UNIFORM RANDOM GENERATION OF (R,S)-FIBONACCI TILINGS

D. Merlini
Dipartimento di Sistemi e Informatica - Università di Firenze
merlini@dsi.unifi.it

R. Sprugnoli
Dipartimento di Sistemi e Informatica - Università di Firenze
sprugnoli@dsi.unifi.it

F. Uncini
Dipartimento di Sistemi e Informatica - Università di Firenze
uncini@dsi.unifi.it

M. C. Verri
Dipartimento di Sistemi e Informatica - Università di Firenze
verri@dsi.unifi.it

Abstract

In this paper we study the problem of tiling a strip of dimensions $rs \times n$ by using rectangles $r \times s$ with $r < s$ and r, s relatively prime. We use a generating tree approach to construct the tilings and prove that they are counted by the n^{th} (r, s)-Fibonacci number. This construction is done in terms of *prime components* and, by studying the tilings with respect to the length of the strip and to the number of prime components, we are able to give an algorithm which uniformly generate a random tiling of length n in time $O(n)$, where the constant multiplying n is strictly less then 1.

1. *Introduction*

The problem of covering a strip with a given set of pieces is well-known, and is widely treated for example in [2, 6, 9, 10]. In particular, the tiling problem which only uses two types of pieces is very interesting because of its relation with physical models. In these models the pieces can represent two different types of gas molecules, and the tiling is related to their disposition in a certain space (see, e.g., [7, 15]).

A characterization of this type of problem which only uses a vertical and a horizontal dimer will be called $(1,2)-$Fibonacci tiling. This name, as we will show later, is due to the fact that, once fixed the strip length n, the total number of tilings is given by the n^{th} $(1,2)-$Fibonacci number.

In general, we define the $(r,s)-$Fibonacci numbers, as the sequence given by the following recurrence relation:

$$\begin{cases} F_i^{[r,s]} = 0, & 0 < i < s, \text{ and } i \neq jr \\ F_i^{[r,s]} = 1, & 0 < i < s, \text{ and } i = jr, 0 \leq j \leq \lfloor \frac{s}{r} \rfloor \\ F_n^{[r,s]} = F_{n-r}^{[r,s]} + F_{n-s}^{[r,s]} & \text{otherwise.} \end{cases} \tag{1}$$

The $(1,2)-$Fibonacci tilings, and, more in general, the $(1,s)-$Fibonacci tilings have been widely studied (see, e.g., sequences A000045, A000930, A003269, A003520, A005708, A005709 and A005710 in [14]). In this paper, we make a generalization by studying the $(r,s)-$Fibonacci tiling problem which consists in tiling a strip of dimensions $rs \times n$ by using rectangles $r \times s$ with $r < s$ and r, s relatively prime.

In [10], it is proved that every strip tiling problem corresponds to a regular language. More precisely, the paper presents an algorithm which takes as input the pieces and the height of the strip and returns a regular grammar which defines the tilings with respect to their length. By applying Schützemberger methodology [12], or, equivalently, the symbolic method [13], it is then possible to find generating functions counting the number of tilings with respect to various parameters as the length and the number of pieces. The grammar presented in [10] builds each tiling by adding a piece after another.

In this paper we present a different approach, which makes use of generating trees and builds the various tilings in terms of *prime components*, that is, subtilings which cannot be further decomposed. In particular, we prove that the number of tilings of length n with k prime components is given by $f_{n,k}^{[r,s]} = \binom{k}{\frac{n-rk}{s-r}}$ and that $\sum_{k=0}^{n} f_{n,k}^{[r,s]} = F_n^{[r,s]}$, that is, the number of tilings of length n is given by the n^{th} (r,s)-Fibonacci number.

The approach in terms of prime components allows us to generate a random tiling of length n in time $O(n)$, where the constant multiplying n is strictly less then 1. We first observe that the probability that a tiling of length n has k prime components is $p_k = \frac{f_{n,k}^{[r,s]}}{F_n^{[r,s]}}$. Moreover, the mean number of prime components in a tiling of length n is $\mu_n^{[r,s]} = C \cdot n$ with $C < 1$ and the number of prime components is concentrated near the mean. Fixed the length n of the strip, the idea is to uniformly generate the number

k of prime components with probability p_k. In this paper, as we explain in Section 2, we study a tiling problem having only two kinds of prime components, thus, once k has been generated, the problem can be reduced to the random generation of a bit sequence of length k with a suitable number of $0'$s and $1'$s. The uniform random generation of the number k of prime components is quite difficult and we solve the problem by using the normal distribution to approximate p_k.

We conclude our paper by giving the results of some simulations of the algorithm. In particular, by using the χ^2 test [8], we give empirical evidence of the fact that the obtained generation is uniform.

2. *The tiling generating tree approach*

In this section we will show a correspondence between tilings and *generating trees.* A generating tree is a rooted labeled tree with the property that if v_1 and v_2 are any two nodes with the same label then, for each label l, v_1 and v_2 have exactly the same number of children with label l. To describe a generating tree it therefore suffices to specify the label of the root and a set of rules explaining how to derive from the label of a parent node the labels of all of its children.

Generating trees are a device to represent the development of many classes of combinatorial objects which can then be enumerated by counting the different labels in the various levels of the tree: recent references in this sense are [3, 11].

In order to use generating trees to represent a tiling problem, we need to define a *prime component:* every tiling can be decomposed into a juxtaposition of some completely filled subtilings which cannot be further decomposed; these subtilings are called *prime components* of the tiling.

We can represent a tiling problem as a tree of degree p, where p is the total number of prime components of the tiling. Each node is labeled with a value corresponding to the length of the covered strip; the root, corresponding to an empty tiling, is labeled with 0.

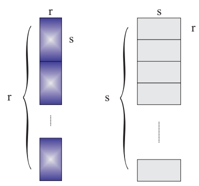

Figure 1: The two prime components for the $(r, s)-$Fibonacci tiling problem.

Let r and s be two relatively prime numbers, with $r < s$: we want to cover a strip of height rs with pieces having dimensions $r \times s$ or $s \times r$. In this case we only have 2 prime components: the first is composed by r pieces placed vertically and the other by s pieces placed horizontally (see Figure 1). By adding a prime component of the first kind, we increase by r the length of the strip; in the other case the length increases by s.

If r, s are not relatively prime numbers, let $d = GCD(r, s)$ be their greatest common divisor. In this case we can simply study the problem with $r_1 = r/d$, $s_1 = s/d$ and a strip with height rs/d^2.

The rule for the corresponding generating tree can be specified as follows:

$$\begin{cases} (0) \\ (k) \to (k+r)(k+s) \end{cases} \tag{2}$$

At level 1 of this tree, we have a node with label r and a node with label s which correspond to the two prime components; in general, at level j the labels of the nodes correspond to the length of the tilings obtained by adding a prime components to the tilings at level $j - 1$. In Figure 2 we show the first levels of the generating tree corresponding to $r = 1$ and $s = 2$.

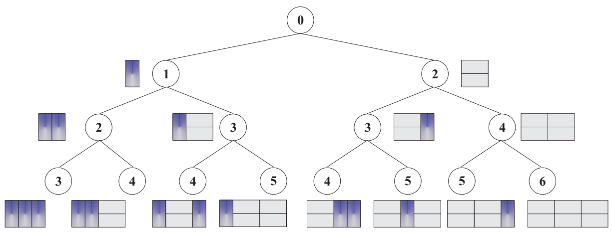

Figure 2: The generating tree and the corresponding tilings when $r = 1$ and $s = 2$.

We can now introduce another concept related to the structure of a generating tree. An infinite matrix $A = \{a_{k,n}\}_{k,n\in\mathbf{N}}$ is said to be *associated* to a generating tree with root (c) if $a_{k,n}$ is the number of nodes at level k with label $n + c$ (by convention, the level of the root is 0). In the present case, $a_{k,n}$ counts the number of tilings of length n composed by k prime components.

THEOREM 2.1 *The bivariate generating function of the matrix associated to the specification rule (2) is:*

$$F^{[r,s]}(t, z) = \frac{1}{1 - z(t^r + t^s)}. \tag{3}$$

n/k	0	1	2	3	4	5	6
0	1	0	0	0	0	0	0
1	0	1	1	0	0	0	0
2	0	0	1	2	1	0	0
3	0	0	0	1	3	3	1
4	0	0	0	0	1	4	6
5	0	0	0	0	0	1	5
6	0	0	0	0	0	0	1

n/k	0	1	2	3	4	5	6
0	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0
2	0	1	1	0	0	0	0
3	0	0	2	1	0	0	0
4	0	0	1	3	1	0	0
5	0	0	0	3	4	1	0
6	0	0	0	1	6	5	1

Table 1: Matrices A (on the left) and F (on the right) when $r = 1$ and $s = 2$.

In this function, the variables z and t count the length of the strip and the number of prime components respectively; that is, $a_{k,n} = [z^k t^n]F^{[r,s]}(t, z)$.

Proof: The generating function counting the number of prime components is $P^{[r,s]}(t) = t^r + t^s$ since we only have two prime components: one increases the length of the strip by r and the other by s. The expression for $F^{[r,s]}(t, z)$ follows immediately by a simple application of the symbolic method (see, for example, [13]). $\square$

In matrix A the column index denotes the length of the covered strip, so tilings with the same length are distributed onto different rows. In this paper we consider strips of a fixed length n and for this reason we study in details the transpose $F = A^T$ of the matrix A. Table 2 illustrates matrices A and F in the case $r = 1$ and $s = 2$. Obviously $f_{n,k} = a_{k,n}$ and we have the following:

THEOREM 2.2 *The number of tilings of length n composed by k prime components is:*

$$f_{n,k} = \begin{cases} \binom{k}{\frac{n-rk}{s-r}} & \textit{if } n - rk = j(s-r), 0 \le j \le k \\ 0 & \textit{otherwise}. \end{cases} \tag{4}$$

Proof: We have:

$$f_{n,k} = [t^n z^k]\frac{1}{1 - z(t^r + t^s)} = [t^n z^k]\sum_{j=0}^{\infty} z^j (t^r + t^s)^j = [t^n](t^r + t^s)^k =$$

$$= [t^n]t^{rk}(1 + t^{s-r})^k = [t^{n-kr}](1 + t^{s-r})^k = \binom{k}{\frac{n-rk}{s-r}}.$$

The lower value of the previous binomial must be an integer value, so we derive the statement of the Theorem. $\square$

Let $F_n^{[r,s]} = \sum_{k=0}^{n} f_{n,k}$ be the number of tilings of length n, then the numbers $F_n^{[r,s]}$ can be found by extracting the $[t^n]$ coefficient from $F_n^{[r,s]}(t,1) = F^{[r,s]}(t) = 1/(1-t^r-t^s)$.

We can easily observe that the $F_n^{[r,s]}$ satisfy the recurrence relation (1) given in the Introduction. In fact we have $(1-t^r-t^s)F(t) = 1$, hence:

$$[t^n]F(t) - [t^{n-r}]F(t) - [t^{n-s}]F(t) = \delta_{n,0}. \tag{5}$$

From this equation we find the recurrence relation: $F_n^{[r,s]} = F_{n-r}^{[r,s]} + F_{n-s}^{[r,s]}$, with $n \geq s$. The initial conditions in (1) can be easily found by specializing (5) when $n < s$. This recurrence relation generalizes the well-known recurrence for Fibonacci numbers and this explains the name $(r,s)-$Fibonacci tilings and numbers. The asymptotic behavior of (r,s)-Fibonacci numbers is therefore directly related to the zeroes of the denominator $d(t) = 1 - t^r - t^s$ of function $F^{[r,s]}(t)$. The $F_n^{[r,s]}$'s are obviously nonnegative, then the smallest zero in modulus of $d(t)$ has to be real, positive and smaller than 1. This sentence is supported by the fact that sequences are increasing and it can be proved directly by observing that $d(0) = 1$ and $d(1) = -1$.

THEOREM 2.3 *The dominant singularity of $F^{[r,s]}(t)$ can be approximated as follows:*

$$\rho \approx 1 - \frac{1}{r+s} - \frac{r^2+s^2-r-s}{r^3+s^3+4r^2s+4rs^2-r^2-s^2+4rs+2r+2s}.$$

Proof: First observe that if we set to $1-\epsilon$ the zero we are looking for, and substitute it into $d(t)$, we find:

$$1 - \left(1 - s\epsilon + \frac{s(s-1)}{2}\epsilon^2 + \cdots\right) - \left(1 - r\epsilon + \frac{r(r-1)}{2}\epsilon^2 + \cdots\right) =$$

$$= -1 + (s+r)\epsilon - \frac{s(s-1)+r(r-1)}{2}\epsilon^2 + \cdots.$$

If we imagine that ϵ is small, this expression being almost 0, we should have $\epsilon \approx 1/(r+s)$, and this gives us a first approximation to the zero we are looking for, that is $z_0 \approx 1 - 1/(r+s)$. We are now in a position to apply Newton's method to improve this approximation. We have:

$$z_1 = z_0 + \frac{1 - z_0^r - z_0^s}{rz_0^{r-1} + sz_0^{s-1}}. \tag{6}$$

The powers of z_0 can be computed as follows:

$$z_0^m \approx 1 - \frac{m}{r+s} + \frac{m(m-1)}{2(r+s)^2} + \cdots \tag{7}$$

and by using (6) and (7) we immediately have the statement of the theorem. □

We explicitly observe that ρ has multiplicity one. In fact we have:

$$r\rho^r + s\rho^s > r\rho^r + r\rho^s = r(\rho^r + \rho^s) = r \geq 1; \tag{8}$$

hence $d'(\rho) \neq 0$.

THEOREM 2.4 *The $(r, s)-$Fibonacci numbers can be approximated as follows:*

$$F_n^{[r,s]} \approx \frac{1}{r\rho^r + s\rho^s} \frac{1}{\rho n}.$$

Proof: We can express the generating function of $(r, s)-$Fibonacci numbers in terms of its dominant singularity: $1/(1 - t^r - t^s) \approx B/(1 - t/\rho)$. The constant B can be obtained by computing the following limit:

$$B = \lim_{t \to \rho} \frac{1 - \frac{t}{\rho}}{1 - t^r - t^s} = \lim_{t \to \rho} \frac{1/\rho}{rt^{r-1} + st^{s-1}} = \frac{1}{r\rho^r + s\rho^s}.$$

So, the desired value is:

$$F_n^{[r,s]} \approx [t^n] \frac{B}{1 - t/\rho} = B \left(\frac{1}{\rho}\right)^n = \frac{1}{r\rho^r + s\rho^s} \frac{1}{\rho n}.$$

□

3. *Uniform random generation of tilings*

In this section we present an efficient algorithm for the uniform random generation of $(r, s)-$Fibonacci tilings of length n. The random generation procedure performs in time $O(n)$, where the constant multiplying n is strictly less then 1: in fact, a tiling is generated in a time proportional to the number of prime components which is considerably less than its length.

To support this sentence we start by computing the mean number of prime components. In the following formulas the indexes r, s are omitted for ease of notation.

THEOREM 3.1 *The mean number of prime components of an $(r, s)-$Fibonacci tiling of length n is:*

$$\mu_n = \frac{n}{\rho P'(\rho)} + O(1)$$

where $P(t) = t^r + t^s$, $P'(t)$ is the derivative of $P(t)$ and ρ is the dominant singularity of $F(t)$.

Proof: The mean number of prime components is given by:

$$\mu_n = \frac{\sum_{k=0}^{n} k f_{n,k}}{\sum_{k=0}^{n} f_{n,k}}.$$

Since the bivariate generating function of $f_{n,k}$ is $F(t, z) = 1/(1 - zP(t))$ we can apply Proposition 9.1 in [5, p. 17], and prove the Theorem. □

COROLLARY 3.2 *Let $C = \frac{1}{\rho P'(\rho)}$ be the multiplicative constant in Theorem (3.1). Then C is lower than 1 and can be approximated as follows:*

$$C \approx \frac{r+s}{2rs} \quad r, s > 0 \textit{ and } r < s.$$

Proof: The first part of the Corollary follows immediately from (8). Let us now prove the second part; we have:

$$\frac{1}{C} = r\rho^r + s\rho^s = r\rho^r + s\rho^s + r\rho^s - r\rho^s = r(\rho^r + \rho^s) + (s-r)\rho^s = r + (s-r)\rho^s =$$

$$= r + (s-r)(1 - \frac{s}{s+r} + \cdots) \approx r + (s-r)\frac{r}{s+r} = r(1 + \frac{s-r}{s+r}) = \frac{2rs}{s+r}$$

as desired. □

THEOREM 3.3 *The variance of the number of prime components composing an $(r, s)-$Fibonacci tiling of length n is:*

$$\sigma_n^2 = \frac{n\rho P''(\rho) + P'(\rho) - \rho P'(\rho)^2}{\rho^2 P'(\rho)^3} + O(1)$$

where $P(t)$ is the same as in the Theorem (3.1), $P'(t), P''(t)$ are the first and the second derivatives of $P(t)$ and ρ is the dominant singularity of $F(t)$.

Proof: The variance of the number of prime components is given by:

$$\sigma_n^2 = \frac{\sum_{k=0}^{n} k^2 f_{n,k}}{\sum_{k=0}^{n} f_{n,k}} - \mu_n^2.$$

We can proceed as in Theorem 3.1 and use Proposition 9.1 in [5, p.17] to find the formula in the statement of the Theorem. □

The two formulas for μ_n and σ_n^2 allow us to state that the $(r, s)-$Fibonacci tilings of length n have a number of prime components which is concentrated around the mean. This sentence is supported by the *concentration property* defined in [5]:

DEFINITION 3.4 *A sequence of random variables X_n with mean μ_n satisfies the concentration property if, for any $\epsilon > 0$, one has*

$$\lim_{n\to\infty} Pr\left\{1-\epsilon < \frac{X_n}{\mu_n} < 1+\epsilon\right\} = 1.$$

We can state the following:

COROLLARY 3.5 *The number of prime components in an $(r, s)-$Fibonacci tiling of fixed length n satisfies the concentration property.*

Proof: From Theorems 3.1 and 3.3 we have: $\lim_{n\to\infty} \sigma_n^2/\mu_n = 0$, so we can directly apply Theorem 9.2 in [5, p.8]. □

	ρ	$\frac{1}{\rho P'(\rho)}$
$r = 1,\ s = 2$	0.6190476190	0.7217675941
$r = 1,\ s = 5$	0.7539682540	0.5070440777
$r = 3,\ s = 7$	0.8617834395	0.2277316102

Table 2: Some values of ρ and $C = \frac{1}{\rho P'(\rho)}$ for different r and s.

In Table 2 we show the values obtained for ρ and $C = \frac{1}{\rho P'(\rho)}$ by using formulas in Theorem 2.3 and Corollary 3.2 for different r and s.

We are now ready to introduce the following probabilities which will allow us to define Algorithm 1:

- p_k^n is the probability that an $(r,s)-$Fibonacci tiling of length n has k prime components:
$$p_k^n = \frac{f_{n,k}}{\sum_{k=0}^{n} f_{n,k}} = \frac{f_{n,k}}{F_n^{[r,s]}}. \tag{9}$$
- π_k^n is the *cumulate probability*, that is, the probability that a tiling of length n has a number of prime components less or equal to k:
$$\pi_k^n = \sum_{j=0}^{k} p_j^n = \sum_{j=0}^{k} \frac{f_{n,j}}{F_n^{[r,s]}} = \frac{1}{F_n^{[r,s]}} \sum_{j=0}^{k} \binom{j}{\frac{n-rj}{s-r}}. \tag{10}$$

Algorithm 1: Guidelines for the random generation of $(r,s)-$Fibonacci tilings

Input: an integer number *n*.
Output: a random tiling of length *n*.

1 Generate a uniformly distributed variable α in the interval $(0,1)$
2 Generate a value k with probability p_k^n. The value k corresponds to the least integer number such that $\alpha \leq \pi_k^n$.
3 Generate a binary sequence, *T*, of length k, having $C_0^{[r,s]}$ bits 0 and $C_1^{[r,s]}$ bits 1, with $C_0^{[r,s]}$ and $C_1^{[r,s]}$ given by:
$$\begin{cases} C_0^{[r,s]} + C_1^{[r,s]} & = & k \\ rC_0^{[r,s]} + sC_1^{[r,s]} & = & n \end{cases}$$
4 **return** a shuffle of *T*.

The output of this algorithm is a random bit string of length k where every bit represents a step in a path from the root to level k in the generating tree associated to the (r,s)-Fibonacci problem or, equivalently, one of the two prime components associated to this problem. Since we fix the length n of the tiling, we can determine exactly the numbers $C_0^{[r,s]}$ and $C_1^{[r,s]}$ of each prime component, in fact they correspond to the solution of the linear system of line 3 in Algorithm 1. If we suppose we are able to generate the value k with the proper probability and in constant time, we can surely state that the previous algorithm is linear in k. So, the only problem is represented by the computation of the probability function p_k^n.

We cannot hope to perform the generation in constant time if we have to compute the exact values of the binomial coefficients. Unfortunately the binomial coefficients cannot be approximated in an easy way and for this reason we decided to use the *normal distribution* to approximate our distribution (see, e.g., [4]). A normal distribution in a variate X with mean μ and variance σ^2 has the following *probability density function*:

$$P(x) = Pr\{X = x\} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)} \tag{11}$$

which represents how likely each value of the random variable is. By substituting $\mu = 0$ and $\sigma^2 = 1$ in (11) we obtain the so-called *standard normal distribution* with formula:

$$\hat{P}(z) = \frac{1}{\sqrt{2\pi}} e^{\frac{-z^2}{2}}. \tag{12}$$

Vice versa, $P(x)$ can be obtained from $\hat{P}(z)$ by applying the change of variable $z = (x - \mu)/\sigma$, or, equivalently,

$$x = z\sigma + \mu. \tag{13}$$

The so-called *cumulative distribution function* $\Phi(x)$ gives the probability that a variate will assume a value $\leq x$ and is given by:

$$\Phi(x) = Pr\{X \leq x\} \equiv \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{\left(\frac{-(t-\mu)^2}{2\sigma^2}\right)} dt,$$

while the *standard normal cumulative distribution function* is:

$$\hat{\Phi}(z) = Pr\{X \leq z\} \equiv \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{\left(\frac{-t^2}{2}\right)} dt.$$

The function $\Phi(x)$ will be used to approximate our discrete distribution function $\pi_n(x) = \sum_{k \leq x} p_k^n = \pi_x^n$. More precisely, in our algorithm we will use $\hat{\Phi}(z)$ and then the relation

$$\Phi(x) = \hat{\Phi}(\frac{x - \mu}{\sigma})$$

to obtain $x = z\sigma + \mu$. More precisely, once $\alpha \in (0, 1)$ has been uniformly generated, we need to invert the equation $\hat{\Phi}(z) = \alpha$. The standard normal cumulated distribution function cannot be exactly inverted but there exist rational approximations of it which can be used to this purpose. The following approximation can be found in [1, p. 932]:

$$\hat{\Phi}(z) \approx 1 - \frac{1}{2(C_0 + C_1 z + C_2 z^2 + C_3 z^3 + C_4 z^4 + C_5 z^5 + C_6 z^6)^{16}} \tag{14}$$

Algorithm 2: Random generation of $(r,s)-$Fibonacci tilings

Input: an integer number *n*.
Output: a random tiling of length *n*.

Generate a random number $\alpha \in (0,1)$;
if $\alpha < 1/2$ **then**
    $\alpha := 1-\alpha$;
    little := **true**;
else
    little := **false**;
Solve: $\hat{\Phi}(z)=\alpha$, with respect to the variable z;
$\backslash *$ $\hat{\Phi}(z)$ *is the function defined in (14)* $* \backslash$
Assign the previous result to variable *temp*;
if (*little* = **true**) **then**
    temp := -*temp*;
Compute μ_n and σ_n^2;
$\backslash *$ *By using formulas in Theorems (3.1) and (3.3)* $* \backslash$
$k := \lfloor temp \cdot \sigma_n^2 + \mu_n + (s-r)/2 \rfloor$;
$\backslash *$ *Change of variable, as described in (13)* $* \backslash$
j:=1;
if $(s-r) \neq 1$ **then**
    while $((r * j) \ mod \ (s-r)) \neq 1$ **do**
        j:=j+1;
$p := (n * j) \ mod \ (s-r)$;
$k := \lfloor (k-p)/(s-r)) * (s-r) + p \rfloor$
$\backslash *$ *Shift to take the smallest value of k* $* \backslash$
$C_1^{[r,s]} := \frac{n-rk}{s-r}; \quad C_0^{[r,s]} := k - C_1^{[r,s]}$;
$\backslash *$ *See the discussion related to the system of line 3 in Algorithm (1)* $* \backslash$
Generate a bitstring T with length k, composed by $C_0^{[r,s]}$ bits 0 and $C_1^{[r,s]}$ bits 1;
return a shuffle of T;

where

$$C_0 = 1,\ C_1 = 0.0498673470,\ C_2 := 0.0211410061,\ C_3 = 0.0032776263$$

$$C_4 = 0.0000380036,\ C_5 = 0.0000488906,\ C_6 = 0.0000053830.$$

The solution of $\hat{\Phi}(z) = \alpha$ can be now transformed in $x = z\sigma + \mu$ as desired.

Algorithm 2 represents the pseudo-code for the uniform random generation of $(r,s)-$Fibonacci tilings. Let us now explain some of the instructions of the algorithm which we implemented in *Maple 8*. The random generation in line 1 is obtained by using the function `rand()` and then normalizing the value so computed. The shift of variable α, required in line 2, is mandatory, because (14) provides a function assuming values in the interval $(1/2, 1)$. By the way, $\hat{\Phi}(z)$ is an odd function, symmetric with respect to the value $\hat{\Phi}(z) = 1/2$; for this reason, if in line 1 we obtain a value less than $1/2$, we have to consider the value $1 - \alpha$, invert the function with that number, and than (see line 9) consider the proper value of z (that is $-z$). In line 7 we used the function `fsolve()` to invert the function $\hat{\Phi}(z)$. Line 11 requires the computation of mean and variance; we used formulas (3.1) and (3.3) to obtain good approximations for these two quantities in constant time. The formula computed in line 12 is required to come back to $\Phi(x)$; it represents the change of variable explained before (see formula (13)), plus a shifting necessary to center the intervals. Some care is required to explain the operations performed in lines 13-17. Depending on the values of r and s, our distribution can have very sparse values; in this case, every line of matrix F is constituted by few elements different from 0, spaced by a certain number of zeroes. More precisely, apart a number of zeroes at the beginning of each row, we have an element different from zero every $s - r$ elements. This observation implies that we have the same cumulative distribution for different values of k. The transformation in lines 13-17 allows us to determine the smallest among these values of k. In line 18 we compute the exact number of prime components of the two kinds. These numbers are given by the solution of the linear system shown in line 3 of Algorithm 1. Line 19 requires a generation of a bit string with the proper number of zeroes and ones and in line 20 the previous string is shuffled, to obtain the random tiling as required. We emphasize that every instruction of lines 1-19 is performed in constant time. The only instruction which requires a linear time is the shuffling of line 20. This linearity depends on the parameter k, representing the number of prime components of the $(r,s)-$Fibonacci problem, and whose mean value has been studied in Section 2 where we also proved that for this quantity the concentration property holds true (see Theorem 3.1 and Corollary 3.5).

In Table 3 we summarize the results obtained with our algorithm. We fixed the values of r, s and n and computed the χ^2-test for 20 different experiments: in each experiment we made $N = 2500$ random generations and grouped in Y_k the number of generations corresponding to the same value k. We then computed the statistic:

$$\chi^2 = \sum_{k:\ Np_k \geq 5} \frac{(Y_k - Np_k)^2}{Np_k}.$$

	ν	Theoretical χ^2	Experimental χ^2
$r = 1,\ s = 2,\ n = 100$	16	$5.14 - 39.25$	$10.16 - 29.24$
$r = 1,\ s = 2,\ n = 300$	27	$11.81 - 55.47$	$16.48 - 38.60$
$r = 1,\ s = 5,\ n = 500$	19	$6.84 - 43.82$	$15.34 - 39.11$
$r = 1,\ s = 5,\ n = 700$	22	$8.64 - 48.27$	$19.09 - 43.62$
$r = 3,\ s = 7,\ n = 700$	7	$0.98 - 24.32$	$4.43 - 19.76$
$r = 3,\ s = 7,\ n = 1000$	9	$1.73 - 27.88$	$4.87 - 21.16$

Table 3: In each row, the results of 20 experiments consisting of 2500 random generations.

In Table 3 we report the minimal and maximal values of χ^2 obtained in the 20 experiments. The degrees of freedom ν correspond to the number of k such that $Np_k \geq 5$, minus 1. The results, compared to the theoretical values, are very good and make evident the uniformity of our random generation.

References

[1] M. Abramowitz and I. Stegun. *Handbook of mathematical functions*. Dover Publications Inc., 1964.

[2] J. H. Ahrens. Paving the chessboard. *Journal of Combinatorial Theory, Series A*, 31:277–288, 1981.

[3] C. Banderier, M. Bousquet-Mélou, A. Denise, P. Flajolet, D. Gardy, and D. Gouyou-Beauchamps. Generating functions of generating trees. *Discrete Mathematics*, 246:29–55, 2002.

[4] W. Feller. *An introduction to probability theory and its applications*. Wiley, 1950.

[5] Ph. Flajolet and R. Sedgewick. The average case analysis of algorithms: counting and generating functions. Technical Report 1888, INRIA, 1993.

[6] S. W. Golomb. Tiling rectangles with polyominoes. *Math. Intelligencer*, 18:38–47, 1996.

[7] P. W. Kasteleyn. The statistics of dimers on a lattice. *Physica*, 27:1209–1225, 1961.

[8] D. E. Knuth. *The art of computer programming. Vol. 2*. Addison-Wesley, 1973.

[9] W. R. Marshall. Packing rectangles with congruent polyominoes. *Journal of Combinatorial Theory, series A*, 77:181–192, 1997.

[10] D. Merlini, R. Sprugnoli, and M. C. Verri. Strip tiling and regular grammar. *Theoretical Computer Science*, 242,1-2:109–124, 2000.

[11] D. Merlini and M. C. Verri. Generating trees and proper Riordan Arrays. *Discrete Mathematics*, 218:167–183, 2000.

[12] M. P. Schützenberger. Context-free language and pushdown automata. *Information and Control*, 6:246–264, 1963.

[13] R. Sedgewick and P. Flajolet. *An introduction to the analysis of algorithms*. Addison-Wesley, 1996.

[14] N. Sloane. On-line Encyclopedia of Integer Sequences. http://www.research. att.com/∼njas/sequences.

[15] H. N. V. Temperley and M. E. Fisher. Dimer problem in statistical mechanics, an exact result. *Philosophical Magazine*, 6:1061–1063, 1961.