Strip Tiling and Regular Grammars

Donatella Merlini, Renzo Sprugnoli, M. Cecilia Verri
Dipartimento di Sistemi e Informatica
via Lombroso 6/17, 50134
Firenze, Italia

Abstract

We study the problem of tiling a rectangular $p \times n$-strip ($p \in \mathbf{N}$ fixed, $n \in \mathbf{N}$) with pieces, i.e., sets of simply connected cells. Some well-known examples are strip tilings with dimers (dominoes) and/or monomers. We prove, in a constructive way, that every tiling problem is equivalent to a regular grammar, that is, the set of possible tilings constitutes a regular language. We propose a straight-forward algorithm to transform the tiling problem into its corresponding grammar. By means of some standard methods, we are then able to obtain some counting generating functions that are rational. We go on to give some examples of our method and indicate some of its applications to a number of problems treated in current literature.

1 Introduction

A *dimer* (or *domino*) is a rectangle having the dimensions of 1×2 units. A typical problem (see e.g. Graham, Knuth and Patashnik [4]) consists in determining the number of ways a strip having width 2 and length n units (called a $2 \times n$ strip, for short) can be filled (or *tiled*) with dimers. The solution is F_{n+1}, where F_n is the n^{th} Fibonacci number. We call a *monomer* a single-unit square and the problem can become more complex if we want to find out the number of ways a $2 \times n$ strip can be tiled with dimers and monomers. As a matter of fact, a straight-forward generalization is obtained by considering a $p \times n$ strip ($p \in \mathbf{N}$ fixed, $n \in \mathbf{N}$) to be tiled with dimers and/or monomers.

In general, a *strip tiling problem* consists of counting the number of ways a $p \times n$ strip ($p \in \mathbf{N}$ fixed, $n \in \mathbf{N}$) can be tiled with some sort of *pieces*, i.e., sets of simply connected *cells* (squares of one unit length sides). Monomers and dimers are the simplest examples of pieces, and dimers have been used as a simple model for gas molecules when they assume two different directions (horizontal and vertical) with respect to the direction of the tube (modelled as a strip) containing the gas. See [6, 8, 13] for this physical model, and [1, 5, 7, 10, 12] for some other combinatorial examples.

Monomers and dimers are by no means the only pieces studied in the literature. For instance, in a recent study, Woan, Rogers and Shapiro [14] consider the pieces obtained by *path pairs* (PP) of length m: they are pairs of paths such that both paths start at the origin of $\mathbf{Z}^2$, consist of m unit horizontal and vertical steps and meet again for the first time after

m steps (they are also called *parallelogram polyominoes*). If C_m denotes the m^{th} Catalan number, there are exactly C_{m-1} pieces, when m is fixed. An unsolved problem is whether or not the C_{m-1} PP's of length m can be used to tile a $2^{m-2} \times 2^{m-2}$ checkerboard. According to the authors, the case $m = 5$ makes an amusing puzzle. Questions of this kind can obviously be stated as a set of strip tiling problems when we drop the condition on the length and only keep the width condition. In this paper, we show how the problem can be solved for every single m, at least theoretically.

Another set of strip tiling problems is treated by W. R. Marshall in [9]. The problems were first posed by Golomb and we refer the reader to [3] for this. As far as [9] is concerned, these problems seem to require a "brute force" approach, because no systematic method for determining a possible tiling is known. However, in the present paper, we wish to show that any single strip tiling problem can be approached in a systematic way; we give an algorithm that enables us to find whether or not the problem has a solution and, if so, how many solutions it has. More specifically, we prove the following basic results:

1. Every strip tiling problem is equivalent to a regular grammar i.e., the set of tilings is a regular language (this is folklore, but it is difficult to give a precise reference; to our knowledge, in the literature, no actual solution of tiling problems has ever directly used this sort of approach).

2. An algorithm exists that finds the regular grammar corresponding to a strip tiling problem (this is completely new, as far as we know).

3. Consequently, we can find the rational function $T(t) = \sum_n T_n t^n$ counting the number of ways a strip of length n can be tiled with the given pieces.

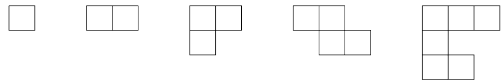
4. It follows that we can find out if there is at least one solution ($[t^{n_0}]T(t) \neq 0$) for any value n_0 of n; we can also determine the number of eventual solutions.

As often happens, a general constructive solution to a problem gives a standard way to approach any particular case but may lack the efficiency of an ad-hoc solution. When the number of pieces and/or the width of the strip is high (say, $p \geq 7$) the regular grammar becomes very large and, consequently, the denominator degree of $T(t)$ is almost intractable. Therefore, what remains to be shown is whether, for a single tiling problem or a class of problems, particular efficient solutions exist. However, due to the exponential growth of tiling possibilities when the width of the strip and/or the number of pieces increase, it is very likely that no "intrinsecally" better solution exists.

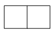
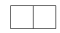
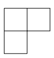
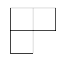
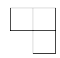
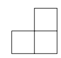
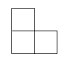
The structure of the paper is as follows. In Section 2 we describe the *strip tiling problem* and prove that it can be solved by a finite state automaton M (Theorem 2.6); in Section 3 we give some examples of strip tiling problems and obtain some regular grammars which we can apply Schützenberger methodology to; in Section 4 we illustrate some other applications by taking into considerations some non-trivial cases: in particular, Theorems 4.1 and 4.2 give the solution of some problems which Marshall [9] leaves to intuition.

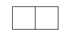
2 Strip tiling problems

We wish to start out with some elementary concepts as a basis for our definition of a "*strip tiling problem*". Our basic unit is a *cell* which can be represented as a square  . A *piece* is a set of simply connected cells, i.e., cells having at least one pairwise common side and no holes:

 .

A piece can have one, two or four different directions; an *oriented piece* is a piece having a definite direction:

-  has a single direction;
-  corresponds to two oriented pieces  ,  ;
-  corresponds to four oriented pieces  ,  ,  ,  .

The *length* and *height* of an oriented piece correspond to the number of its columns and rows (for example, the two oriented pieces which correspond to  have length 2 and height 1 and length 1 and height 2, respectively).

A *p-strip tiling problem* is a rectangular strip having a size of $p \times n$ ($p \in \mathbf{N}$ a fixed parameter, $n \in \mathbf{N}$) and a finite set of (oriented) pieces. When referring to a piece, we always have to state whether we take it as it is or refer to it with some or all of its directions. Below, we only take oriented pieces into consideration and, as a result, we always consider a horizontal and vertical dimer as two different objects.

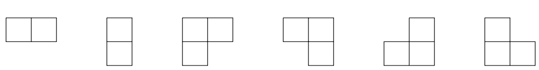
A p-strip tiling problem is solved when we find out the number of ways the strip can be filled up by the pieces. We denote this number by $T_n^{[p]}$; our main result is showing how the generating function $T^{[p]}(t) = \sum_{n=0}^{\infty} T_n^{[p]} t^n$ can be computed. The first basic step consists in proving that all the possible tilings of a $p \times n$ strip make up a regular language; we need some definitions and notations to do this.

Let $\mathcal{P} = \{P_1, P_2, \cdots, P_s\}$ be the oriented pieces of a given p-strip tiling problem, and let r be the maximum length of the pieces:

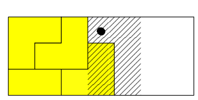
$$r = \max\{\text{length}(P_i) \mid P_i \in \mathcal{P}\}.$$

Definition 2.1 (States) *A state is a $p \times r$ strip whose cells can be either occupied or free (in our examples, a free cell is white and an occupied cell is grey).*

In order to give an intuitive idea of what we mean by "state", let us consider a partially filled $3 \times n$ strip in the 3-strip tiling problem defined by the following pieces:

 (2.1)

If we start from the left, the 3×7 partially filled strip can be the following:

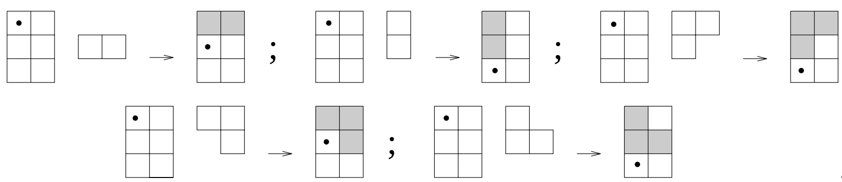


We call the leftmost, highest non-occupied cell (the marked cell in our example) the *pivot cell*. In tiling construction, we can always assume that the new piece is added in such a way that it covers the pivot cell (this position has to be occupied in some way). Therefore, the added piece cannot extend more than r positions to the right and the $p \times r$ substrip containing the pivot cell in its leftmost column is the only part of the strip affected by the insertion of the new piece (the striped part). This is our concept of "state".

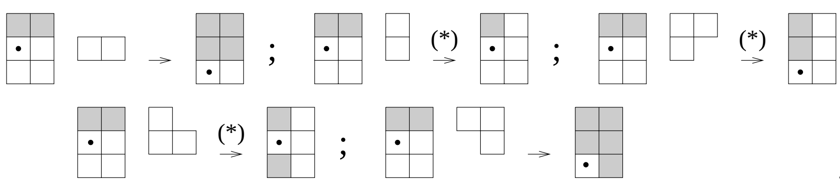
The *initial state* is the state of the strip at the beginning of the tiling process and so it is a $p \times r$ strip containing only free cells. It is worth noting that it is also the "final state", in the sense that it is the state produced when the strip is completely filled up . We denote the initial state by $T^{[p]}$, or simply T. It plays a fundamental role in our development for various reasons. First of all, it allows us to define the important concept of an *admissible state*:

Definition 2.2 (Admissible states) *1) The initial $T^{[p]}$ state is admissible; 2) a state is admissible if it is obtained i) by adding a piece to an admissible state so that it covers the pivot cell; ii) by deleting its completely occupied leftmost columns (if any) and by adding an equal number of free cell columns to its right; 3) there are no other admissible states.*

In our sample problem, the initial state generates five possible admissible states:



We wish to point out that we cannot add the remaining piece to the initial state because it could not cover the pivot cell. In turn, from the first admissible state just obtained, we have:



The transitions denoted by $(*)$ correspond to the application of rule ii) in point 2 above. According to our definition, the last generated state is admissible; however, it is obvious that no piece can be added to it in such a way that the pivot cell is occupied. In a tiling construction, this would stop the process and so this is not a "good" state.

Definition 2.3 (Bad admissible states) *A bad admissible state is an admissible state to which no piece able to cover the pivot cell can be added.*

If an admissible state only produces bad admissible states, it also stops the correct tiling process; we therefore give the following definitions:

Definition 2.4 (Iteratively bad admissible states) *1) A bad admissible state is an iteratively bad admissible state; 2) if an admissible state only produces iteratively bad admissible states when we add some pieces covering the pivot element to it, then it is an iteratively bad admissible state; 3) there are no other iteratively bad admissible states.*

Definition 2.5 (Good admissible states) *A good admissible state is an admissible state which is not an iteratively bad admissible state.*

We take the following observations into account when deriving our results:

a) the number α of admissible states is finite because the total number of possible states is 2^{pr}, i.e., $\alpha \leq 2^{pr}$;

b) the number of possible combinations (state, piece) to be considered during tiling construction is also finite, and is obviously limited by $\alpha\beta$, if β is the number of pieces in a given p-strip tiling problem;

c) therefore, the number of bad and iteratively bad admissible states is also finite; all iteratively bad admissible states can be found by an iterative process starting with bad admissible states; this identification process takes finite time;

d) as a consequence, good admissible states can be determined in finite time.

Unfortunately, as these observations imply, the number of states grows exponentially with p and r, and therefore the complexity of tiling problems increases extremely fast as these parameters increase. We can summarize points a)..d) in the following:

Theorem 2.6 *Let a p-strip tiling problem be defined by the set $\mathcal{P}$ of its pieces; the problem is equivalent to a finite state automaton $M = (Q, \Sigma, q_0, F, \mathcal{T})$ in which:*

- *the set Q of states is the set of good admissible states;*
- *the alphabet Σ is the set $\mathcal{P}$ of pieces;*
- *the initial state q_0 is the initial state $T^{[p]}$ of the tiling problem;*
- *the set of final states F is $\{T^{[p]}\}$;*
- *the set $\mathcal{T}$ of transitions is the set of all possible triples $X\pi \rightarrow Y$, where X, Y are good admissible states and $\pi \in \mathcal{P}$.*

Proof: Let us assume that our p-strip tiling problem has a solution for some $n \in \mathbf{N}$. This, in turn, can be constructed by starting out with the initial state $T^{[p]}$ and by subsequently adding a piece to the previously obtained configuration. The piece can always be attached so that it occupies the pivot cell; in fact, this position always has to be occupied in some way. Consequently, the columns to the left of this position should be filled up and the occupied positions cannot extend more than r positions to the right. These r columns are a state and, actually, a good admissible state because they were filled up during a legal tiling construction. When we add a piece to the tiling, we actually go from one good admissible state to another. The set of all the possible piece attachments is a finite state automaton transition diagram and a tiling is complete when we reach the $T^{[p]}$ good admissible state. On the contrary, if the problem has no solution, $T^{[p]}$ is not a good admissible state. ■

As a simple consequence of the previous theorem, we obtain the following:

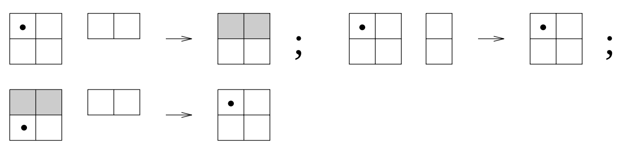
Corollary 2.7 *If a given p-strip tiling problem has a solution, then the set of all its possible solutions is a regular language.*

Proof: We only need to translate the finite state automaton of the previous theorem into the corresponding regular grammar $G = \{N, T, S_0, P\}$ where $N = Q$ is the set of non terminal symbols, $T = \Sigma$ is the set of terminal symbols, $S_0 = q_0$ is the initial symbol and $P = \{Y ::= X\pi : X\pi \to Y \in \mathcal{T}\} \cup \{S_0 \to \epsilon\}$ is the production set. ■

We wish to point out that Corollary 2.7 defines a left regular grammar for a given p-strip tiling problem and that we should obtain an equivalent right regular grammar taking $P = \{X ::= \pi Y : X\pi \to Y \in \mathcal{T}\} \cup \{S_0 \to \epsilon\}$.

3 Examples and further results

As mentioned in the Introduction, one of the simplest tiling problems consists in covering a $2 \times n$ strip by dimers or dominoes. In this case, we only have two pieces: the horizontal and the vertical dimer, and $p = 2$. The state transition diagram can be easily found:



If we denote the initial state by T and the other admissible state by A, use the same letters as the non-terminal symbols in the corresponding regular grammar, and denote the two pieces by h, v, the regular grammar is (BNF notation):

$$T ::= \epsilon \mid Tv \mid Ah$$

$$A ::= Th.$$

We use Schützenberger's methodology [2, 11] to find the counting generating function $T(t)$: we assign the indeterminate t to the two symbols h, v, the value 1 to the empty string ϵ, and the names $T(t), A(t)$ to the functions corresponding to the non-terminal symbols. We thus obtain the simple system:

$$\begin{cases} T(t) = 1 + tT(t) + tA(t) \\ A(t) = tT(t) \end{cases}$$

whose solution is the displaced Fibonacci function:

$$T(t) = \frac{1}{1 - t - t^2} = \frac{1}{t}F(t).$$

If we also assign the indeterminate w to the symbol v we can show that the number of tilings of a $2 \times n$ strip containing exactly k vertical dimers is:

$$T_{n,k} = \binom{(n+k)/2}{(n-k)/2},$$

where the binomial coefficient is to be taken as 0 when $n-k$ is not even, that is, when n is even and k is odd or vice versa.

In general, Schützenberger's methodology can be used for finding many tiling counting properties such as the number of ways a $p \times n$ strip can be tiled or the number of pieces needed to form a tiling when the pieces in a p-strip tiling problem do not have the same area. The following theorem solves both questions:

Theorem 3.1 *If a given p-strip tiling problem has a solution, we can obtain the bivariate generating function* $T^{[p]}(t,w) = \sum_{n,k} T^{[p]}_{n,k} t^n w^k$, *where* $T^{[p]}_{n,k}$ *is the number of tilings of a* $p \times n$ *strip made up of* k *pieces. We can also obtain the generating functions* $T^{[p]}(t) = \sum_n T^{[p]}_n t^n$, *where* $T^{[p]}_n$ *is the number of tilings of a* $p \times n$ *strip and* $\bar{T}^{[p]}(w) = \sum_k \bar{T}^{[p]}_k w^k$, *where* $\bar{T}^{[p]}_k$ *is the number of tilings made up of* k *pieces. Finally,* $T^{[p]}(t)$ *and* $\bar{T}^{[p]}(w)$ *are rational functions.*

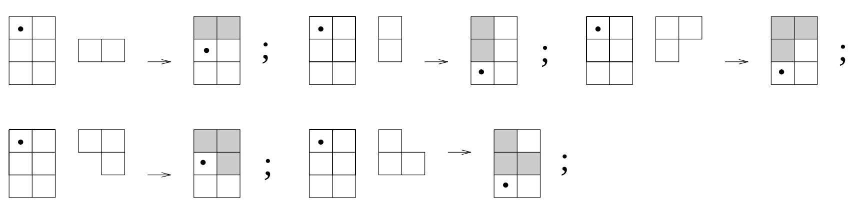
Proof: We use Schützenberger's methodology to associate the indeterminate t to a tiling's length and the indeterminate w to the number of pieces it contains. Therefore, if s is the number of columns added to the tiling, when we attach x to the state Z, every production $W \to Zx$ in the regular grammar becomes a term in the bivariate generating function $W(t,w)$ in the form of $t^s w Z(t,w)$ (the number of pieces is only increased by one). In this way, we obtain a system of equations in the unknown functions having the same name as non-terminal symbols. By solving the system in the unknown $T = T^{[p]}$, we obtain the bivariate generating function desired and have:

$$T^{[p]}(t) = T^{[p]}(t,1) \qquad \text{and} \qquad \bar{T}^{[p]}(w) = T^{[p]}(1,w),$$

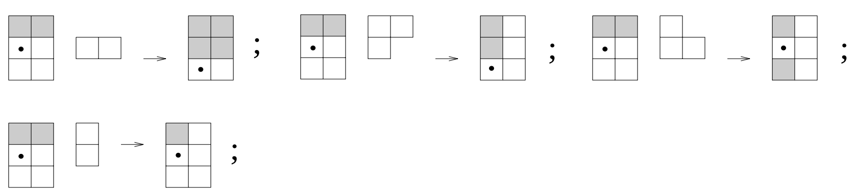
which are rational functions because T is a regular language. ■

At this point, we can go on with the example started in the previous section, with pieces (2.1) and $p = 3$. The complete set of transitions is as follows:

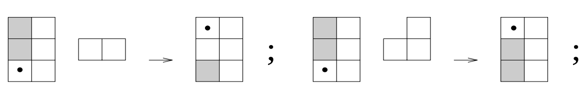
(T)



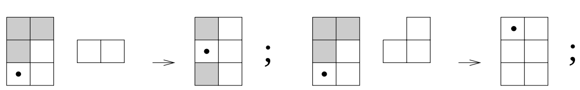
(A)

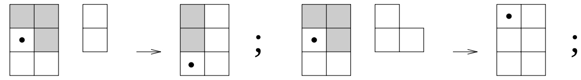


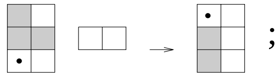
(B)

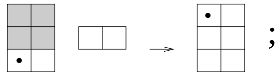


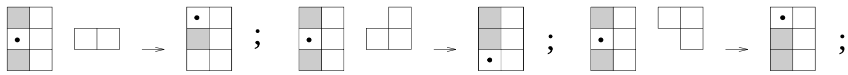
(C)

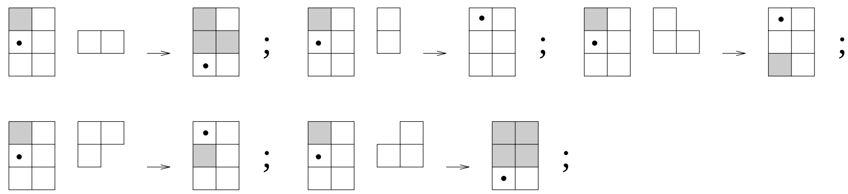


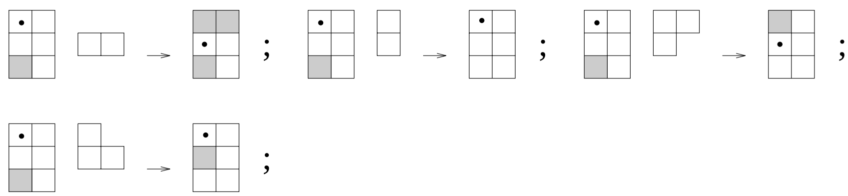
(D) 

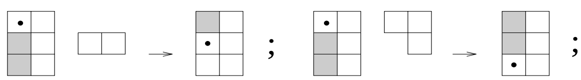
(E) 

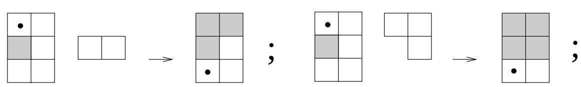
(F) 

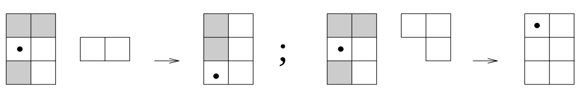
(G) 

(H) 

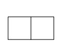
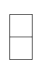
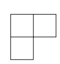
(I) 

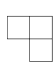
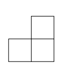
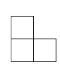
(J) 

(K) 

(L) 

We only reported the good admissible states (13) and the corresponding 35 transitions. By using the letters on the left as non-terminal symbols to translate the (common) left member state, and the following lower case letters for the oriented pieces:

m for , g for , p for ,

q for , d for , b for ,

we obtain the regular grammar:

$T ::= \epsilon \mid Cd \mid Db \mid Fm \mid Hg \mid Ig \mid Lq$

$A ::= Tm$

$B ::= Tg \mid Ap \mid Dg \mid Gd \mid Jq \mid Lm$

$C ::= Tp \mid Km$

$D :: Tq$

$E ::= Tb \mid Hm$

$F ::= Am \mid Hd \mid Kq$

$G ::= Ab \mid Cm$

$H ::= Ag \mid Ip \mid Jm$

$I ::= Bm \mid Hb$

$J ::= Bd \mid Em \mid Gq$

$K ::= Gm \mid Hp \mid Ib$

$L ::= Im.$

For demonstrative purposes, we set up a Maple program that starts out with the definition of a p-strip tiling problem and automatically generates the regular grammar. It then applies Schützenberger's methodology to find out the bivariate generating function as stated in Theorem 3.1.

We can use the strip tiling problem above as a running example; the input phase begins:

> **tiling();**
Enter a piece:$\{1,2\}$;
Do you want to rotate it? (y/n) y;
Another piece? (y/n) y;
Enter a piece: $\{1,2\},\{1\}$;
Do you want to rotate it? (y/n) y;
Another piece? (y/n) n;
Enter the strip's height:3;

Maple outputs the grammar and gives the following generating function:

> **T(t,w);**

$$T(t,w) := -(-2\,t\,w + 2\,t^3\,w^4 + 1 - 2\,t^2\,w^3 + w^2\,t^2 - 3\,w^5\,t^4 + t^4\,w^6) \Big/ (-7\,t^6\,w^8 + t^6\,w^9 + 4\,t^5\,w^6 + 6\,t^6\,w^7 + 14\,w^5\,t^4 + 2\,w^4\,t^4 - 5\,t^4\,w^6 - 2\,t^5\,w^7 - 4\,t^3\,w^3 + 5\,t^2\,w^3 + w^2\,t^2 + 2\,t\,w - 1)$$

which has the following series development:

> **map(factor,series(T(t,w),w,7));**

$$1 + (3w^3 + 2w^2)t^2 + 8w^4t^3 + (40w^5 + 11w^6 + 4w^4)t^4 + (80w^6 + 60w^7)t^5$$

$$+(356w^8 + 228w^7 + 41w^9 + 8w^6)t^6 + O(t^7)$$

Thus, for example, we have 356 different solutions for the strip 3×6, each with 8 pieces. By setting $w = 1$ and $t = 1$ we get the functions $T^{[p]}(t)$ and $\bar{T}^{[p]}(t)$ of Theorem 3.1:

> **Tt:=factor(subs(w=1,T(t,w)));**

$$Tt := \frac{2\,t - 2\,t^3 - 1 + t^2 + 2\,t^4}{2\,t^5 + 11\,t^4 - 4\,t^3 + 6\,t^2 + 2\,t - 1}$$

> **Tw:=factor(subs(t=1,T(t,w));**

$$Tw := -\frac{(\,w^2 - w + 1\,)\,(\,w^4 - 2\,w^3 - w^2 - w + 1\,)}{-7\,w^8 + w^9 - w^6 + 4\,w^7 + 14\,w^5 + 2\,w^4 + w^3 + w^2 + 2\,w - 1}$$

By series development, we obtain:

> **series(Tt,t,10);**

$$1 + 5\,t^2 + 8\,t^3 + 55\,t^4 + 140\,t^5 + 633\,t^6 + 1984\,t^7 + 7827\,t^8 + 26676\,t^9 + \mathrm{O}(\,t^{10}\,)$$

> **series(Tw,w,10);**

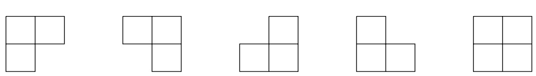
$$1 + 2\,w^2 + 3\,w^3 + 12\,w^4 + 40\,w^5 + 99\,w^6 + 288\,w^7 + 772\,w^8 + 2185\,w^9 + \mathrm{O}(\,w^{10}\,)$$

If we use the Maple program to solve the same tiling problem with $p = 4$ we find a grammar with 36 non terminal symbols and 104 productions; the bivariate generating function is quite complex and has the following series development:

$$1 + w^2t + (5w^4 + 6w^3)t^2 + (11w^6 + 4w^4 + 40w^5)t^3 + (36w^8 + 96w^6 + 248w^7)t^4 +$$

$$+(95w^{10} + 1048w^9 + 1112w^8 + 64w^7)t^5 + O(t^6).$$

Another example is illustrated in Table 1, in which we examine the strip tiling problem with pieces:



(3.2)

for various p's values. We assign the letters a_1, a_2, a_3, a_4, a_5 to the terminal symbols (the pieces) and letters N_i, for increasing i, to the non-terminals.

p	Regular Grammar	Generating Function
2	$N_1 ::= \epsilon \mid N_1a_5 \mid N_2a_3 \mid N_3a_2$ $N_2 ::= N_1a_1$ $N_3 ::= N_1a_4$	$1/(1-t^2\,w-2\,t^3\,w^2) =$ $1+w\,t^2+2\,w^2\,t^3+w^2\,t^4+4\,w^3\,t^5+\mathrm{O}(t^6)$
3	$N_1 ::= \epsilon \mid N_2a_4 \mid N_3a_3$ $N_2 ::= N_1a_1$ $N_3 ::= N_1a_2$	$1/(1-2w^2t^2) =$ $1+2\,w^2\,t^2+4\,w^4\,t^4+8\,w^6\,t^6+\mathrm{O}(t^8)$
4	$N_1 ::= \epsilon \mid N_2a_5 \mid N_{22}a_3 \mid N_{25}a_2$ $N_2 ::= N_1a_5 \mid N_3a_2 \mid N_{17}a_3$ $N_3 ::= N_4a_1 \mid N_6a_4$ $N_4 ::= N_5a_4$ $N_5 ::= N_1a_1$ $N_6 ::= N_7a_3 \mid N_{21}a_2 \mid N_{10}a_5$ $N_7 ::= N_2a_1 \mid N_8a_3 \mid N_{20}a_4$ $N_8 ::= N_9a_1$ $N_9 ::= N_{10}a_4$ $N_{10} ::= N_{11}a_3 \mid N_{14}a_2 \mid N_6a_5$ $N_{11} ::= N_5a_5 \mid N_{12}a_3 \mid N_8a_2$ $N_{12} ::= N_{13}a_1$ $N_{13} ::= N_4a_4 \mid N_{10}a_1$ $N_{14} ::= N_{15}a_5 \mid N_{16}a_3 \mid N_{18}a_2 \mid N_{19}a_4$ $N_{15} ::= N_1a_4$ $N_{16} ::= N_{13}a_4 \mid N_{17}a_1$ $N_{17} ::= N_6a_1$ $N_{18} ::= N_9a_4$ $N_{19} ::= N_{17}a_2$ $N_{20} ::= N_9a_2$ $N_{21} ::= N_2a_4$ $N_{22} ::= N_{23}a_3 \mid N_{24}a_2 \mid N_{13}a_5$ $N_{23} ::= N_5a_1$ $N_{24} ::= N_{15}a_1 \mid N_4a_5 \mid N_{11}a_2 \mid N_{21}a_3$ $N_{25} ::= N_4a_3 \mid N_{26}a_2 \mid N_9a_5$ $N_{26} ::= N_{15}a_4$	$(1-t\,w-6\,t^3\,w^4)/$ $(16\,t^7\,w^9+22\,t^6\,w^8-w^5\,t^4-10\,t^3\,w^4+$ $t^3\,w^3-w^2\,t^2-t\,w+1+3\,t^5\,w^6+4\,t^9\,w^{12}) =$ $1+w^2\,t^2+4\,w^4\,t^3+(5\,w^5+w^4)\,t^4+16w^6\,t^5+\mathrm{O}(t^6)$

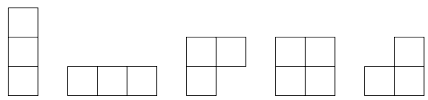
Table 1: The strip tiling problem with the pieces (3.2)

4 Some applications

The Maple program quoted in the previous section allows us to solve a lot of simple tiling problems. Maple is not very efficient as a programming language, at least compared with Pascal or C, and we cannot solve very complicated problems with it. As previously mentioned, the number of possible states grows exponentially with the two parameters p (the strip's width) and r (the longest piece's length). Consequently, whenever one of them is not small, millions of states are generated which fill up the computer's central memory. However, some non-trivial cases can be illustrated to give an idea of our method's possibilities.

Tiling problems involving monomers and dimers have been treated in literature, at least when p is small, and can be used as a general check for the program. We now go on to examine the first non-trivial case treated by Woan, Rogers and Shapiro in [14]: we set

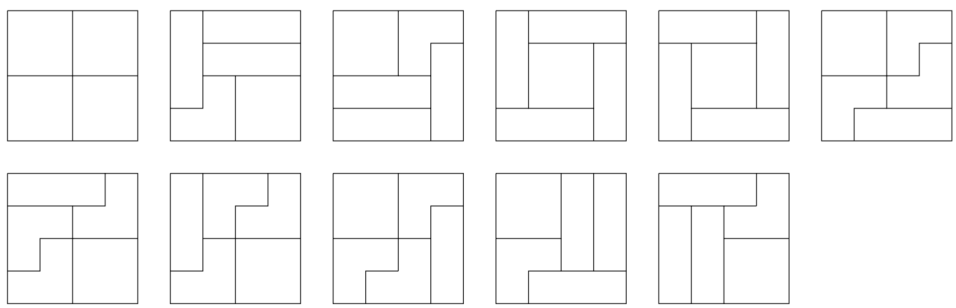
$n = 4$, which corresponds to $C_3 = 5$ and to the oriented pieces:



Since we are looking for the tilings of the 4×4 square, we set $p = 4$ and give the corresponding tiling problem to the computer. Maple produces 104 admissible states and a grammar with 191 productions. By solving the corresponding system, Maple finds the counting generating function, which is very complicated (and not shown in this paper). By setting $w = 1$, we find that the denominator polynomial has a degree of 78 and 0.3162167948 as its smallest module's real solution. Therefore, the number of tilings for a $4 \times n$-strip grows as 3.162387376^n. On the other hand, if we look at the Taylor expansion of $T(t, w)$ around $t = 0$, we find:

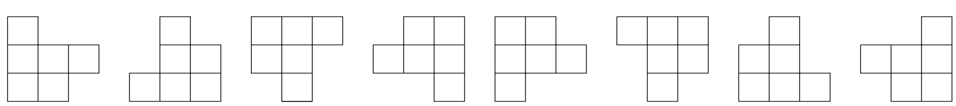
$$T(t, w) = 1 + w^2t^2 + 12w^4t^3 + (w^4 + 10w^5)t^4 + 41w^6t^5 + \cdots$$

This means that the original problem has 11 solutions: one made up of 4 pieces and the other ten made up of 5 pieces, as follows:



It can be noted that no single solution uses all the five pieces of the problem[1]. The next case, $n = 5$, is an amusing puzzle according to Woan, Rogers and Shapiro. In fact, an (admissible) state is a binary array 8×4 and the problem goes beyond the possibilities of our Maple program. However, it could be solved by a more sophisticated program.

As noted in the Introduction, W. R. Marshall's paper [9] is another source of interesting problems. According to his definition, a polyomino or m-omino has order k if k is the minimal number of congruent copies of the polyomino necessary for tiling a rectangle. In our terminology, a congruent copy of a polyomino is an oriented piece obtained by the polyomino or by its mirror image. For example, the hexaomino in Fig. 1 in [9] corresponds to the following eight pieces:



We have the following result which proves the intuition of Marshall:

[1]Someone interprets this problem by considering all the possible oriented pieces. In that case we obtain 36 different solutions.

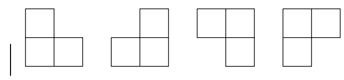
Theorem 4.1 *The order of the Marshall's hexaomino is* 18.

Proof: In order to find the order of a polyomino like this, we try by using ascending values of p until we find at least one solution. By trying $p = 3, 4, 5, 6, 7, 8$, we find that no solution exists; however, for $p = 9$ we find 429 admissible states, 49 good admissible states and a grammar with 52 productions. The counting generating function is:

$$T(t, w) = \frac{1 - w^{12}t^8}{1 - w^{12}t^8 - w^{18}t^{12}} = 1 + 2w^{18}t^{12} + 2w^{30}t^{20} + 4w^{36}t^{24} + \cdots$$

This shows that there are 2 possible tilings of a 9×12-strip, containing 18 pieces. For $p \geq 10$, n, the strip's length, should be greater than 9 because otherwise we would have found the solution in one of the previous cases. On the other hand, any solution with $n \geq 10$ would give an order greater than 18. ∎

The same paper also treats the problem of L-shaped $(2m+1)$-ominoes, and tries to determine the minimal rectangle of an odd size which can be tiled by means of some congruent copies of the polyomino. The simple case $m = 1$, concerns the following pieces:



and we have the following result:

Theorem 4.2 *The minimal rectangle of an odd size is a* 5×9*-strip, tiled with* 15 *pieces. There are* 384 *different solutions to the problem and the construction illustrated by Marshall in Fig. 11 of his paper [9] corresponds to* 128 *of our solutions (*32 *solutions considering the* 4 *possible symmetries).*

Proof: For $p = 3$, we found 4 admissible states, 3 good admissible states and a grammar with 4 productions. We obtained the following generating function:

$$T(t, w) = \frac{1}{1 - 2w^2t^2} = 1 + 2w^2t^2 + 4w^4t^4 + 8w^6t^6 + \cdots$$

which does not correspond to any solution. For $p = 5$, we find 96 admissible states, 67 good admissible states and a grammar with 112 productions. Moreover:

$$T(t, w) = \frac{1 - 2w^5t^3 - 31w^{10}t^6 - 40w^{15}t^9 - 20w^{20}t^{12}}{1 - 2w^5t^3 - 103w^{10}t^5 - 280w^{15}t^9 - 380w^{20}t^{12}} =$$

$$= 1 + 72w^{10}t^6 + 384w^{15}t^9 + 8544w^{20}t^{12} + 76800w^{25}t^{15} + \cdots$$

(the number of solutions grows as 2.3123262128^n). This proves the theorem. ∎

For $p = 7$, we found 29103 admissible states and 4293 good admissible states but could not produce the generating function. According to Marshall, our program should be able to find some $w^{21}t^{15}$ solutions.

There are obviously a great number of analogous examples that could be given.

5 Concluding remarks

We think that our approach can be useful in several ways to anyone interested in tiling problems. First of all because it proposes an automatic way of solving medium-sized problems in a constructive manner. Once the grammar for the problem has been determined, there are some simple programs able to generate all the tilings of a $p \times n$-strip (i.e., all the words of length n belonging to the language). It is also possible to derive procedures that generate a random tiling in a uniform way and in linear time; this is true for any regular language.

Moreover, our method proposes a systematic way of approaching some more difficult problems. The reader is referred to Fig. 12 in Marshall's paper [9] for an example of how constructive arguments can be used for obtaining some negative results.

We wish to point out that a systematic study of our program's complexity could be very useful in deciding whether or not a specific problem can be solved by computer. As far as we know, our method represents the first attempt to furnish a systematic approach to tiling problems.

Finally, we observe that in this paper we considered figures (the strip tiling and the pieces) made up of square cells of unit length sides □. As a matter of fact, our algorithm could be used with other types of cells, for example △ ⬡, and if handled with attention, could also be applied in 3 dimensions, i.e., with 3-dimensional cells.

Acknowledgments

We wish to thank Prof. M. Nivat for his encouragement in dealing with this subject and the anonymous referees whose advice allowed us to improve the contents and readability of our paper.

References

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

[2] M. P. Delest and X. G. Viennot. Algebraic languages and polyominoes. *Theoretical Computer Science*, 34:169–206, 1984.

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

[4] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics*. Addison-Wesley, 1989.

[5] J.L. Hock and R. B. McQuistan. A note on the occupational degeneracy for dimers on a saturated two-dimensional lattice space. *Discrete Applied Mathematics*, 8:101–104, 1984.

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

[7] D. Klarner and J. Pollack. Domino tilings of rectangles with fixed width. *Discrete Mathematics*, 32:45–52, 1980.

[8] E. H. Lieb. Exactly soluble models. *Physica*, 73:226–236, 1974.

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

[10] D. G. Rogers and S. Zaks. Combinatorial proofs of recurrence relations for matchings in rectangular grids. Private communication.

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

[12] R. P. Stanley. On dimer coverings of rectangles of fixed width. *Discrete Applied Mathematics*, 12:81–87, 1985.

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

[14] W. J. Woan, D. Rogers, and L. Shapiro. The catalan numbers, the Lebesque integral and 4^{n-2}. *American Mathematical Monthly (to appear)*.