ell_{p,p+i-1}, \ell_{p,p+i-2}, \ldots, \ell_{p,i+1}) \\ &= (0, 0, \ldots, 0, \ell_{p,p+i}, \ell_{p,p+i-1}, \ell_{p,p+i-2}, \ldots, \ell_{p,i+1}), \end{aligned}$$

and

$$u^T \cdot M_p^{p+i+1} \cdot e = \ell_{p,p+i+1}, \qquad \forall i \geq 1,$$

thus the sequence $(a_n)_{n \geq 0}$ induced by (Φ_p) verifies:

$$a_n = u^T \cdot M_p^n \cdot e = \begin{cases} 0 & \text{if } n < 0 \\ \ell_{p,n} + 1 & \text{if } 0 \leq n \leq p-1 \\ \ell_{p,n} & \text{if } n \geq p. \end{cases}$$

We deduce that a_n is equal to the well known p-generalized Lucas sequence $\ell_{p,n}$ (except for $n \leq p-1$ where $a_n = \ell_{p,n} + 1$). $\square$

By means of simple calculations, we can deduce the generating function of the sequences corresponding to (Φ_p) as follows:

$$\ell_p(z) = \frac{\sum_{i=0}^{p-1} (z^i - i \cdot z^{2p-i})}{1 - z - z^2 - \ldots - z^p}.$$

As for the case of Fibonacci and since $\ell_p(z) \underset{p \to \infty}{\longrightarrow} \frac{1}{1-2z}$, these succession rules produce a kind of discrete continuity between the p-generalized Lucas sequences $(p \geq 2)$ and the sequences of binary strings.

Here the generating tree of (Φ_p) cannot be encoded by the set $L_{p,n}$. However, it is encoded by another binary string set $K_{p,n}$ defined as follows: $K_{p,n} \subseteq F_{p,n}$ and each binary string of $K_{p,n}$ contains at least two 0s in theirs length $p+1$ prefix. For example, the only binary string in $F_{2,4}$ but not in $K_{2,4}$ is 1010 and $F_{3,5} \backslash K_{3,5} = \{10110, 11011, 11010\}$. In [16] the authors provide a constructive bijection ϕ between $K_{p,n}$ and $L_{p,n}$ defined as follows: if $x \in K_{p,n}$, x can be written $x = 1^i 0 y$ with $0 \leq i < p$ and $\phi(x) = y 0 1^i \in L_{p,n}$.

Proposition 2 *The generating tree of* (Φ_p) *is encoded by the strings of the set* $K_{p,n}$. *An* n*-length binary string is obtained from one of length* $(n-1)$ *by inserting 0 or 1 into its last position (see Figure 4).*

Proof. Obviously, this is true for the root which is encoded by the empty binary string λ (for convenience let the level of the root be 0). Notice that the predecessor of each node on a level $k \le p-1$ has two successors. This means that each level k $(k \le p-1)$ is encoded by the set of all binary strings, therefore by $K_{p,k}$. For the level $k = p$, the nodes are encoded by the binary strings different from 1^p which proves the results for the level p.

Now we will prove the result for $k \ge p+1$ by recurrence. Assume that this holds until the level $k \ge p$, i.e. each level $i \le k$ is encoded by the elements of $K_{p,i}$. We will show that this is also true for the level $k+1$.

Let τ be the $(k+1)$-length binary string of one node of the level $k+1$ and let δ be its predecessor on level k.

Firstly, if τ is obtained from δ by inserting 0 on the right, then τ also belongs to $K_{p,k+1}$. Secondly, if τ is obtained by inserting 1 on the right of δ, then δ has two children, hence its label is (2_i), $(1 \le i \le p-1)$.

We distinguish between two cases.

- δ is labeled (2_{p-1}). Here we obtain $\delta = \delta'0$ and $\tau = \delta'01$ with $\delta' \in K_{p,k-1}$ by the recurrence hypothesis. Remark that $\delta' \ne 1^{k-1}$, i.e. it contains at least one zero. Indeed, since $k-1 \ge p-1$, if $\delta' = 1^{k-1}$ then its label is $(1')$ and the label of δ is (1) which is a contradiction. Thus τ belongs to $K_{p,k+1}$.

- δ is labeled (2_i), $1 \le i \le p-2$. Since $k \ge p$, its predecessor δ_1 is labeled (2_{i+1}). We repeat the process with δ_1 until we find a label (2_{p-1}). Note that since $k \ge p$ the process always finishes. Indeed if the process does not reach the label (2_{p-1}), it inevitably meets $(2'_j)$ with $j \ge i$. This would mean that $\delta = 1^{p-1-j}01^{j-i}$ and δ would be on the level $k = p-i$ which contradicts $k \ge p$. So we have constructed a path $\delta_1, \ldots, \delta_{p-1-i}$ where δ_j is successor of δ_{j+1} for $(1 \le j \le p-2-i)$. Moreover δ_j is labeled (2_{i+j}) and in particular δ_{p-1-i} by (2_{p-1}). So we deduce that δ is of the form $\delta = \delta'01^{p-i-1}$ and $\tau = \delta'01^{p-i}$ where $\delta' \in K_{p,k-p+i}$ using the recurrence hypothesis. As δ_{p-1-i} is labeled (2_{p-1}), $\delta' = 1^{k-p+i}$ implies that δ' is labeled $(2'_{p-1})$. Thus $\delta' = \lambda$ and $\tau = 01^{p-i}$ which contradicts $k+1 \ge p+1$. So, τ has at least two zeros in its length $(p+1)$ prefix and we conclude that $\tau \in K_{p,k+1}$.

Via Theorem 4, each level n of the tree contains exactly $\ell_{p,n}$ nodes for $n > p$ and $\ell_{p,n}+1$ nodes for $n \ge p$. which is the same cardinality as $K_{p,n}$ (see [16]). This shows that the tree is encoded by $K_{p,n}$, for $n \ge 1$. $\square$

3.2 Generating trees using pattern avoiding permutations

In this subsection, we consider generalized patterns. A *barred* pattern $\overline{\tau}$ is a permutation in S_k having a bar over one of its elements. Let τ be a permutation

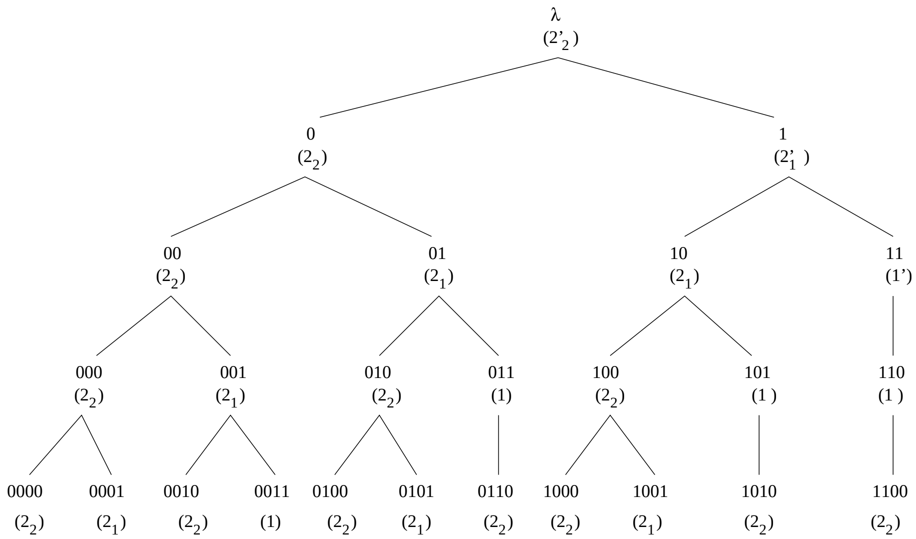

Figure 4: The first five levels of the generating tree of 3-generalized Lucas sequences encoded by $K_{3,n}$.

on $[k]$ identical to $\overline{\tau}$ but unbarred and $\hat{\tau}$ be the permutation on $[k-1]$ made up of the $(k-1)$ unbarred elements of $\overline{\tau}$, rewritten to be a permutation on $[k-1]$. Then $\pi \in S_n$ contains the pattern $\overline{\tau}$ if π contains the pattern $\hat{\tau}$ that cannot be expanded into a pattern τ in π. For example, if $\overline{\tau} = 4\bar{1}32$ then $\pi = 58132674 \in S_8(4\bar{1}32)$.

Let us also define other patterns as permutations possibly with added colons between entries (e.g. $\tau = 13\!:\!24$). If two consecutive entries in τ are separated by a *colon*, then the permutation π contains the pattern τ if the entries of π corresponding to those elements are adjacent in π (and in the same order given by τ). For example, 13524 fails to be $13:24$ avoiding but is 1324 avoiding. Notice that this is a particular case of the *distanced patterns* defined in [15].

In the following, we provide two new sets of pattern-avoiding permutations which are also enumerated by the succession rules (Φ_p).

Theorem 5 *Let us assume*

$$T_p = \begin{cases} \bar{1}34\ldots(p+1)2\!:\!(p+3)(p+2) \\ \bar{1}34\ldots p2\!:\!(p+2)(p+3)(p+1) \\ \ldots \\ \bar{1}32\!:\!56\ldots(p+3)4. \end{cases}$$

The succession rule (Φ_p) gives a generating tree encoded by the permutations in $S_n(321, 312, 234\ldots(p+1)1, T_p)$. The active sites are the two last positions if the label is (2_i) or $(2'_i)$; otherwise only the last position is active (see Figure 5).

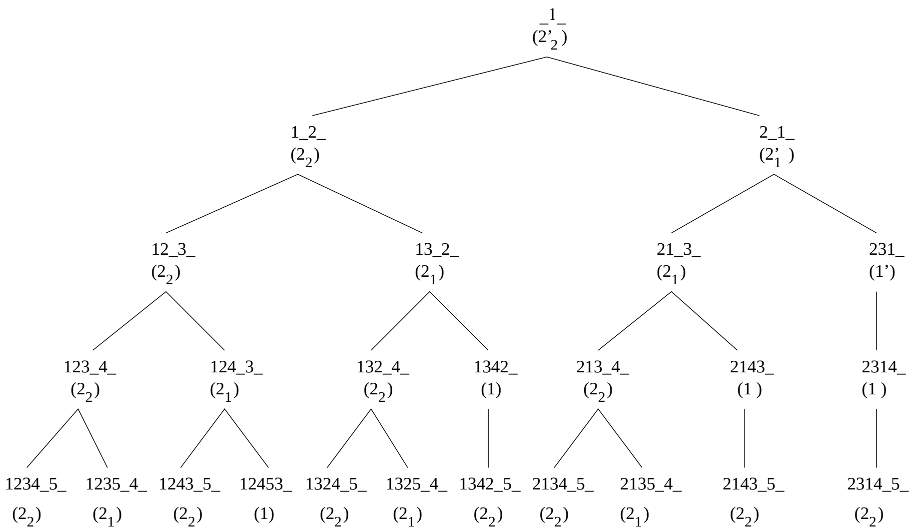

Figure 5: The first five levels of the generating tree of (Φ_3)-generalized Lucas sequences. Each node is encoded by a permutation in $S_n(321, 312, 2341, \overline{1}342\!:\!65, \overline{1}32\!:\!564)$.

Proof. Obviously, this is true for the root which is encoded by the permutation of length one (for convenience assume the level of the root is 1). Notice that the predecessor of each node on a level $k \le p$ has two successors. This means that each level k ($k \le p$) is encoded by the set of permutations avoiding 321 and 312 which is also the sets $S_k(321, 312, 234\dots(p+1)1, T_p)$ for $k \le p$. Note also that each node of level p has two successors with one exception which is labeled $(1')$ and encoded by $234\dots p1$. This means that the permutation $234\dots(p+1)1$ cannot belong to the level $p+1$. We therefore obtain the result for the level $p+1$ as well.

Now we will prove the result for $k \ge p+1$ by recurrence. Assume that this holds up to the level $k \ge p+1$, i.e. each level $i \le k$ is encoded by the elements of $S_i(321, 312, 234\dots(p+1)1, T_p)$. We will show that this is also true for the level $k+1$.

Let τ be the $(k+1)$-length permutation of one node of the level $k+1$ and let δ be its predecessor on the level k.

Firstly, if τ is obtained from δ by inserting $k+1$ on the right, then τ also belongs to $S_{k+1}(321, 312, 234\dots(p+1)1, T_p)$. Secondly, if τ is obtained by inserting $k+1$ just before the last entry of δ then δ has two children, hence its label is (2_i), $(1 \le i \le p-1)$.

We distinguish between two cases.

- δ is labeled (2_{p-1}). Here we obtain $\delta = \delta' k$ and $\tau = \delta'(k+1)k$ with $\delta' \in S_{k-1}(321, 312, 234\dots(p+1)1, T_p)$ via the recurrence hypothesis. Notice that $\delta' \neq 23\dots(k-2)(k-1)1$ since $k-1 \ge p$. Indeed if $\delta' = 23\dots(k-$

$2)(k-1)1$ then the node of δ' is labeled $(1')$ and thus the label of δ would be (1) which is a contradiction. Thus τ belongs to $S_{k+1}(321, 312, 234\dots(p+1)1, T_p)$.

- δ is labeled (2_i), $1 \le i \le p-2$. Since $k \ge p+1$, its predecessor δ_1 is labeled (2_{i+1}). We repeat the process with δ_1 until we find a label (2_{p-1}). Note that since $k \ge p+1$ the process always finishes. Indeed, if the process does not reach the label (2_{p-1}), it inevitably meets $(2'_j)$ where $j \ge i$. This would mean that $\delta = 23\dots(p-1-j)(p-j)1(p-j+2)(p-j+3)\dots(p-i)(p+1-i)(p-j+1)$ and δ would be on the level $k = p+1-i$ which contradicts $k \ge p+1$. So we have constructed a path $\delta_1, \dots, \delta_{p-1-i}$ where δ_j is the successor of δ_{j+1} for $(1 \le j \le p-2-i)$. Moreover δ_j is labeled (2_{i+j}) and in particular δ_{p-1-i} is labeled (2_{p-1}). So we deduce that δ is of the form $\delta = \delta'(k-p+i+2)(k-p+i+3)\dots(k-1)k(k-p+i+1)$ and $\tau = \delta'(k-p+i+2)(k-p+i+3)\dots k(k+1)(k-p+i+1)$ where $\delta' \in S_{k-p+i}(321, 312, 234\dots(p+1)1, T_p)$ by the recurrence hypothesis. As δ_{p-1-i} is labeled (2_{p-1}), $\delta' = 23\dots(k-p+i)1$ implies that δ' is labeled $(2'_{p-1})$. Thus $\delta' = 1$ and $\tau = 13\dots(p-i+1)(p-i+2)2$ which contradicts $k+1 = p-i+2 \ge p+2$. So $\delta' \ne 23\dots(k-p+i)1$ and we conclude that $\tau \in S_{k+1}(321, 312, 234\dots(p+1)1, T_p)$.

According to [16] $S_{k+1}(321, 312, 234\dots(p+1)1, T_p)$ and $K_{p,k+1}$ have the same cardinality which shows that the tree is encoded by the permutations in $S_n(321, 312, 234\dots(p+1)1, T_p)$ for $n \ge 1$. $\square$

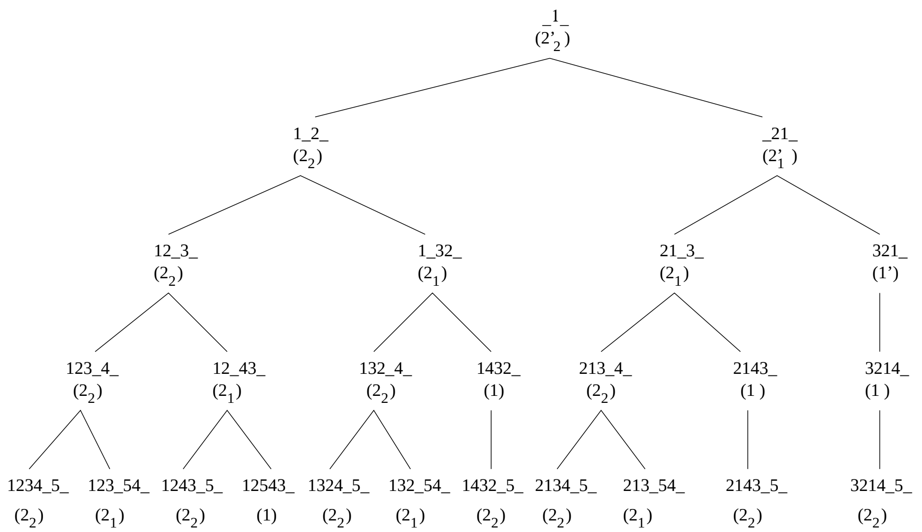

Figure 6: The first five levels of the generating tree of 3-generalized Lucas sequences. Each node is encoded by a permutation in $S_n(231, 312, 4321, \bar{1}432{:}65, \bar{1}32{:}654)$.

Now we will obtain similar results for another permutation set.

Theorem 6 *Let us assume*

$$T'_p = \begin{cases} \bar{1}(p+1)p\dots 32:(p+3)(p+2) \\ \bar{1}p(p-1)\dots 32:(p+3)(p+2)(p+1) \\ \dots \\ \bar{1}32:(p+3)(p+2)\dots 54. \end{cases}$$

The generating tree of (Φ_p) can be encoded by the permutations π in $S_n(231, 312, (p+1)p\dots 321, T'_p)$. A permutation π always has an active site on the last position; moreover in the case where the label is (2_i) or $(2'_i)$, it has another one just before the maximal entry of π (see Figure 6).

Proof. We will prove this using the same method as we did in the proof of Theorem 5. □

4 Algorithmic considerations

In this section, we provide efficient algorithms to generate the different permutation sets obtained in Sections 1 and 2. More precisely, the complexity of our algorithms is always in Constant Amortized Time (CAT). Recall that an algorithm is CAT if the number of computations after a small amount of preprocessing is proportional to the number of objects generated.

4.1 Generation for Fibonacci permutations

Consider now the generating tree of $S_n(321, 312, 234\dots(p+1)1)$. We provide below a simple recursive procedure for the generation of $S_n(321, 312, 234\dots(p+1)1)$ using the succession rules (Ω_p) defined in Section 1. The generation is anchored by $f1gen(1, p-1)$ where σ is initialized by the identity permutation $12\dots n$.

In the procedure $f1gen(i, k)$, i is the level of the recursive call (i.e. the level in the tree) and the index k corresponds to the label (2_k) for $1 \leq k \leq p-1$ and (1) for $k = 0$. Between two recursive calls of $f1gen$, the current permutation σ is updated by transposing its entry $\sigma(i)$ with $\sigma(i+1)$, i.e. $\sigma = \sigma \cdot \langle i, i+1 \rangle$.

$f1gen(i, k)$

    if $i = n$ **then** output σ;

    else

        if $k \neq 0$ **then**

            $f1gen(i+1, p-1)$;

            $\sigma = \sigma \cdot \langle i, i+1 \rangle$; {*transpose the entries* $\sigma(i)$ *and* $\sigma(i+1)$}

            $f1gen(i+1, k-1)$;

            $\sigma = \sigma \cdot \langle i, i+1 \rangle$;

        else $f1gen(i+1, p-1)$;

end ;

By comparing (in the generating tree) the number of nodes having one successor with the number of nodes with two successors and the number of objects generated, we show that the generation of Fibonacci permutations sets is CAT.

PROPOSITION 3 *The procedure $f1gen$ generates $S_n(321, 312, 234\dots(p+1)1)$ in constant amortized time.*

Proof. Let n,$p \in \mathbb{N}$ and T be the tree induced by the recursive calls of the procedure $f1gen$ and m the number of leaves of T. T has the following properties: (a) the root has 2 successors, (b) if a node x has a single successor, say y, then y is a terminal node or it has two successors.

By using these properties, we deduce

- by merging in T each one successor node with its child, we obtain a complete binary tree with $(m-1)$ internal nodes which is the number of two successor nodes.
- the number of one successor nodes is less than the number of two successor nodes.

So T has less than $2(m-1)$ internal nodes.

We have:

$$\frac{\#recursive\ calls}{\#generated\ objects} = \frac{\#internal\ nodes\ of\ T}{m} \leq \frac{2(m-1)}{m} \leq 2.$$

Moreover, the number of operations in each $f1gen(i,k)$ is negligible, thus *the average cost* of $f1gen$ is a constant. Therefore, this algorithm is CAT. □

For the generation of $S_n(231, 312, (p+1)p\dots321)$, we give the procedure $f2gen(i,k,t)$ and we run $f2gen(1,p-1,1)$ anchored by $\sigma = 12\dots n$. The first two parameters are the same as above, and t corresponds to the position of the entry i in the permutation σ. Here, between two recursive calls, we must insert $i+1$ (which is on the $(i+1)$-th position of σ) just before the entry i (which is in the position t). So we apply to σ the product of successive transpositions $\langle i, i+1\rangle$, $\langle i-1, i\rangle$, …, $\langle t, t+1\rangle$; i.e. $\sigma = \sigma \cdot \langle i, i+1\rangle \cdot \langle i-1, i\rangle \cdot \ldots \cdot \langle t, t+1\rangle$.

$f2gen(i,k,t)$

```
        if i = n then output σ;
        else
            if k ≠ 0 then
                f2gen(i+1, p−1, i+1);
                for j = i downto t
                    σ = σ · ⟨j, j+1⟩;
                f2gen(i+1, k−1, t);
```

```
            for j = t to i
               σ = σ · ⟨j, j+1⟩;
         else f2gen(i+1, p−1, i+1);
end ;
```

Proposition 4 *The procedure $f2gen$ generates $S_n(231, 312, (p+1)p\ldots 321)$ in constant amortized time.*

Proof. Clearly, according to the algorithm, the number of operations between two recursive calls in $f2gen(i, k)$ is $\mathcal{O}(p)$. Thus, similarly to the proof above, we have the average cost of $f2gen$ in $\mathcal{O}(p)$. So, this algorithm is also CAT. □

Remark 1 A simple adaptation of the procedure $f1gen$ for the binary strings gives the generation of the p-generalized Fibonacci strings in $F_{p,n}$ in constant amortized time.

4.2 Generation of Lucas permutations

As for p-generalized Fibonacci permutations, we also obtain CAT algorithms for the generation of Lucas permutation sets in Section 3.2. More precisely, the number of computations divided by the number of generated objects is in $\mathcal{O}(1)$ (respectively $\mathcal{O}(p)$) for $S_n(321, 312, 234\ldots(p+1)1, T_p)$ (respectively $S_n(231, 312, (p+1)p\ldots 321, T'_p)$). The procedures are constructed in the same way as for $f1gen$ and $f2gen$ in the previous subsection, thus we just exhibit here the procedure $l1gen$ which generates the set $S_n(321, 312, 234\ldots(p+1)1, T_p)$ and the produce $l2gen$ which generates the set $S_n(231, 312, (p+1)p\ldots 321, T'_p)$. Notice that these procedures use $f1gen$ and $f2gen$ and are anchored by $l1gen(1, p-1)$ and $l2gen(1, p-1, 1)$. Moreover the set $K_{p,n}$ can be generated in constant average cost $\mathcal{O}(1)$ by a simple adapted algorithm for binary strings. And by means of an ad hoc data structure we obtain the generation of $L_{p,n}$ in constant average cost $\mathcal{O}(p)$.

```
l1gen(i, k)

if i = n then output σ;
else
    if k ≠ 0 then
       f1gen(i+1, k);
       σ = σ · ⟨i, i+1⟩;
       l1gen(i+1, k−1);
       σ = σ · ⟨i, i+1⟩;
    else f1gen(i+1, 0);
end;
```

```
l2gen(i, k, t)

if i = n then output σ;
else
    if k ≠ 0 then
       f2gen(i+1, k, i+1);
       for j = i downto t
          σ = σ · ⟨j, j+1⟩;
       l2gen(i+1, k−1, t);
       for j = t to i
          σ = σ · ⟨j, j+1⟩;
    else f2gen(i+1, 0, i+1);
end;
```

An applet for the generation of theses permutation sets is available on the web site http://www.u-bourgogne.fr/ jl.baril/applet.html.

5 Concluding remarks

In this paper, we use the ECO method to show several algorithms for the generation of different sets of binary strings or pattern-avoiding permutations enumerated by the p-generalized Fibonacci or Lucas sequences. Each given algorithm runs in constant amortized time. We summarize our results in Table 1 where, for each studied set, we give the average complexity of our algorithm.

p-generalized Fibonacci and Lucas sets	Average complexity
$F_{p,n}$	$\mathcal{O}(1)$
$S_n(321, 312, 234\ldots(p+1)1)$	$\mathcal{O}(1)$
$S_n(231, 312, (p+1)p\ldots321)$	$\mathcal{O}(p)$
$K_{p,n}$	$\mathcal{O}(1)$
$L_{p,n}$	$\mathcal{O}(p)$
$S_n(321, 312, 234\ldots(p+1)1, T_p)$	$\mathcal{O}(1)$
$S_n(231, 312, (p+1)p\ldots321, T'_p)$	$\mathcal{O}(p)$

Table 1: Average complexity for each generating algorithm.

However, some problems are still left open. Indeed, can one generate these permutation sets in Gray code order? Some results concerning Fibonacci permutation sets are given in [16] but none are for Lucas permutations. Can one find other pattern-avoiding permutation sets enumerated by p-generalized Fibonacci and Lucas sequences? This will also be interesting to exhibit constructive bijection between the permutation sets found in this paper.

References

[1] S. Bacchelli, E. Barcucci, E. Grazzini and E. Pergola, *Exhaustive generation of combinatorial objects by ECO*, Acta Informatica, **40** (2004), 585–602.

[2] C. Banderier, P. Flajolet, D. Gardy, M. Bousquet-Melou, A. Denise, and D. Gouyou-Beauchamps, *Generating functions for generating trees*, Discrete Mathematics, **246** (2002), 29–55.

[3] E. Barcucci, A. Del Lungo and E. Pergola, *Random generation of trees and other combinatorial objects*, Theoretical Computer Science, **218** (2) (1999), 218–232.

[4] E. Barcucci, A. Del Lungo, E. Pergola and R. Pinzani, *A methodology for plane tree enumeration*, Discrete Mathematics, **180** (1998), 45–64.

[5] E. Barcucci, A. Del Lungo, E. Pergola and R. Pinzani, *Directed animals, forests of trees and permutations*, Discrete Mathematics, **204** (1999), 41–71.

[6] E. Barcucci, A. Del Lungo, E. Pergola and R. Pinzani, *ECO: a methodology for the enumeration of combinatorial objects*, Journal of Difference Equations and Applications, **5** (1999), 435–490.

[7] J.L. Baril and P.T. Do, *p-Generalized Fibonacci permutations generation by the ECO method*, CANT06, May 8-19, 2006.

[8] J.L. Baril and V. Vajnovszki, *Minimal change list for Lucas strings and some graph theoretic consequences*, Theoretical Computer Science, **346** (2005), 189–199.

[9] S. Brlek, E. Duchi, E. Pergola and S. Rinaldi, *On the equivalence problem for succession rules*, Discrete Math., **298** (2005), 142–154.

[10] E. Deutsch, L. Ferrari and S. Rinaldi, *Production matrices*, Advances in Applied Mathematics, **34** (2005), 101–122.

[11] E. Duchi, A. Frosini, R. Pinzani and S. Rinaldi, *A note on rational succession rules*, Journal of Integer Sequences, Article 03.1.7, 6, 2003.

[12] E. Egge and T. Mansour, *Restricted permutations, Fibonacci numbers, and k-generalized Fibonacci numbers*, Integers Electronic Journal of Combinatorial Number Theory, Article A01, **5** (2005).

[13] L. Ferrari, E. Pergola, R. Pinzani and S. Rinaldi, *An algebraic characterization of the set of succession rules*, Theoretical Computer Science, **281** (2002), 351–367.

[14] L. Ferrari, E. Pergola, R. Pinzani and S. Rinaldi, *Jumping succession rules and their generating functions*, Discrete Math., **271** (2003), 29–50.

[15] G. Firro and T. Mansour, *Restricted permutations and polygons*, The Third International Conference on Permutations Patterns, Gainsville, Florida., March 7–11, 2005.

[16] A. Juarna and V. Vajnovszki, *More about a Simion-Schmidt's bijection*, (Preprint Submitted).

[17] A. Juarna and V. Vajnovszki, *Isomorphism between classes counted by Fibonacci numbers*, Words, **36** (2005), 51–62, September 13–17.

[18] A. Del Lungo, A. Frosini and S. Rinaldi, *ECO method and the exhaustive generation of convex polyominoes*, DMTCS, pages 103–116, 2003.

[19] T. Mansour, *Permutations avoiding a pattern from S_k and at least two patterns from S_3*, Ars Combin., **62** (2002), 227–239.

[20] R. Simion and F.W. Schmidt, *Restricted permutations*, European J. Combin., **6** (1985), 383–406.

[21] J. West, *Generating trees and forbidden subsequences*, Discrete Mathematics, **157** (1-3) (1996), 363–374.