Minimal change list for Lucas strings and some graph theoretic consequences

Jean-Luc BARIL, Vincent VAJNOVSZKI
LE2I– CNRS UMR 5158, Université de Bourgogne
B.P. 47 870, 21078 DIJON-Cedex France
e-mail: {barjl}{vvajnov}@u-bourgogne.fr

Abstract

We give a minimal change list for the set of order p length-n Lucas strings, i.e., the set of length-n binary strings with no p consecutive 1's nor a 1^ℓ prefix and a 1^m suffix with $\ell + m \geq p$. The construction of this list proves also that the order p n-dimensional Lucas cube has a Hamiltonian path if and only if n is not a multiple of $p+1$, and its second power always has a Hamiltonian path.

keyword: minimal change list, Gray code, Fibonacci and Lucas string, Lucas cube, Hamiltonian path

1 Introduction

The Hamming distance between two strings in $\{0,1\}^n$ is the number of positions in which they differ. A *k-Gray code* for a set of binary strings $B \subset \{0,1\}^n$ is an *ordered* list $\mathcal{B}$ for B, such that the Hamming distance d between any two consecutive strings in $\mathcal{B}$ is at most k. In addition, if the list $\mathcal{B}$ minimizes both $k = max\ d(x,x')$ and $\sum d(x,x')$, where (x,x') ranges over all pairs of successive strings in $\mathcal{B}$, then it is called *minimal change list* or *minimal Gray code*. Obviously, a 1-Gray code is a minimal change list.

We call Hamiltonian a graph having a Hamiltonian path and a k-Gray code is a Hamiltonian path in $Q_n^k|B$, the restriction of the kth power of the hypercube Q_n to the set B.

The set $F_{n,p}$, of order p length-n Fibonacci strings, is the set of length-n binary strings such that there are no p consecutive 1's. The set $L_{n,p}$, of order p length-n Lucas strings, is the set of all strings in $F_{n,p}$ which do not begin by 1^ℓ and end by 1^m with $\ell + m \geq p$. In other words, $L_{n,p}$ is the set of length-n binary strings such that there are no 1^p factors if strings are regarded circularly, i.e., the last entry of a string is followed by the first one.

A number of papers concerning the Fibonacci and Lucas strings have been published [3, 6, 7, 8, 9, 13, 15]. In the present one we introduce an order relation on $\{0,1\}^n$ which induces a minimal change list on $L_{n,p}$. Note that this order relation yields a 1-Gray code on the set $F_{n,p}$, of order p length-n Fibonacci strings [13].

This paper is the extended version of [2] and the remaining is organized as follows. In the next section we prove that a 1-Gray code for $L_{n,p}$ is possible only if $(p+1)$ does not divide n. In Section 3 we give such a Gray code and a 2-Gray code when $(p+1)$ divides n; both of them are minimal change lists. Few graph theoretic consequences are presented in Section 4 and in the final part some algorithmic considerations are given.

2 Parity difference relation

For a binary string set B we denote by B' (resp. B'') the subset of B of strings with an odd (resp. even) number of 1's. Let $Q(L_{n,p})$ be the order p n-dimensional Lucas cube, i.e., the restriction of the hypercube Q_n to the set $L_{n,p}$. The graph $Q(L_{n,p})$ is bipartite, and with the notations above $\{L'_{n,p}, L''_{n,p}\}$

is a bipartition. No Hamiltonian path is possible in $Q(L_{n,p})$ (or equivalently, 1-Gray code for $L_{n,p}$) if $|card(L''_{n,p}) - card(L'_{n,p})| > 1$, i.e., the number of vertices in the two bipartitions differs by more than one.

The main result of this section is Theorem 1. In the following we suppose $p > 1$ fixed and more often in this section we will omit the subscript p for the sets $F_{n,p}$, $L_{n,p}$ and other related things.

Let $\{\phi_n\}_{n\geq 0}$ and $\{\lambda_n\}_{n\geq 0}$ be the parity difference integer sequences corresponding to Fibonacci and Lucas strings defined by

- $\phi_n = card(F''_n) - card(F'_n)$, and
- $\lambda_n = card(L''_n) - card(L'_n)$.

Lemma 1 *1. ϕ_n satisfies*

$$\phi_n = \phi_{n-1} - \phi_{n-2} + \cdots + (-1)^{p+1}\phi_{n-p}, \ \textit{for } n \geq p+1, \tag{1}$$

2. λ_n is related to ϕ_n by

$$\lambda_n = \phi_{n-2} - 2 \cdot \phi_{n-3} + \cdots + (-1)^{p+1} p \cdot \phi_{n-p-1}, \ \textit{for } n \geq p+2. \tag{2}$$

Proof 1. The recursive definition in [13]

$$F_n = 0 \cdot F_{n-1} \cup 10 \cdot F_{n-2} \cup 110 \cdot F_{n-3} \cup \ldots \cup 1^{p-1}0 \cdot F_{n-p}, \text{ for } n \geq p+1$$

can be expanded as

$$F'_n = 0 \cdot F'_{n-1} \cup 10 \cdot F''_{n-2} \cup 110 \cdot F'_{n-3} \cup \ldots,$$

and

$$F''_n = 0 \cdot F''_{n-1} \cup 10 \cdot F'_{n-2} \cup 110 \cdot F''_{n-3} \cup \ldots,$$

and (1) holds.

2. When $n \geq p+2$ the set L_n is the union of the sets in the table below, where the strings with prefix $1^{i-1}0$ are in the ith line and those with the suffix 01^{j-1} are in the jth column, $1 \leq i, j \leq p$.

$0 \cdot F_{n-2} \cdot 0,$	$0 \cdot F_{n-3} \cdot 01,$	$\cdots$	$0 \cdot F_{n-p} \cdot 01^{p-2},$	$0 \cdot F_{n-p-1} \cdot 01^{p-1}$
$10 \cdot F_{n-3} \cdot 0,$	$10 \cdot F_{n-p-4} \cdot 01,$	$\cdots$	$10 \cdot F_{n-p-1} \cdot 01^{p-2}$	
$\vdots$	$\vdots$	$\vdots$		
$1^{p-2}0 \cdot F_{n-p} \cdot 0,$	$1^{p-2}0 \cdot F_{n-p-1} \cdot 01$			
$1^{p-1}0 \cdot F_{n-p-1} \cdot 0$				

In this decomposition, each F_{n-k-1} occurs exactly k times, and reading this table diagonally one has

$$card(L_n) = \sum_{j=1}^{p} j \cdot card(F_{n-j-1}),$$

and F_{n-k-1} appears as $1^s0 \cdot F_{n-k-1} \cdot 01^t$, with $s + t = k - 1$, thus

$$card(L'_n) = card(F'_{n-2}) + 2 \cdot card(F''_{n-3}) + 3 \cdot card(F'_{n-4}) \ldots,$$

and

$$card(L''_n) = card(F''_{n-2}) + 2 \cdot card(F'_{n-3}) + 3 \cdot card(F''_{n-4}) \ldots,$$

so (2) holds. □

The next proposition gives the generating function for the sequences $\{\phi_n\}_{n\geq 0}$ (except its first term) and $\{\lambda_n\}_{n\geq 0}$ (except its $p+2$ first terms).

Proposition 1 *If $\phi(z)$ and $\lambda(z)$ denote the generating functions for the sequences $\{\phi_n\}_{n\geq 0}$ and $\{\lambda_n\}_{n\geq 0}$ respectively then*

1.

$$\phi(z) = \phi_0 + z \cdot (-z)^{p-1} \cdot \frac{1+z}{1-(-z)^{p+1}}, \tag{3}$$

2.

$$\lambda(z) = \sum_{j=0}^{p+1} \lambda_j z^j + z^{p+2} \cdot (-1)^{p+1} \cdot \frac{1-(p+1)(-z)^p + p(-z)^{p+1}}{(1-(-z)^{p+1}) \cdot (1+z)}. \tag{4}$$

Proof 1. For $i = 1, 2, \ldots, p-1$ all strings in $\{0,1\}^i$ are in F_i, and half of them are in F_i' and other half are in F_i'', so $\phi_1 = \phi_2 = \ldots = \phi_{p-1} = 0$. If $i = p$ then, a single string in $\{0,1\}^p$ does not belong to F_p, namely 1^p, so $\phi_p = (-1)^{p+1}$. By the relation (1) we have

$$\phi(z) = \phi_0 + z \cdot \frac{f(z)}{\frac{1-(-z)^{p+1}}{1+z}}$$

where $f(z) = (-z)^{p-1}$ is given by the values of $\phi_1, \phi_2, \ldots, \phi_p$. See for instance page 79 of Flajolet's and Sedgewick's seminal book [4].
2. If $\lambda^*(z) = \lambda(z) - \sum_{j=0}^{p+1} \lambda_j z^j$, then the relation (2) gives (see again [4])

$$\lambda^*(z) = (\phi(z) - \phi_0) \cdot (z^2 - 2z^3 + \cdots + (-1)^{p+1} p z^{p+1})$$

and finally

$$\lambda^*(z) = z^{p+2} \cdot (-1)^{p+1} \cdot \frac{1-(p+1)(-z)^p + p(-z)^{p+1}}{(1-(-z)^{p+1}) \cdot (1+z)}. \tag{5}$$

□

For $n \geq p+2$, λ_n is given by the coefficient of z^n in (4). The next corollary shows that the sequence $\{\lambda_n\}_{n\geq 0}$ is periodic from $n \geq p+1$ and gives its generating function if it is extended by periodicity to $\lambda_0, \lambda_1, \ldots \lambda_{p+1}$.

Corollary 1 *The sequence $\{\lambda_n\}_{n\geq p+2}$ has the period $2(p+1)$. In addition, if one defines $\lambda_n = \lambda_{n+2(p+1)}$ for all $n = 0, 1, \ldots, p+1$ then its generating function becomes*

$$\lambda(z) = \frac{(-z)^{p+1} + (p+1)z + p}{(1-(-z)^{p+1}) \cdot (1+z)}. \tag{6}$$

Proof. For $\lambda^*(z)$ given by relation (5) it is easy to show that $\frac{\lambda^*(z)}{z^{p+2}}(1 - z^{2(p+1)})$ is a polynomial of degree less than $2p+2$. Thus $\frac{\lambda^*(z)}{z^{p+2}}$ is the generating function for a periodic integer sequence with the period $2(p+1)$, and so is $\lambda(z)$ assuming it is extended by periodicity.

For (6) it is enough to find a polynomial $g(z)$ of degree less than $p+2$ such that $\lambda^*(z) - g(z)$ is divisible by $z^{2(p+1)}$, and in this case $\lambda(z) = \frac{\lambda^*(z)-g(z)}{z^{2(p+1)}}$. It is not hard to check that $g(z) = \frac{(-z)^{p+2}((-z)^p-1)}{z+1}$ satisfies this and we obtain (6). □

Corollary 2 *The parity difference integer sequence corresponding to Lucas strings satisfies*

$$\lambda_{n,p} = \begin{cases} (-1)^{n+1} & \text{if } (p+1) \nmid n, \\ (-1)^n \cdot p & \text{if } (p+1) | n. \end{cases} \tag{7}$$

Proof. $\lambda(z)$ in the relation (6) can be expressed as

$$\begin{aligned} \lambda(z) &= \frac{p+1}{1-(-z)^{p+1}} - \frac{1}{1+z} \\ &= (p+1) \cdot \sum_{k\geq 0} (-z)^{k(p+1)} - \sum_{n\geq 0} (-z)^n \end{aligned}$$

and $\lambda_{n,p}$ is the coefficient of z^n in $\lambda(z)$. □

Observe that the choice of $\lambda_{i,p}$, $i = 1, 2, \ldots, p+1$, in Corollary 1 seems arbitrary since extended by periodicity. In fact, for $i = 1, 2, \ldots, p$ all the 2^i strings in $\{0,1\}^i$ are in $L_{i,p}$, except 1^i which contains p consecutive 1's if strings are regarded circularly. In this case $\lambda_{i,p} = (-1)^{i+1}$, and similarly, $\lambda_{p+1,p} = (-1)^{p+1}p$, which are in accord with (7).

Theorem 1 *If $Q(L_{n,p})$ is Hamiltonian then $(p+1) \not| n$.*

Proof. If $Q(L_{n,p})$ is Hamiltonian then $|\lambda_{n,p}| \leq 1$ so $(p+1) \not| n$. □

The Hamiltonism of $Q(L_{n,p})$, when $(p+1) \not| n$, is shown constructively in the next section.

3 The Gray codes

We adopt the convention that lower case bold letters represent length-n binary strings, e.g., $\boldsymbol{x} = x_1 x_2 \ldots x_n$; and we use the same group of letters to denote a set A and an *ordered list* $\mathcal{A}$ for a set A. A list $\mathcal{A}$ for the set $A \subset \{0,1\}^n$ is equivalent to an order relation on A: $\boldsymbol{x} < \boldsymbol{y}$ iff $\boldsymbol{x}$ precedes $\boldsymbol{y}$ in $\mathcal{A}$. For example, if $\boldsymbol{x} \neq \boldsymbol{y} \in \{0,1\}^n$ and i is the leftmost position with $x_i \neq y_i$ then:

- the lexicographic order is given by: $\boldsymbol{x} < \boldsymbol{y}$ iff x_i is even $(= 0)$, and y_i is odd $(= 1)$,
- the *reflected Gray code order* due to Frank Gray in 1953 [5] is given by: $\boldsymbol{x} < \boldsymbol{y}$ iff $\sum_{j=1}^{i} x_j$ is even (and $\sum_{j=1}^{i} y_j$ is odd).

We say that an order relation $<$ on a set of strings induces a *k-Gray code* if the set listed in $<$ order yields a k-Gray code, i.e., successive strings differ in at most k positions. So, the reflected Gray code order above induces a 1-Gray code on $\{0,1\}^n$; and its restriction to the strings with fixed density (i.e., strings in $\{0,1\}^n$ with a constant number of 1's) induces a 2-Gray code, called *revolving door code* by Nijenhuis and Wilf [10].

According to [13] we recall the following definition which gives another order relation on binary strings and all of those presented here are particular cases of *genlex order* [14], that is, any set of strings listed in such an order has the property that strings with a common prefix are contiguous.

Definition 1 *We say that $\boldsymbol{x}$ is less than $\boldsymbol{y}$ in* local reflected order*, denoted by $\boldsymbol{x} \prec \boldsymbol{y}$, if $\sum_{j=1}^{i}(1-x_j)$ is odd and $\sum_{j=1}^{i}(1-y_j)$ is even, where i is the leftmost position with $x_i \neq y_i$.*

Remark 1

1. *$\boldsymbol{x} \prec \boldsymbol{y}$ iff the prefix $x_1 x_2 \ldots x_i$ contains an odd number of 0's,*
2. *$\boldsymbol{x} \prec \boldsymbol{y}$ iff $\overline{\boldsymbol{x}} > \overline{\boldsymbol{y}}$ in reflected Gray code order, with $\overline{\boldsymbol{x}}$ and $\overline{\boldsymbol{y}}$ the bitwise complement of $\boldsymbol{x}$ and $\boldsymbol{y}$,*
3. *As the reflected Gray code order, the local reflected order $\prec$ induces a 1-Gray code on $\{0,1\}^n$ and a 2-Gray code on length-n binary strings with fixed density,*
4. *In [13] it is shown that, unlike the reflected Gray code order, the local reflected order $\prec$ induces a 1-Gray code on the set $F_{n,p}$.*

The main result of this section is Corollary 3 and Theorem 2 which say that $\prec$ induces also a minimal Gray on the set $L_{n,p}$, of order p length-n Lucas strings.

If $\boldsymbol{z}$ is a length-k string, we denote by $\boldsymbol{z}^{\frac{n}{k}}$ the length-n prefix of the infinite string $\boldsymbol{zzz}\ldots$, or equivalently,

$$\boldsymbol{z}^{\frac{n}{k}} = \underbrace{\boldsymbol{zz}\ldots\boldsymbol{z}}_{\lfloor \frac{n}{k} \rfloor} \boldsymbol{z}_r,$$

where $r = n \mod k$, and $\boldsymbol{z}_r$ is the length-r prefix of $\boldsymbol{z}$. In the following, the length-$(p+1)$ binary string

$$\chi = \underbrace{1\dots1}_{p-1}00$$

plays a central role for our purposes.

Let $\mathcal{F}_{n,p}$ and $\mathcal{L}_{n,p}$ be the lists obtained by ordering the sets $F_{n,p}$ and $L_{n,p}$, respectively, by the relation $\prec$. In [13] it is proved that the first and the last strings of $\mathcal{F}_{n,p}$ are $first(\mathcal{F}_{n,p}) = 0\chi^{\frac{n-1}{p+1}}$ and $last(\mathcal{F}_{n,p}) = \chi^{\frac{n}{p+1}}$. The next lemma gives similar results for $\mathcal{L}_{n,p}$.

Lemma 2

1. $first(\mathcal{L}_{n,p}) = 0\chi^{\frac{n-1}{p+1}}$
2. $last(\mathcal{L}_{n,p}) = \chi^{\frac{n-1}{p+1}}0$.

Proof 1. Let $f_1f_2\dots f_n = 0\chi^{\frac{n-1}{p+1}}$ and $1 \le j \le n$ such that $\sum_{i=1}^{j}(1-f_i)$ is even, then: (1) $j > 0$ and $f_j = 0$, and (2) f_{j-1} is the rightmost 1 bit in a contiguous 1's sequence of length $p-1$ and, by Definition 1, $0\chi^{\frac{n-1}{p+1}}$ has no predecessor in $L_{n,p}$ in $\prec$ order. The proof of 2. is similar, the string $\chi^{\frac{n-1}{p+1}}0$ has no successor in $\mathcal{L}_{n,p}$. □

Now we describe how we compute the successor of a string in the lists $\mathcal{F}_{n,p}$ and $\mathcal{L}_{n,p}$. Since in the lists $\mathcal{F}_{n,p}$ and $\mathcal{L}_{n,p}$ strings with a common prefix are contiguous, the successor of $\boldsymbol{x} \in \mathcal{F}_{n,p}$ (resp. of $\boldsymbol{x} \in \mathcal{L}_{n,p}$) is given by changing the rightmost bit in $\boldsymbol{x}$ such that the obtained string remains in $\mathcal{F}_{n,p}$ (resp. in $\mathcal{L}_{n,p}$), and it is greater than $\boldsymbol{x}$ in $\prec$ order. More formally we have

Lemma 3 *Let $\boldsymbol{x} \ne last(\mathcal{F}_{n,p})$ and $s(\boldsymbol{x})$ its successor in $\mathcal{F}_{n,p}$, then either 1, 2 or 3 below holds.*

1. *$\boldsymbol{x}$ contains an odd number of 0's and it has not a suffix of the form $1^{p-1}0$. In this case $s(\boldsymbol{x}) = x_1\dots x_{n-1}(1-x_n)$.*
2. *$\boldsymbol{x}$ contains an even number of 0's and ends by $1^{p-1}0$. Then $s(\boldsymbol{x}) = x_1\dots x_{n-2}0x_n$.*
3. *$\boldsymbol{x}$ contains an even number of 0's and does not end by $1^{p-1}0$, or it contains an odd number of 0's and ends by $1^{p-1}0$. Let $x_1x_2\dots x_{k-1}x_k$ be the length minimal prefix of $\boldsymbol{x}$ with an odd number of 0's and such that $\boldsymbol{x} = x_1x_2\dots x_{k-1}x_k0\chi^{\frac{n-k-1}{p+1}}$ (with $\chi^{\frac{n-k-1}{p+1}}$ eventually empty). In this case $s(\boldsymbol{x}) = x_1x_2\dots x_{k-1}(1-x_k)0\chi^{\frac{n-k-1}{p+1}}$.*

Note that if $\boldsymbol{x}$ is like described in point 3 of Lemma 3 then the required prefix $x_1x_2\dots x_{k-1}x_k$ always exists. For example, in $\mathcal{F}_{6,3}$ we have: by point 1, $s(01001\mathbf{1}) = 01001\mathbf{0}$; by point 2, $s(0001\mathbf{1}0) = 0001\mathbf{0}0$; and by point 3, $s(1001\mathbf{00}) = 1001\mathbf{10}$ and $s(1\mathbf{0}0110) = 1\mathbf{1}0110$. In [13] a similar idea is used to compute, in constant time, the successor $s(\boldsymbol{x})$ of a Fibonacci string $\boldsymbol{x}$.

Obviously, if $\boldsymbol{x}$ is a Lucas string then $s(\boldsymbol{x})$ is a Fibonacci string but not necessarily a Lucas string too. The next proposition states that if we denote by $succ(\boldsymbol{x})$ the successor of $\boldsymbol{x} \in L_{n,p}$ in the list $\mathcal{L}_{n,p}$ then $succ(\boldsymbol{x})$ is either $s(\boldsymbol{x})$, $s^2(\boldsymbol{x}) = s(s(\boldsymbol{x}))$ or $s^3(\boldsymbol{x}) = s(s(s(\boldsymbol{x})))$.

Proposition 2 *Let $\boldsymbol{x} \in L_{n,p}$, $\boldsymbol{x} \ne last(L_{n,p})$ and $s(\boldsymbol{x}) \notin L_{n,p}$ then either 1 or 2 below holds.*

1. *$(p+1)|n$ and $\boldsymbol{x} = 1^k00\chi^{\frac{n-k-2}{p+1}}$ with $0 \le k < p-1$. In this case*

$$\begin{aligned} succ(\boldsymbol{x}) &= 1^{k+1}0\chi^{\frac{n-k-3}{p+1}}0 \\ &= s^2(\boldsymbol{x}). \end{aligned}$$

2. *$\boldsymbol{x} = 1^k\boldsymbol{z}1^\ell0$ with $0 < k, \ell < p-1$ and $k+\ell \ge p-1$. In this case*

$$\begin{aligned} succ(\boldsymbol{x}) &= 1^k s(\boldsymbol{z})1^\ell0 \\ &= s^3(\boldsymbol{x}). \end{aligned}$$

Table 1: The lists $\mathcal{L}_{4,2}$ and $\mathcal{L}_{4,3}$. Changed bits are in bold-face.

$\mathcal{L}_{4,2}$	$\mathcal{L}_{4,3}$
0 1 0 **0**	0 1 **1** 0
0 **1** 0 1	0 1 0 **0**
0 0 0 **1**	0 **1** 0 1
0 0 **0** 0	0 0 0 **1**
0 0 1 0	0 0 **0** 0
1 0 **1** 0	0 0 1 **0**
1 0 0 0	**0** 0 1 **1**
	1 0 **1** 0
	1 0 0 **0**
	1 **0** 0 **1**
	1 1 0 0

Proof. When $s(\boldsymbol{x})$ is not a Lucas string then it is obtained from $\boldsymbol{x}$ by changing the 0 bit which either follows a 1's prefix or is in the last position. □

Remark 2

1. *If $(p+1)|n$ then there exist exactly $p-1$ Lucas strings as in point 1 of Proposition 2, one for each k, $0 \leq k < p-1$. In addition, if $\boldsymbol{x}$ is such a string and d denotes the Hamming distance then*

   (a)

$$\begin{aligned} d(\boldsymbol{x}, succ(\boldsymbol{x})) &= d(\boldsymbol{x}, s^2(\boldsymbol{x})) \\ &= 2, \end{aligned}$$

   (b) the string $\boldsymbol{v} = 1^k 00\chi^{\frac{n-k-3}{p+1}}0 \in L_{n,p}$ is the predecessor of $\boldsymbol{x}$ in $\prec$ order (i.e., $succ(\boldsymbol{v}) = \boldsymbol{x}$) and $d(\boldsymbol{v}, succ(\boldsymbol{x})) = 1$. (In fact $\boldsymbol{v} = \boldsymbol{x} \cdot succ(\boldsymbol{x})$, the bitwise product of $\boldsymbol{x}$ and $succ(\boldsymbol{x})$.)

2. *If $\boldsymbol{x}$ is a Lucas string as in point 2 of Proposition 2 then*

$$\begin{aligned} d(\boldsymbol{x}, succ(\boldsymbol{x})) &= d(1^k\boldsymbol{z}1^\ell 0, 1^k\boldsymbol{z}'1^\ell 0) \\ &= d(\boldsymbol{z}, \boldsymbol{z}') \\ &= 1. \end{aligned}$$

This remark proves the following

Corollary 3

1. *If $(p+1) \not| n$ then $\mathcal{L}_{n,p}$ is a 1-Gray code for $L_{n,p}$.*
2. *If $(p+1)|n$ then $\mathcal{L}_{n,p}$ is a 2-Gray code for $L_{n,p}$ and there are exactly $p-1$ strings with $d(\boldsymbol{x}, succ(\boldsymbol{x})) = 2$.*

Theorem 2 *$\mathcal{L}_{n,p}$ is a minimal change list for the set $L_{n,p}$.*

Proof. By Corollary 2, there are no more restrictive Gray code as $\mathcal{L}_{n,p}$. □
See Table 1 and Figure 1.(b) for the list $\mathcal{L}_{4,2}$ and $\mathcal{L}_{4,3}$.

4 Graph theoretic issues

Here we present some graph theoretic consequences of the previous results.

4.1 Hamiltonicity

For a graph G let G^k be its kth power, where edges connect vertices which are linked by a path in G of length at most k. A k-Gray code for $B \subset \{0,1\}^n$ is a Hamiltonian path for $Q_n^k|B$, the restriction of the kth power of the hypercube to the set B. Generally, $(Q_n|B)^k$ is a subgraph of $Q_n^k|B$ and equality holds if Hamming distance and shortest path length coincide on $Q_n|B$; this is the case for $B = F_{n,p}$ or $L_{n,p}$, but not for Dyck words for instance, since the restriction of Q_n to length-n Dyck words is not a connected graph.

As in Section 2, let $Q(L_{n,p}) = Q_n|L_{n,p}$ denote the Lucas cube. Corollary 3 says that $\mathcal{L}_{n,p}$ is a Hamiltonian path in:

- the Lucas cube iff $(p+1) \not| n$,
- the second power of the Lucas cube elsewhere. (In general, $Q(L_{n,p})$ is not 2-connected as it may be checked for $L_{3,2}$, and so, the Hamiltonicity of its second power can not be obtained trivially from a well-known result in graph theory.)

Corollary 4 *If* $(p+1)|n$ *then*

1. *The minimal number of paths covering* $Q(L_{n,p})$ *is* p.
2. *The length of the maximal path in* $Q(L_{n,p})$ *is* $card(L_{n,p}) - p + 1$.

Proof. The point 1 follows from Corollary 3. For the point 2, from Remark 2 results that by bypassing in $\mathcal{L}_{n,p}$ the $p-1$ strings $\boldsymbol{y} = succ(\boldsymbol{x})$, with $d(\boldsymbol{x}, \boldsymbol{y}) = 2$, we obtain a maximal length path in $Q(L_{n,p})$. □

When a graph does not have a Hamiltonian path it may be desirable to visit each vertex but not necessarily once, such that the Hamming distance between two successive vertices is one. Following [12], a graph is in the class $\mathcal{H}(s,t)$ if it has a path that visits every vertex at least s times and at most t times, and such a path is called $\mathcal{H}(s,t)$-path. Thus a graph is in $\mathcal{H}(1,1)$ exactly if it is Hamiltonian. In this context, we have

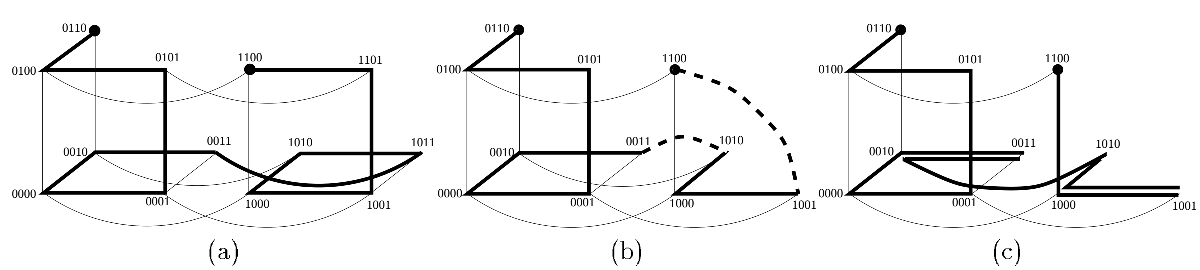
Corollary 5 *If* $(p+1)|n$ *then* $Q(L_{n,p})$ *is in* $\mathcal{H}(1,2)$.

Proof. When $(p+1)|n$, by point 1(a) of Remark 2, in the list $\mathcal{L}_{n,p}$ there are strings $\boldsymbol{x}$ which differ from $succ(\boldsymbol{x})$ in two positions. For each such string, we insert $\boldsymbol{v}$—the predecessor of $\boldsymbol{x}$, see point 1(b) of Remark 2,— between $\boldsymbol{x}$ and $succ(\boldsymbol{x})$ and one obtains an $\mathcal{H}(1,2)$ path in $Q(L_{n,p})$.

□

Notice that in the $\mathcal{H}(1,2)$ paths above exactly $p-1$ strings are visited twice, and this is optimal. See Figure 1.(c) for an $\mathcal{H}(1,2)$ path in $Q(L_{4,3})$.

Figure 1: (a) The Hamiltonian path $\mathcal{F}_{4,3}$ in $Q(F_{4,3})$. (b) The *'path'* $\mathcal{L}_{4,3}$ in $Q(L_{4,3})$, dashed arcs connect distance-2 vertices. (c) An $\mathcal{H}(1,2)$-path in $Q(L_{4,3})$.



4.2 Some structural properties

Let now recall some definitions concerning a connected graph G with vertices set V:

- the eccentricity of a vertex v is $e(v) = \max_{u \in V} d(u, v)$,
- the diameter of G is $diam(G) = \max_{u,v \in V} d(u, v) = \max_{v \in V} e(v)$,
- the radius of G is $rad(G) = \min_{v \in V} e(v)$,
- and the center of G is $Z(G) = \{u \in G \mid e(u) = rad(G)\}$.

The next proposition generalizes similar results presented in [9] for $L_{n,2}$.

Proposition 3 *Let $n \geq 1$, $p \geq 2$.*

1. $diam(Q(L_{n,p})) = \begin{cases} n-1 & \text{if } p = 2 \text{ and } n \text{ odd,} \\ n & \text{otherwise,} \end{cases}$
2. $e(0^n) = n - \lceil \frac{n}{p} \rceil$,
3. $rad(Q(L_{n,p})) = n - \lceil \frac{n}{p} \rceil$ *and* $Z(Q(L_{n,p})) = \{0^n\}$.

Proof 1. Let $\boldsymbol{x} = 1010\ldots$ ($x_i = 1$ iff i is odd) and $\boldsymbol{y} = 0101\ldots$ ($y_i = 1$ iff i is even). If $p = 2$ and n is even or if $p > 2$ then $\boldsymbol{x}, \boldsymbol{y} \in L_{n,p}$ and they are at maximal distance equal to n. If $p = 2$ and n is odd there does not exist a pair of strings in $L_{n,p}$ at distance n and it is easy to find two strings at distance $n-1$.
2. Let $\boldsymbol{v} = 1^{p-1}0\ldots1^{p-1}01^k0$ with $0 \leq k \leq p-1$. Clearly, $\boldsymbol{v} \in L_{n,p}$ and it has a minimum number of 0's, namely $\lceil \frac{n}{p} \rceil$, or equivalently, a maximal number of 1's, equal to $n - \lceil \frac{n}{p} \rceil$, So, $e(0^n) = d(0^n, \boldsymbol{v}) = n - \lceil \frac{n}{p} \rceil$. The same result can be obtained by replacing $\boldsymbol{v}$ by any of its circular shifts.
3. For any Lucas string $\boldsymbol{u} \neq 0^n$ we construct another Lucas string at distance greater than $n - \lceil \frac{n}{p} \rceil$ to $\boldsymbol{u}$. The bitwise product $\boldsymbol{v} \cdot \boldsymbol{x}$ of a Lucas string $\boldsymbol{v}$ by a binary (not necessarily Lucas) string $\boldsymbol{x}$ is a Lucas string and $d(\boldsymbol{x}, \boldsymbol{v} \cdot \boldsymbol{x})$ equals the number of positions i where $v_i = 0$ and $x_i = 1$.

Let $\boldsymbol{u} \in L_{n,p}$, $\boldsymbol{u} \neq 0^n$, $\overline{\boldsymbol{u}}$ its bitwise complement and $\boldsymbol{v}$ a circular permutation of the string $1^{p-1}0\ldots1^{p-1}01^k0$ given in point 2, such that $\overline{\boldsymbol{u}}$ and $\boldsymbol{v}$ have at least one 0 in the same position. $\boldsymbol{v} \cdot \overline{\boldsymbol{u}} \in L_{n,p}$ and since $\boldsymbol{v}$ has exactly $\lceil \frac{n}{p} \rceil$ 0's and at least one of them corresponds to a 0 in $\overline{\boldsymbol{u}}$ one has $d(\overline{\boldsymbol{u}}, \boldsymbol{v} \cdot \overline{\boldsymbol{u}}) < \lceil \frac{n}{p} \rceil$. By the triangle inequality

$$d(\boldsymbol{u}, \boldsymbol{v} \cdot \overline{\boldsymbol{u}}) + d(\boldsymbol{v} \cdot \overline{\boldsymbol{u}}, \overline{\boldsymbol{u}}) \geq d(\boldsymbol{u}, \overline{\boldsymbol{u}}) = n$$

and so,

$$d(\boldsymbol{u}, \boldsymbol{v} \cdot \overline{\boldsymbol{u}}) \geq n - d(\overline{\boldsymbol{u}}, \boldsymbol{v} \cdot \overline{\boldsymbol{u}}) > n - \lceil \frac{n}{p} \rceil.$$

□

A *stable set* of a graph is a subset of vertices such that there are not two adjacent vertices and the *stability number*, denoted by $\alpha(G)$, is the number of vertices in a stable set of maximum cardinality.

Proposition 4 $\alpha(Q(L_{n,p})) = \max\left(card(L'_{n,p}), card(L''_{n,p})\right)$ *where* $\{L'_{n,p}, L''_{n,p}\}$ *is the bipartition of* $L_{n,p}$.

Proof. Let A be the set of maximum cardinality between $L'_{n,p}$ and $L''_{n,p}$, and B the other set. Clearly, A is stable and we will prove that any stable set has at most the same cardinality as A. By Corollaries 2 and 3, the successor $succ(\boldsymbol{x})$ of $\boldsymbol{x}$ in the list $\mathcal{L}_{n,p}$, induces an injective function $succ : B \to A$; in addition, $\boldsymbol{x} \in B$ and $succ(\boldsymbol{x}) \in A$ are connected in $Q(L_{n,p})$. Thus, if a stable set contains $\boldsymbol{x} \in B$ then it must not contain $succ(\boldsymbol{x}) \in A$, and so the cardinality of a stable set does not excede that of A. □

5 Algorithmic considerations

In [13] is given an exhaustive generating algorithm for the list $\mathcal{F}_{n,p}$ which runs with constant delay between any two successive strings. Now we show how one can modify this algorithm in order to produce efficiently the list $\mathcal{L}_{n,p}$.

Proposition 2 guarantees that at most two Fibonacci strings exist between any two consecutive Lucas string in $\mathcal{L}_{n,p}$; and by point 2 of Lemma 2 the last string in $\mathcal{L}_{n,p}$ is followed by at most one Fibonacci string. So, the algorithm in [13] can be modified to generate the list $\mathcal{L}_{n,p}$ by simply bypassing the Fibonacci strings which are not Lucas strings. The obtained algorithm inherits the constant delay property if one can decide, in constant time, if the current generated Fibonacci string is also a Lucas string. Additional variables, as: (1) the length of the contiguous prefix of 1's and (2) the length and the first position of the rightmost contiguous sequence of 1's, can be used to distinguish in constant time Fibonacci from Lucas strings.

Acknowledgments

We thank Olivier Togni who presented us the problem we deal with in this paper and for some technical suggestions.

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] J.-L. Baril and V. Vajnovszki, Gray codes for order p Lucas strings, in *Proceedings of the 4th International Conference on Combinatorics on Words*, September 2003, Turku, Finland (ed. Tero Harju and Juhani Karhumäki), 149-158. TUCS General Publication, No 27, August 2003.

[3] E. Dedó, D. Torri and N. Zagaglia Salvi, The observability of the Fibonacci and the Lucas cubes, *Combinatorics '98 (Palermo). Discrete Math.* **255**(1-3)(2002), 55 –63.

[4] P. Flajolet and R. Sedgewick, *Introduction à l'analyse des algorithmes*, Thomson Publishing, 1996.

[5] F. Gray, Pulse code communication, U.S. Patent 2632058 (1953).

[6] W-J. Hsu, Fibonacci cubes – a new interconnection topology, *IEEE Transactions on Parallel and Distributed Systems* **4**(1) (1993), 3–12.

[7] J.C. Lagarias and D.P. Weisser, Fibonacci and Lucas cubes, *Fibonacci Quart.* **19**(1)(1981), 39–43.

[8] J. Liu, W-J. Hsu and M.J. Chung, Generalized Fibonacci cubes are mostly Hamiltonian, *Journal of Graph Theory* **18**(8) (1994), 817–829.

[9] E. Munarini, C. Perelli Cippo and N. Zagaglia Salvi, On the Lucas cubes, *Fibonacci Quart.* **39**(1)(2001), 12 –21.

[10] A. Nijenhuis and H.S. Wilf, *Combinatorial Algorithms for Computers and Calculators*, Academic Press, 1978.

[11] J. Pallo, Enumerating, ranking and unranking binary trees, *The Computer Journal* **29**(1986), 171-175.

[12] G. Pruesse and F. Ruskey, Generating linear extensions fast, *SIAM J. Comput.* **23**(1994), 373–386.

[13] V. VAJNOVSZKI, A loopless generation of bitstrings without p consecutive ones, *Discrete Mathematics and Theoretical Computer Science*, Springer 2001, 227–240.

[14] T.R. WALSH, Generating Gray codes in $O(1)$ worst-case time per word, *DMTCS03, Lect. Notes Comput. Sci. 2731*, Springer 2003, 72-88.

[15] J. WU, Extended Fibonacci cubes, *IEEE Transactions on Parallel and Distributed Systems* **8**(12)(1997), 1203–1210.