A strip-like tiling algorithm

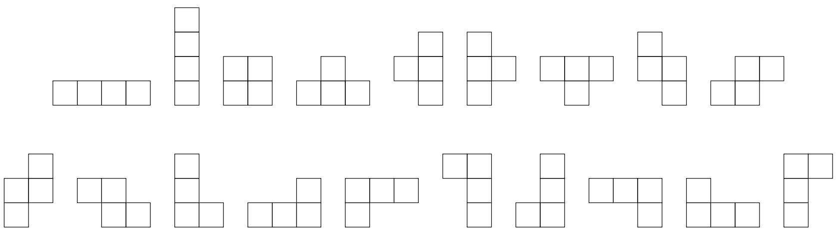
Donatella Merlini, Renzo Sprugnoli, M. Cecilia Verri
Dipartimento di Sistemi e Informatica
via Lombroso 6/17, 50134
Firenze, Italy

Abstract

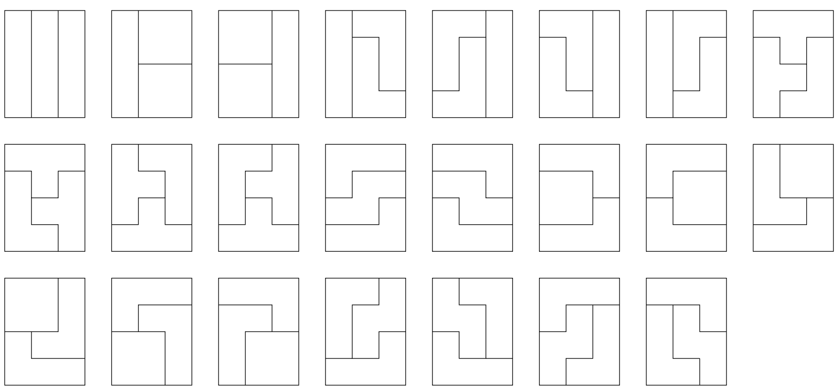
We extend our previous results on the connection between strip tiling problems and regular grammars by showing that an analogous algorithm is applicable to other tiling problems, not necessarily related to rectangular strips. We find generating functions for monomer and dimer tilings of `T-` and `L-` *shaped figures, holed* and *slotted strips, diagonal strips* and combinations of them, and show how analogous results can be obtained by using different pieces.

1 Introduction

An amusing puzzle-problem is to establish in how many different ways a $p \times n$ strip ($p \in \mathbf{N}$ fixed, $n \in \mathbf{N}$) can be tiled with the 19 Tetris pieces:



Typically, we have $p = 10$ but also a smaller value of p leads to non trivial puzzles, due to the large number of solutions as p increases. As a curiosity, here are the 23 solutions corresponding to $p = 4$ and $n = 3$:



In this simple example, proving that these are all the possible solutions is not difficult, but in more complex situations how can we prove that the number of solutions is exactly the one we think? And how can we find the various solutions?

For fixed p, if we denote by $T(t) = \sum_n T_n t^n$, the generating function that counts the number T_n of n-length strip tilings made up of the previous pieces, then we can say that this puzzle-problem is solved if we are able to find an exact or an asymptotic value for T_n and a method for building the tilings. It is obvious that if we denote by k the number of pieces necessary to complete a Tetris puzzle then we must have $pn = 4k$ (we observe that every one of the previous 23 solutions is made up of three pieces) but solving the general problem seems not to be trivial at all.

In [9], the authors study the general puzzle-problem (technically called a *tiling problem*) of counting the number of different 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 unit length sides). The problem is approached by proving the following basic results:

1) every puzzle-problem is equivalent to a regular grammar (i.e., the set of tilings is a regular language);

2) an algorithm exists that can find the regular grammar corresponding to a puzzle-problem;

3) as a consequence of 1) and 2), it is possible to find the rational function $T(t) = \sum_n T_n t^n$, that counts the n-length strip tilings made up of the pieces assigned;

4) it is possible to find out if there is at least one solution ($T_{n_0} = [t^{n_0}]T(t) \neq 0$) for any value n_0 of n; the number of possible solutions can also be determined ($[t^n]T(t)$ denotes the coefficient of t^n in the generating function $T(t)$).

As far as we know, our algorithm is the first attempt to give a systematic approach to tiling problems.

By using the previous algorithm we can solve the Tetris puzzle, at least for small p. For example, for $p = 4$ we find a quite complex rational function $T(t)$ which has the following series development:

$$T(t) = 1 + t + 4t^2 + 23t^3 + 117t^4 + 454t^5 + 2003t^6 + 9157t^7 + 40899t^8 + O(t^9).$$

We observe that $[t^3]T(t) = 23$, as expected. By using the obtained grammar, we can also draw all the different solutions.

In current literature, similar problems have been studied from various points of view: the reader is referred to [7, 11] for a physical model and to [1, 6, 8] for a combinatorial approach; tiling problems have also been used as a tool in several fields to prove complexity results and one of the most definite works on the subject is [3].

In this paper, we extend our results by showing that a simple variation of the algorithm in point 2) can also be used for puzzle-problems not necessarily related to a rectangular strip. We do this by developing several examples in which the figure to tile has some particular shape: we examine the problem of tiling `T`-*shaped,* `L`-*shaped, holed, slotted* and *diagonal* figures (we distinguish between `L`-shaped strips and `L`-shaped figures, see Sections 3.1 and

3.2) with monomers and dimers, i.e., rectangular pieces having a dimension of 1×1 and 1×2. However, we wish to point out again that our method is not limited to these kinds of pieces. We always begin by using the algorithm to obtain the regular grammar which defines the specific problem. We then apply the Schützenberger methodology [5] to get the generating functions related to the puzzle-problems and, in some cases, we go on to find some asymptotic formulas for the coefficients of these functions. We call *strip-like tiling solvable* (SLTS in short) the class of problems we can solve by using the algorithm.

Our aim in developing these examples is to indicate some techniques for solving many other problems related to:

- pieces other than monomers and dimers; examples for the case of rectangular strips are given in [9]; the number and shape of the different pieces increase the problem complexity, but do not change the grammar's construction method;

- figures obtained by joining elementary structures, as rectangular or diagonal strips. The `L`-*shaped* figures example is indicative of how a class of tiling problems, to which we cannot apply the algorithm directly, can be solved by decomposing the original figure in subfigures which are SLTS;

- figures obtained by some regular variations, such as holes, slots, prominences, hollows, etc;

- finite combinations of all these.

The only important point seems to be that growth occurs in only one direction. This, together with a precise definition of the *pivot cell*, allows us to introduce the concept of a *state*, and it is just this concept that relates tiling problems to regular grammars. In fact, this connection has some important by-products, which are worth mention:

- simple programs exist to generate all the tilings of a given length; actually, these programs generate all the words, up to a given length, belonging to the regular language; by using the concept of the pivot cell, translation from words into tilings is immediate;

- it is possible to derive procedures which generate a random tiling in a uniform way and in linear time. Again, a passage from words to tilings is necessary;

- nowadays, software visualization programs exist able to show, in a very choreographical but meaningful way, a random tiling as it is generated. For example, we used such a program, developed by Crescenzi et al. [4], to have a didactic presentation of our algorithm.

The paper is structured into two parts. In Section 2, we explain our algorithm, giving all the necessary definitions. In Section 3, we develop our examples to have fun with our algorithm!

2 The Algorithm

We start out by giving some elementary concepts to formally define a *strip tiling problem*; we then discuss how similar concepts can be applied to a *strip-like tiling problem.*

A *piece* P is a set of simply connected cells, i.e., cells having at least one pairwise common side and no holes; each cell can be represented as a square

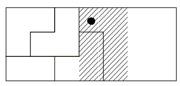
In this paper, we only take *oriented pieces* into consideration and, as a result, we always consider a horizontal and a vertical dimer as two separate objects. The *length* and *height* of an oriented piece correspond to the number of its columns and rows (for example, a horizontal dimer has length 2 and height 1 and a vertical dimer has length 1 and height 2). If a $p \times n$ rectangular strip ($p \in \mathbf{N}$ a fixed parameter, $n \in \mathbf{N}$) and a finite set of pieces are given, a *p-strip tiling problem* consists in finding out the number of ways the strip can be filled up by the pieces. We denote this number by $T_n^{[p]}$ and our algorithm allows us to compute the generating functions $T^{[p]}(t) = \sum_{n=0}^{\infty} T_n^{[p]} t^n$ and other functions that count various tiling distributions. The first basic step consists in proving that all the possible tilings of a $p \times n$ strip make up a regular language.

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 pieces:

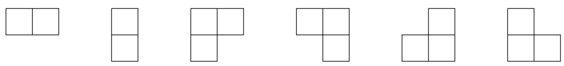
$$r = \max\{\text{length}(P_i) \mid P_i \in \mathcal{P}\}.$$

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 take the striped cells in the partially filled $3 \times n$ strip:



relative to the 3-strip tiling problem defined by the pieces:

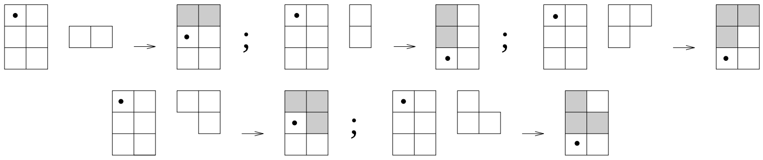


We call the leftmost, highest non-occupied cell (the marked cell in our example) the *pivot cell.* In tiling construction, we can always assume that the new piece is added in such a way that it covers the pivot cell (this position has to be occupied in some way). Therefore, the added piece cannot extend 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".

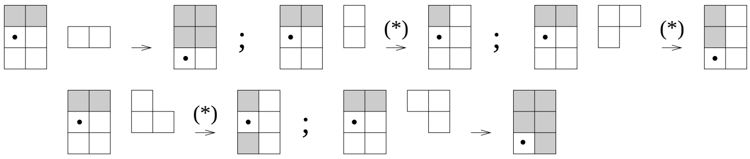
The *initial state* is the state of the strip at the beginning of the tiling process and so it is a $p \times r$ strip containing only free cells. It is worth noting that it is also the "final state", in the sense that it is the state produced when the strip is full. We denote the initial state by $T^{[p]}$, 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 *admissible state* recursively:

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 is unable to cover the pivot cell. From the first admissible state just obtained, we have:



The transitions denoted by $(*)$ correspond to the application of rule ii) in point 2. 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 as to occupy the pivot cell. 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 now wish to point out the following:

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.

We can summarize these concepts in the following (see [9] for the proof):

Theorem 2.6 *Let a p-strip tiling problem be defined by the set $\mathcal{P}$ of its pieces; then the set of all its possible solutions is a regular language defined by the regular grammar $G = \{N, T, S_0, P\}$ where:*

- *the set N of non-terminal symbols is the set of good admissible states;*
- *the set of terminal symbols T is the set $\mathcal{P}$ of pieces;*
- *the initial state S_0 is the initial state $T^{[p]}$ of the tiling problem (if the problem has no solution, $T^{[p]}$ is not a good admissible state);*
- *the set P of productions is the set of all possible triples $X \to \pi Y$, where X, Y are good admissible states and $\pi \in \mathcal{P}$, plus the null production $S_0 \to \epsilon$.*

By using Theorem 2.6 and some standard methods to go from regular grammars on to counting generating functions (e.g., the Schützenberger methodology [10]), we completely solve the p-strip tiling problem.

Let us now consider the strip-like problem of tiling a non-rectangular strip. First of all, we want to clarify what we mean by "non-rectangular strip" and so we describe the properties we want such a figure to have:

- the figure is made up of square cells of unit length sides. As a matter of fact, our algorithm could be used with other types of cells but, for the sake of simplicity, we limit our study to square cells. If handled with attention, the algorithm could also be applied in 3 dimensions, i.e., with 3-dimensional cells.
- the figure grows in only one direction (as in the $p \times n$ rectangular case, in which p is fixed and n changes). This condition is essential if we want to be sure that the number of good admissible states is finite.
- the pivot cell can be defined in an unambiguous way.

As far as the pieces are concerned, we adopt the same definition used for rectangular strips.

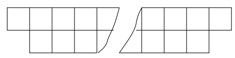
We can now go on and use the above algorithm with the only difference that the process it describes starts out from an "initial" state and ends at one or more "final" states, usually different from the $T^{[p]}$ state previously defined. In particular, we can define the same type of regular grammar described in Theorem 2.1 but which has a different initial state and a null production $F \to \epsilon$ for each final state F. This grammar completely defines the strip-like tiling problem. In the next section, we use several examples to illustrate the algorithm. After the grammar is defined, we immediately go on to find some counting generating functions describing the problem.

3 Beyond Rectangular Strips

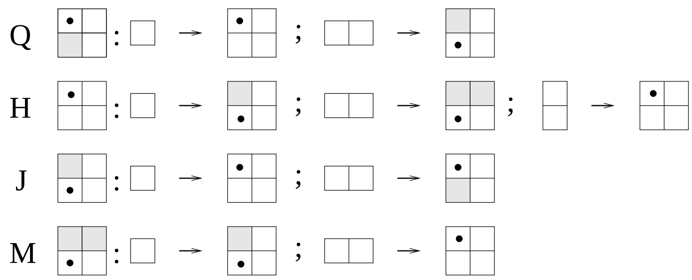
We now take a series of examples into consideration that show that the algorithm in Section 2 can be adapted to non-rectangular strips. For the sake of simplicity, we only examine monomers and dimers, but we would have no problem in treating more complex pieces also.

3.1 T-shaped strips

The simplest example of a tiling problem unrelated to a rectangular strip consists in tiling a T-shaped strip by means of monomers and dimers:



By the algorithm, we obtain the following four good admissible states, with Q as the initial state and J as the final one:



If we use a to denote the monomer, and h and v to denote the horizontal and vertical dimers, we obtain the following production set:

$$Q ::= aH \mid hJ.$$

$$H ::= aJ \mid hM \mid vH$$

$$J ::= aH \mid hQ \mid \epsilon$$

$$M ::= aJ \mid hH$$

By the Schützenberger methodology, we can now find the bivariate generating function $Q(t,w)$ such that $[t^n w^k]Q(t,w)$ represents the number of possible k-length T-shaped strip tilings having n pieces. By length we mean the number of cells in the first row, minus 1. Every piece counts as t, and as w^i, $i = 0,1,2$ because the piece added either does not lengthen the strip or it lengthens it by one or two new cells. For the sake of simplicity, we denote the functions $Q(t,w)$, $H(t,w)$, $J(t,w)$ and $M(t,w)$ by Q, H, J and M. We obtain the following system (in canonical form):

$$\left\{\begin{array}{rrrrcl} Q & -tH & -twJ & & = & 0 \\ & (1-tw)H & -twJ & -tw^2M & = & 0 \\ -twQ & -tH & +J & & = & 1 \\ & -tH & -tJ & +M & = & 0 \end{array}\right. \tag{3.1}$$

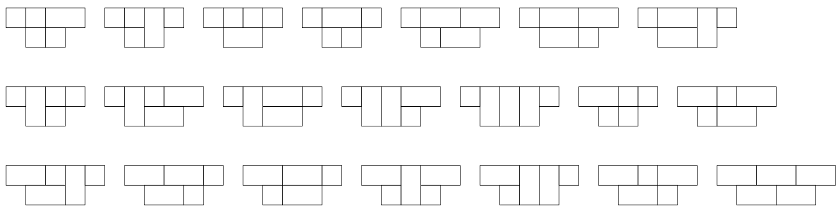
By solving in Q, we find:

$$Q = \frac{tw(1 + t - tw + t^2w - t^2w^2)}{1 - tw - t^2w - 2t^2w^2 - 2t^3w^2 + t^3w^3 - t^4w^3 + t^4w^4}.$$

From this generating function, we can obtain monovariate generating functions that count our tilings by their pieces and length. We set $w = 1$ to find out the number of pieces and obtain:

$$Q(t) = \frac{t}{1 - t - 3t^2 - t^3} = t + t^2 + 4t^3 + 8t^4 + 21t^5 + 49t^6 + 120t^7 + \cdots$$

For example, we have the following 21 tilings containing exactly five pieces: 6 with length 3, 14 with length 4 and 1 with length 5.



As far as the growth rate goes, we have $1 - t - 3t^2 - t^3 = (1 + t)(1 - 2t - t^2)$; therefore, the convergence radius of the series is $\rho = \sqrt{2} - 1 = 0.41421356237$, the root having smallest modulus. Consequently, the number of tilings according to the number of pieces increases as $(1 + \sqrt{2})^n$; in fact, by Bender's theorem [2], we have:

$$Q_n = [t^n]Q(t) \sim \left[\frac{t}{(1 + t)(1 + (\sqrt{2} - 1)t)} \middle| \, t = \sqrt{2} - 1 \right] [t^n] \frac{1}{1 - (\sqrt{2} + 1)t} =$$

$$= \frac{(1 + \sqrt{2})^n}{4}.$$

Of course, we could have found an exact expression for Q_n by a partial fraction decomposition of $Q(t)$. We set $t = 1$ to count the strips by their length:

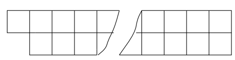
$$\hat{Q}(w) = \frac{2w - w^3}{1 - 2w - 4w^2 + w^4} = 2w + 4w^2 + 15w^3 + 46w^4 + 150w^5 + 480w^6 + \cdots$$

For $\hat{Q}(w)$, we find $\rho = 0.3111078174$ and therefore $\hat{Q}_k$ grows as $\rho^{-k} = (3.214319743)^k$.

We note that if the initial and/or the final state vary, in (3.1), only the right-hand member and the symbol (in respect to which we have to solve the system) change. This implies that we always get a rational function with the same denominator. For example, for a rectangular strip, (which has H as its initial and final state), the right-hand member becomes $(0, 1, 0, 0)$; by solving for H we find:

$$H = \frac{1 - t^2w^2}{1 - tw - t^2w - 2t^2w^2 - 2t^3w^2 + t^3w^3 - t^4w^3 + t^4w^4}.$$

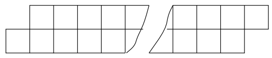
Let us complete our study by examining the strip:



In this case, the initial state is Q and the final state is H; therefore we have the same right-hand member as before but we have to solve for Q. We obtain:

$$\bar{Q} = \frac{t + t^2 w}{1 - tw - t^2 w - 2t^2 w^2 - 2t^3 w^2 + t^3 w^3 - t^4 w^3 + t^4 w^4}.$$

Finally, the strip:



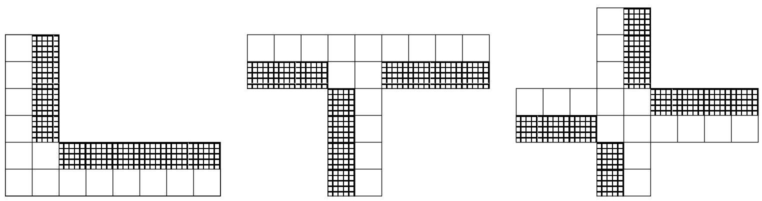
corresponds to the initial state J and to the same final state J; the right-hand member of (3.1) is again $(0, 0, 1, 0)$, but we should solve for J this time:

$$J = \frac{1 - tw - t^2 w^2}{1 - tw - t^2 w - 2t^2 w^2 - 2t^3 w^2 + t^3 w^3 - t^4 w^3 + t^4 w^4}.$$

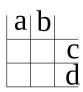
It is worth noting that when the initial state is different from H, the result obtained does not take the occupied cells of the initial state into account. In other words, w is not raised to the appropriate power. Consequently, the generating function should be multiplied by w when the initial state is J or Q, and by w^2 when the initial state is M.

3.2 Strip-like tiling solvable problems

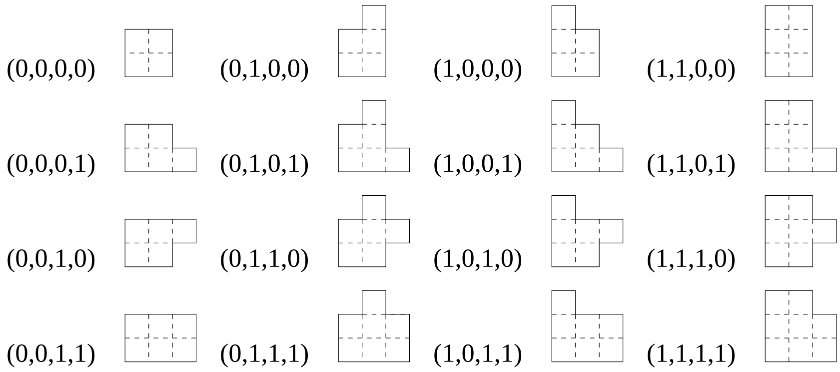
In Section 2 we said that our algorithm can be applied to figures which grow in only one direction. This is not completely true and a class of tiling problems associated to figures which grow in more than one direction can be solved by decomposing the original figure into subfigures which are strip-like tiling solvable. To this class, for example, `L`-*shaped*, `T`-*shaped* and `+`-*shaped* figures belong:



Let us study as a meaningful case, the `L`-shaped figure made up of two rectangular strips joined orthogonally to form an `L` (`T`-shaped and `+` -shaped figures are made up of rectangular strips joined orthogonally to form a `T` or a `+`, respectively). We want to determine the generating function L that counts the tilings in terms of pieces (indeterminate t) and length (indeterminate w). In this case, by *length* we mean the number of inner cells (i.e., the marked part above, which corresponds to a length of 10, in the `L`-shaped figure). The problem can be solved by studying the corner's possible tilings:



This same construction is applicable to the other figures. The complexity grows because the combinations of the central structure are much more than the corner's possible tilings, but the reasoning is just the same. Obviously, we have $2^{2^3} = 256$ configurations for the `T`-shaped figures and $2^{2^4} = 65736$ configurations for the `+` -shaped figures. In the `L`-shaped figure case, the quadruple (a, b, c, d) can be specialized in terms of $0/1$ in the following sixteen configurations:



Each of these configurations corresponds to some tilings and we are interested in those configurations in which the extension outside the corner can be covered by a part of a dimer (otherwise, we obtain a tiling which can also be obtained by a different configuration). Thus, we don't take into consideration the configurations $(0, 1, 1, 0)$, $(0, 1, 1, 1)$, $(1, 1, 1, 0)$

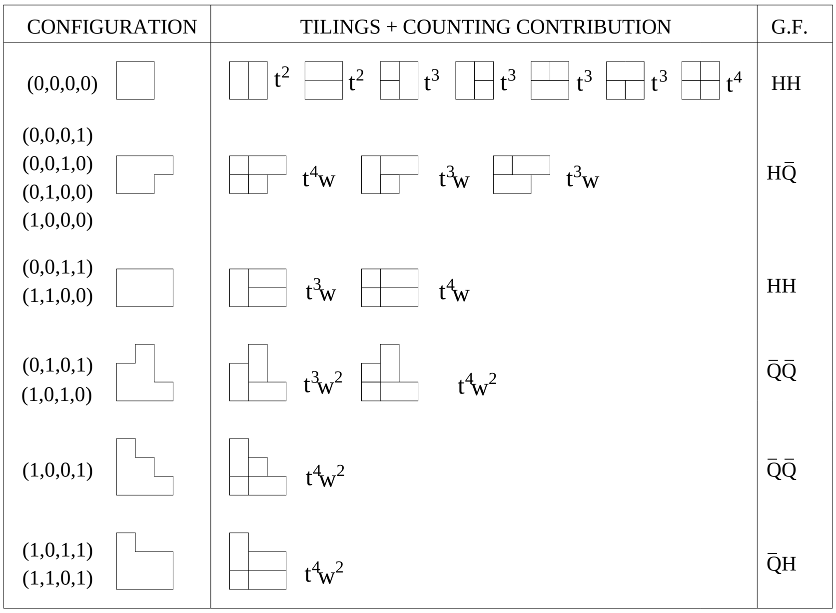


Figure 3.1: The schema for `L`-shaped figures with monomers and dimers as pieces.

and $(1, 1, 1, 1)$ which can be tiled only by using monomers for the extension outside the corner. The other configurations contribute to the generating function desired in terms of number of pieces and length: each monomer and dimer contributes as a t and each extension outside the corner is counted as w : w^0 corresponds to the configuration $(0, 0, 0, 0)$, w^1 corresponds for example to the configurations $(0, 0, 0, 1)$ and $(0, 1, 0, 0)$, w^2 corresponds for example to $(1, 1, 0, 1)$, and so on. It is now evident that a complete tiling of our figure can be obtained by joining the tilings counted by the functions H and $\bar{Q}$ (of the previous subsection) to the corner's possible configurations as follows: we can attach a rectangular strip (counted by H) to any configuration $(0,0)$ and $(1,1)$ and any figure counted by $\bar{Q}$ to any configuration $(0,1)$ and $(1,0)$. Since the two parts to be joined are independent of each other, the generating function desired is the product of the two parts' generating functions and the corner's tiling contribution. The generating function $L(t,w)$ is obtained by summing up all

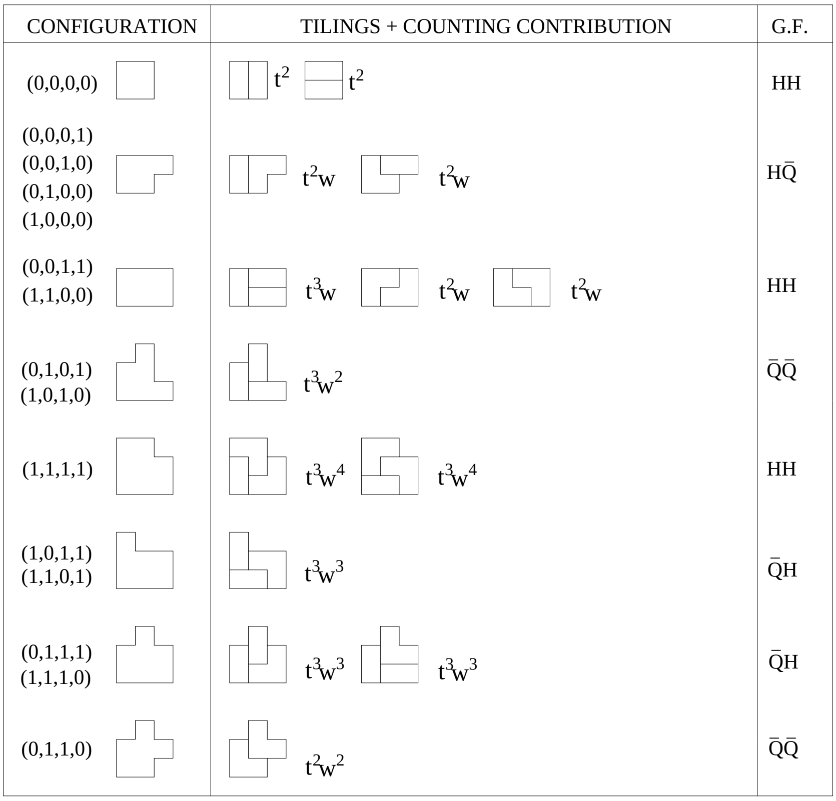


Figure 3.2: The schema for L-shaped figures with pieces (3.2).

the contributions illustrated in Figure 3.1, in which we distinguish the corner configurations, their possible tilings plus the counting contributions and finally the generating functions associated to the two parts to be joined to get the complete figure. We obtain:

$$L(t,w) = (2t^2 + 4t^3 + t^4 + 2t^3w + 2t^4w)H(t,w)^2 + (8t^3w + 4t^4w +$$

$$+2t^4w^2)H(t,w)\bar{Q}(t,w) + (2t^3w^2 + 3t^4w^2)\bar{Q}(t,w)^2 =$$

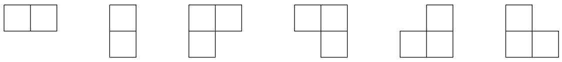
$$= \frac{t^2(2 + 4t - 2wt + 2wt^2 - 2w^2t^2 + t^2 + 2w^3t^3 + 2wt^3 - 4w^2t^3)}{(1 - 2tw - wt^2 - w^2t^3 + w^3t^3)^2},$$

which can be expanded into the following power series:

$$L(t,w) = 2t^2 + (6w + 4)t^3 + (14w^2 + 22w + 1)t^4 + (30w^3 + 76w^2 + 14w)t^5 + O(t^6).$$

This, in turn, indicates that there are two 0-length tilings with 2 pieces, four 0-length tilings with 3 pieces, six 1-length tilings with 3 pieces, and so on.

By proceeding in an analogous way we can obtain counting results for the `T`-shaped and `+` -shaped strips. Obviously, the same method can be applied with different pieces. For example, if we want to tile the `L`-shaped strip with the following pieces:

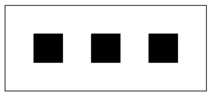


(3.2)

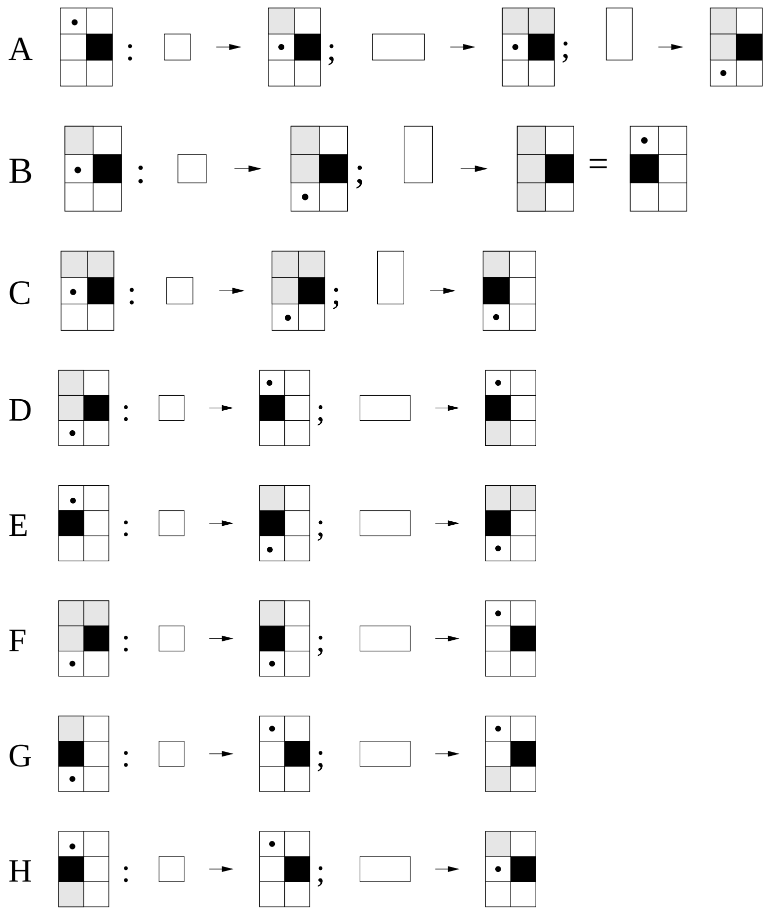
we have to take into consideration Figure 3.2 and sum all the contributions together. Functions H and $\bar{Q}$ are analogous to the functions H and $\bar{Q}$ of the previous subsection but correspond to pieces (3.2). We observe that in this example, we have to exclude only the corner configuration $(1, 0, 0, 1)$ which does not correspond to any good tiling.

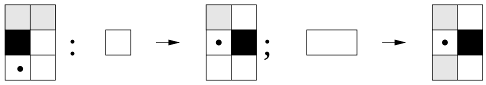
3.3 Holed Strips

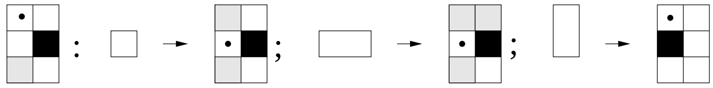
We now study the problem of tiling a $3 \times n$ strip having alternate holes (the black cells) in the second row:

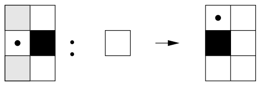


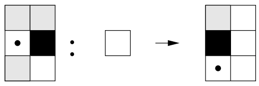
This also is a sample case which can be extended easily to other situations, when the strip is wider and/or the holes have other patterns. In this case, we obtain the following grammar whose initial state is A and final state is E:



I 

J 

K 

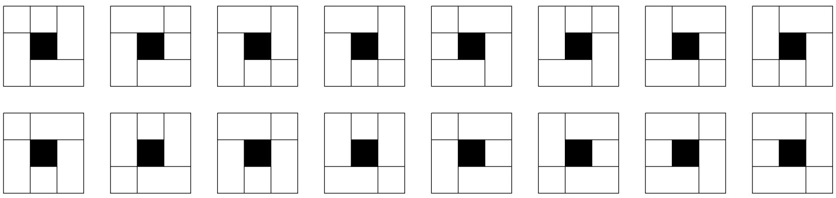
L 

By applying the Schützenberger methodology and solving for A, we get the following generating function:

$$Hol(t,w) = \frac{wt^2(2+t+2w^2t^2+2w^2t^3+w^2t^4)}{1-7w^2t^3-6w^2t^4-w^2t^5+9w^4t^6+6w^4t^7+w^4t^8-w^6t^9} =$$

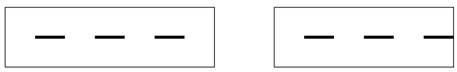
$$= 2wt^2 + wt^3 + 2w^3t^4 + 16w^3t^5 + 20w^3t^6 + (8w^3+14w^5)t^7 + O(t^8).$$

For example, we have the following 16 3-length tilings made up of 5 pieces:



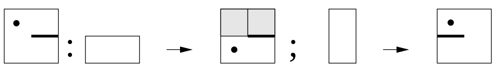
3.4 Slotted Strips

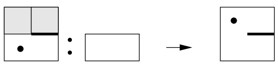
Let us now examine the $2 \times n$ strip having alternate slots:

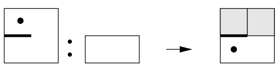


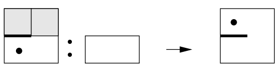
(a) (b)

Situations (a) and (b) correspond to two different final states and we are interested in both. We get the following regular grammar:

A 

B 

C 

D 

in which A is the initial state and A, C the final states. We then obtain the following generating function:

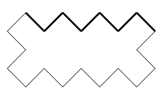
$$Slot(t,w) = \frac{tw}{(1-tw)^2(1+tw)^2} =$$

$$= 1 + wt + w^2t^2 + 2w^3t^3 + w^4t^4 + 3w^5t^5 + w^6t^6 + 4w^7t^7 + w^8t^8 + 5w^9t^9 + O(t^{10}).$$

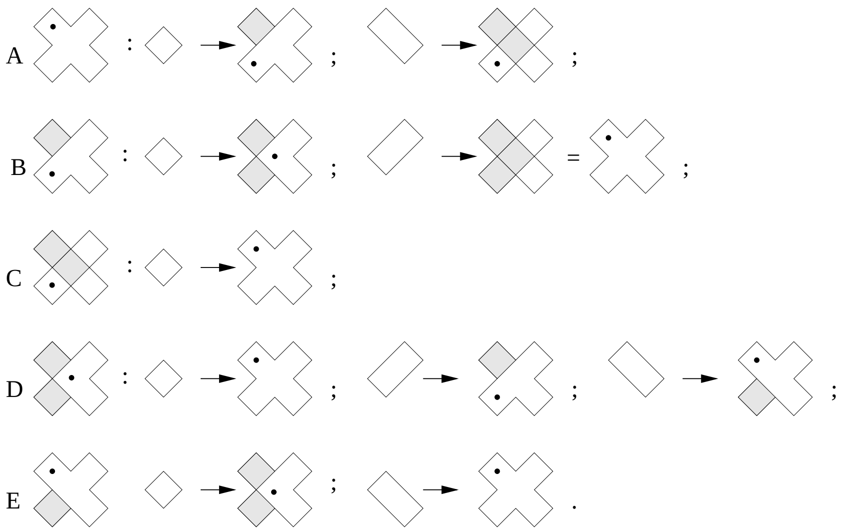
It can be seen that we have a unique solution when the strip length is an even number. This is immediately evident if we build the tiling from right to left.

3.5 Diagonal strips

Our last example can be represented by the following figure shape:



in which the marked line rapresents the length. The initial state is A and the final one is D :



We obtain the following counting generating function:

$$Diag(t,w) = \frac{wt^2}{1-4w^2t^2-w^2t^3+2t^4w^4} =$$

$$= wt^2 + 4w^3t^4 + w^3t^5 + 14w^5t^6 + 8w^5t^7 + (48w^7 + w^5)t^8 + 44w^7t^9 + O(t^{10}).$$

Conclusions

In this paper, we focused our attention in showing how the algorithm introduced in [9] can be applied to figures more complex than a rectangular strip. We showed several examples which we hope are indicative of how many other problems can be solved. As noted in the Introduction, many interesting by-products can be obtained through the connection between tiling problems and regular grammars.

Actually, other research directions are possible. An important point would be to understand how complexity grows as the number of different pieces and/or the width of the strip increase.

Last, but not least, we observe that our tiling algorithm can also be applied to three-dimensional figures, and we have some results for linear strips of dimension $2 \times 2 \times n$ when tiled with three dimensional monomers (i.e., $1 \times 1 \times 1$ pieces) and dimers (i.e., $1 \times 1 \times 2$ pieces, oriented in any direction).

Acknowledgments

We wish to thank the anonymous referees whose comments helped in improving 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] E. A. Bender. Asymptotic methods in enumeration. *SIAM Review*, 16:485–515, 1974.

[3] P. E. Boas. The convenience of tilings. *Complexity, logic and recursion theory, Lecture notes in Pure and Applied Mathematics*, 187:331–363, 1997.

[4] P. Crescenzi and C. Demetrescu and I. Finocchi and R. Petreschi. Reversible Execution and Visualization of Programs with Leonardo. *Journal of Visual Languages and Computing*, 11:125–150, 2000.

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

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

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

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

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