An Almost Totally Universal Tile Set

Grégory Lafitte[1,⋆] and Michael Weiss[2,⋆⋆]

[1] Laboratoire d'Informatique Fondamentale de Marseille (LIF),
CNRS – Aix-Marseille Université,
39, rue Joliot-Curie, F-13453 Marseille Cedex 13, France
[2] Università degli Studi di Milano,
Bicocca Dipartimento di Informatica, Sistemistica e Comunicazione,
336, Viale Sarca, 20126 Milano, Italy

Abstract. Wang tiles are unit size squares with colored edges. In this paper, we approach one aspect of the study of tilings computability: the quest for a universal tile set. Using a complex construction, based on Robinson's classical construction and its different modifications, we build a tile set $\mathfrak{Y}$ (pronounced *ayin*) which *almost always* simulates any tile set. By way of Banach-Mazur games on tilings topological spaces, we prove that the set of $\mathfrak{Y}$-tilings which do not satisfy the universality condition is meager in the set of $\mathfrak{Y}$-tilings.

1 Introduction

Wang was the first to introduce in [Wan61] the study of tilings with colored tiles where a tile is a unit size square with colored edges. Two tiles can be assembled if their common edge has the same color. To tile consists in assembling tiles from a tile set (a finite set of tiles) on the grid $\mathbb{Z}^2$.

Since Berger [Ber66] it is known that Wang tilings can simulate Turing machines. As a model of computation, tilings raise computability questions. One of the first, related to most models of computation, is the existence of universality. To approach such a problem we need a proper notion of reduction. In [LW07], a first approach to reduction, and by extension, universality, was given. Intuitively, a tiling P *simulates* a tiling Q if the tiles of Q can be encoded with macro-tiles of P.

This notion of simulation was then improved in [LW08a] (a close definition is also introduced in [DRS08]) to obtain simulations between tile sets. A tile set τ *totally simulates* a tile set τ' if any τ'-tiling is simulated by a τ-tiling and if any τ-tiling simulates a τ'-tiling. In [LW08a], it has been proved that there exists a tile set that totally simulates any periodic tile set, *i.e.*, tile sets that generate at least one periodic tiling (a tiling invariant by translation of two independent vectors). The question of the existence of a totally universal tile set was asked: does there exists a tile set totally simulating any tile set that tiles the plane? Because

⋆ This author has been supported by the French ANR grant *Sycomore*.
⋆⋆ This author has been supported by the Swiss FNS post-doc grant.

J. Chen and S.B. Cooper (Eds.): TAMC 2009, LNCS 5532, pp. 271–280, 2009.

of the amount of properties that such a tile set would have (having maximal Kolmogorov complexity [DLS01], non-recursivity [Han74, Mye74], self-similarity of any kind [DRS08], Turing universality, invariance by recursive modification [LW08b],...), it has been conjectured that it does not exist.

In this paper, we combine some of the most complex constructions on tilings to build a tile set $\mathfrak{Y}$ which is almost totally universal: almost all $\mathfrak{Y}$-tilings simulate at least one tiling for any tile set (by *almost* we mean that the subset of $\mathfrak{Y}$-tilings that do not satisfy this property is a meager set in the set of $\mathfrak{Y}$-tilings). And therefore, $\mathfrak{Y}$ has the particularity of having *almost always*[1] all the properties enumerated previously. The construction of $\mathfrak{Y}$ uses different technical tools and is mainly based on the aperiodic and self-similar tile set of Robinson within which simulations of Turing machines can be carried out. A detailed explanation of this construction can be found in [Rob71, Han74, Mye74, AD96, DLS01, DLS04]. In [LW08a], and in [DRS08] (ingeniously avoiding the use of Robinson's tilings construction), it has been shown how a Turing machine can be used to simulate a tile set, in the sense that the Turing machine produces space$\times$time diagrams isomorphic to the tiles of a tile set. With this simulation of Turing machines, simulation of tile sets in Robinson's tiling is made possible. The other tool used for the construction is synchronization. This principle was first used by Hanf and Myers [Han74, Mye74] to build a non-recursive tile set, *i.e.*, a tile set that generates only tilings that cannot be defined by a recursive function. We show how to make a synchronization between squares of Robinson's construction in a new way. All these different tools make possible the construction of an almost totally universal tile set.

The last step consists in proving the *almost* part. One of the main tool to prove the meagerness of a set is to use topological games like Banach-Mazur games [Oxt57]. This perfect information game is played by two players on a topological space. A classical result on Banach-Mazur games shows that if Player II has a winning strategy, then A is meager (or, in an equivalent way, the set $X \setminus A$ is residual).

In [LW08c], a topological study of tilings has been made and games *à la* Banach-Mazur on them have been introduced. These games are played on two topological spaces: the Besicovitch one (where the distance between two tilings is defined as the asymptotic proportion of different tiles between them) and the Cantor one (where the distance between two tilings is related to the biggest pattern centered around the origin that they have in common). In this paper we restrict these games to the set of tilings generated by a tile set and we prove, using these games, that the tile set $\mathfrak{Y}$ is almost totally universal.

In the following section, we recall the basic notions concerning tilings and the notions of simulation between tile sets. We also recall the two main topological spaces that can be used on the set of tilings generated by a tile set. Then, in section 3, we show how a synchronization can be made between the different squares in Robinson's construction. In the last section, we build the tile set $\mathfrak{Y}$ and show that it is almost totally universal.

[1] *Almost always* means that almost all of its tilings have the properties.

2 Basic Notions

2.1 Tilings and Simulations

We start by recalling the basic notions of tilings. A tile is an oriented unit size square with colored edges from C, where C is a finite set of colors. A tile set is a finite set of tiles. To tile consists in placing the tiles of a given tile set on the grid $\mathbb{Z}^2$ such that two adjacent tiles share the same color on their common edge. Since a tile set can be described with a finite set of integers, we can enumerate the tile sets, and τ_i designates the i^{th} tile set.

Let τ be a tile set. A tiling P generated by τ is called a τ-tiling. It is associated to a tiling function f_P where $f_P(x, y)$ gives the tile at position (x, y) in P. When we say that we superimpose the tiles of a tile set τ on the tiles of a tile set τ', we mean that for any tile $t \in \tau$ and any tile $t' \in \tau'$, we build a tile $u = t \times t'$ where the colors of the sides of u are the cartesian product of the colors of the sides of t and t'. Then two tiles $u_1 = t_1 \times t'_1$ and $u_2 = t_2 \times t'_2$ match if and only if t_1 and t_2 match and t'_1 and t'_2 match.

Different notions of reductions have been introduced in [LW07]. We recall some of the basic notions here. A pattern is a finite tiling. If it is generated by τ, we call it a τ-pattern. We say that a τ-tiling P *simulates* a τ'-tiling Q if there exist two integers a, b and an application R from the $a \times b$ τ-patterns to the tiles of τ' and if we can cut regularly P in rectangular patterns of size $a \times b$ such that if we replace these rectangular patterns in P by their corresponding tiles given by R we obtain Q. One can see that P does with *macro-tiles*, *i.e.*, rectangular patterns which represent the tiles of another tile set, what Q does with tiles. We denote the reduction by $Q \trianglelefteq^R P$. We generalize this notion to simulations between a set of tilings and a tile set: a set of τ-tilings A *totally simulates* a tile set τ' if there exist $a, b \in \mathbb{Z}$ and a reduction R from the $a \times b$ patterns of τ to the tiles of τ' such that for any τ'-tiling Q, there exists a τ-tiling $P \in A$ such that $Q \trianglelefteq^R P$, and such that for any τ-tiling $P \in A$, there exists a τ'-tiling Q such that $Q \trianglelefteq^R P$. We denote it by $\tau \trianglelefteq A$ (or $\tau' \trianglelefteq^R A$ to specify the reduction R). If A corresponds to the whole set of τ-tilings, then we say that τ simulates τ', and we denote it by $\tau' \trianglelefteq \tau$.

From this pseudometric we obtain a notion of universality: we say that a set of τ-tilings A is *totally universal* if $\tau' \trianglelefteq A$ for any tile set τ' that tiles the plane. If A corresponds to the whole set of τ-tilings, then we say that τ is totally universal. The existence of such a tile set is still open. In this paper we aim at constructing a tile set which is *almost always* totally universal. To have a clear definition of *almost* we recall some notions of topology and topological games. A deeper study of these topological spaces can be found in [LW08c, BDJ08].

The topologies defined in the following subsections are topologies used in cellular automata and adapted to tilings. Different definitions of these topologies can be given and we present here the restrictive case where the distances are defined only between tilings generated by the same tile set.

2.2 The Besicovitch and Cantor Topologies on Tilings

The first metric we introduce is a metric *à la* Besicovitch. This metric deals with the whole tiling and gives the asymptotic proportion of different tiles between two tilings. For two τ-tilings P and Q, we call P_n and Q_n the square patterns of size $2n+1$ centered around the origin. The distance $d_B(P,Q)$ is given by:

$$\delta_B(P,Q) = \text{limsup}_{n\to\infty} \frac{\#\{\,(x,y) \mid f_{P_n}(x,y) \neq f_{Q_n}(x,y)\,\}}{(2n+1)^2}.$$

Therefore, the Besicovitch distance between two τ-tilings corresponds to the assymptotic proportion of different tiles between them. This is a pseudometric on the set of tilings generated by a tile set. We can obtain a metric by adding the condition that two tilings are equivalent if the distance between them is 0. Two tilings not generated by the same tile set are at distance 1. We obtain a topological space by defining the open sets as the balls $\mathcal{B}_B(Q,\epsilon)$, *i.e.*, all tilings at distance at most ϵ of Q.

The second metric, the Cantor one, deals with the local structure of the tilings while the Besicovitch one deals with their global behavior. We first define the function $p : \mathbb{N} \to \mathbb{Z}^2$ such that $p(0) = (0,0)$, $p(1) = (0,1)$, $p(2) = (1,1)$, $p(3) = (1,0)\ldots$ and p keeps having the behavior of a spiral afterward. The metric d_C between two τ-tilings P and Q is defined as $d_C(P,Q) = 2^{-i}$, where i is the smallest integer such that $f_P(p(i)) \neq f_Q(p(i))$, *i.e.*, i is the size of the greatest common pattern of P and Q centered around the origin.

d_C is a metric on the set of τ-tilings. As before, we can obtain naturally a topological space by defining the open sets as the balls $\mathcal{B}_C(Q,\epsilon)$, *i.e.*, all tilings at distance at most ϵ of Q. One can note that in Cantor topology, the set of tilings having in common the same pattern centered around the origin is a clopen set [LW08c, BDJ08].

2.3 Games on Tilings

Now that we have defined notions of topologies on sets of tilings generated by a tile set, the natural next step for studying these sets is to consider infinite games on tilings. In [LW08c], the following definitions of Banach-Mazur games on tilings have been given:

Let X be a set of tilings generated by a tile set and C be a subset of X.

The first game $G(X,C)_B$ is played on Besicovitch topology and has the following rules: Player I chooses a τ-tiling P_1 and an integer n_1. Player II chooses a tiling $P_2 \in \mathcal{B}_B(P_1, 1/n_1)$ and chooses an integer $n_2 > n_1$ and so on. Player II wins the game if $\bigcap_{n>1} \mathcal{B}_B(P_i, 1/n_i) \in C$.

The second game $G(X,C)_C$ is played on Cantor topology and has the following rules: Player I chooses a square pattern A_1 centered around the origin. Player II chooses a square pattern A_2 which is an extension of A_1, i.e., the tiling function of A_2 restricted to the domain of A_1 is the tiling function of A_1, and so on. From the sequence of patterns $\{A_i\}$ we can obtain an infinite tiling P. Player II wins the game if $P \in C$.

The main application of Banach-Mazur games is the study of meager sets. A classical topological result is that a subset C of X is meager, *i.e.*, is the the union of countably many nowhere dense subsets, if and only if Player II has a winning strategy for the game $G(X, X \setminus C)$. *Meagerness* is thus a topological notion of *small* or *negligible* subsets. We obtain the notion of *almost total universality*: a tile set τ is *almost totally universal* if there exists a set of τ-tilings A which is totally universal and such that A is residual in the set of τ-tilings in both Besicovitch and Cantor topologies.

Therefore, an almost totally universal tile set is a tile set totally universal up to a meager set: just a small subset (of its tilings) prevents it to be totally universal. In the following section, we explain some constructions needed to build an almost totally universal tile set.

3 Synchronization within Robinson's Construction

In this section, we show how to synchronize squares of Robinson's tiling (we refer the reader to [AD96] for an explanation of this construction). By synchronization, we mean that any square of any level works on an initial segment of an infinite input. Synchronization was first introduced in [Han74, Mye74] and used in [DLS01]. We propose our own synchronization, adapted for our purpose.

3.1 Synchronization between Squares of Same Level

The first goal to achieve, is to prove that all the squares of a same level have the same information, *i.e.*, any square of a certain level in Robinson's construction have the same input word w on their first line. Since two neighbor squares, either vertical or horizontal, can share the information they have on their facing sides then we need to prove that we can obtain a square which has on its four sides the same input word w. We just need to pass the bits of the input word from the south side to the west side. Then, we can transmit these bits to the north and east sides.

The information going from the south to the west side will pass through three kinds of tiles: it first goes through a tile that transmits the information vertically, then passes a corner and finally goes through tiles that transmit the information horizontally until it reaches the west side. The only condition to add to be sure that all the bits will pass from the south to the west side (like in figure 1) is to force any tile which is not obstructed (obstructions are colored in gray in figure 1) to be one of the three kinds of tiles that transmit information. The obstructed tiles can either transmit vertical or horizontal information, or transmit nothing. Finally, neighbor squares of same level can check if they are computing on the same input word. Therefore all squares of a same level work on the same input word.

3.2 Synchronization between Levels

We now want to synchronize the input word between different levels, *i.e.*, that if w_i is the input word of the squares of level i then w_i is the central word of w_j

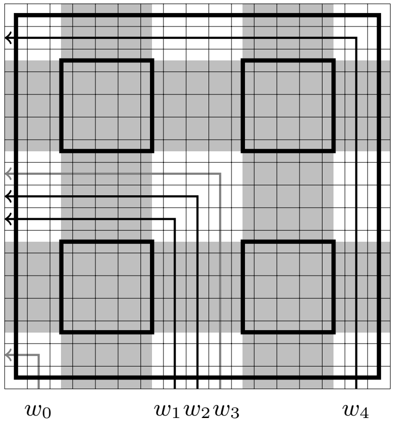


Fig. 1. The transmission of the bits of w from the south side to the west side

for any $j > i$, *i.e.*, there exists two words w_1 and w_2 of same length such that $w_j = w_1 w_i w_2$. In this way all squares of all levels obtain the same computation. We recall that in Robinson's tilings, the squares of level even are colored in black and the squares of level odd are colored in light-gray.

We need to choose a square (the only one) that communicates its input word to the higher level. We give sixteen different labels to the black squares (one of them is labeled in gray) and two kinds of label for the light-gray squares. this in enough to guarantee that any black square of a level n, has a gray square of level $n-1$ in its south-west corner (figure 2.a). This is this gray square who passes the information from its east side to the south side of the square of level n.

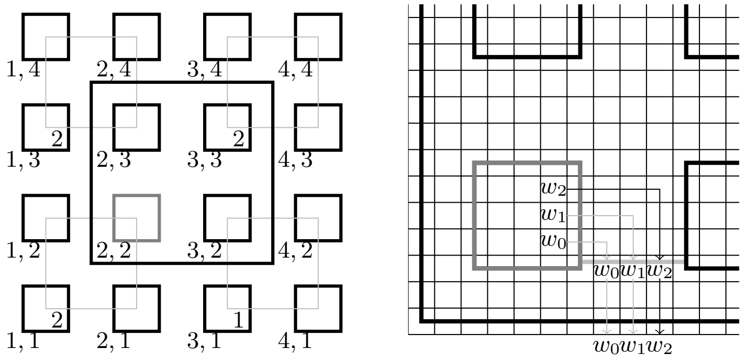
To pass the information, we use the induction process of figure 2.b. The same technique as before is used. We can do this since the number of columns between two neighbor squares is the same as the number of columns in a square. Then, with an induction process, we will pass all the bits from the east side of the gray square to the south side of the black square of higher level.

At the end of the process, the gray row contains the bits of w and the black square of upper level has access to this code and can compute on it. Therefore, any square computes on an initial segment of the same infinite input.

4 An Almost Totally Universal Tile Set

4.1 Description of the Construction

In this section we construct an almost totally universal tile set. We use three Turing machines M, N and P that we simulate in the synchronized construction explained previously. The three machines works on an infinite string $i_1\$i_2\$\ldots$ where i_j is the code of a tile set of j tiles that tiles the plane. M checks if the input is well written. If not, M stops. We add another restriction to M: we want that the code of $\mathfrak{Y}$ appears as an input. The tile set $\mathfrak{Y}$ will have access to its own



(a) The coloring of the black, gray and light-gray squares

(b) The synchronization between squares of different levels

Fig. 2.

code. One can prove, using Kleene's recursion theorem, that a tile set can have access to its own code (see [LW08b, DRS08]). Let m be the number of tiles of $\mathfrak{Y}$. M checks that the input contains two codes of tile sets of m tiles, and checks that one of them is the code of $\mathfrak{Y}$. Therefore, the input has to be of the following form: $i_1\$i_2\$\ldots i_m\$i'_m\$i_{m+1}\$\ldots$, where i'_m is the code of $\mathfrak{Y}$.

The second machine N checks for any n such that any of the τ_{i_j}'s, $i_j < n$, can tile a square of size n. If there exists an integer m such that a tile set τ_{i_j} cannot tile a square of size m, then N stops.

The last machine P is a machine that simulates the tile sets of the input, *i.e.*, it generates space times diagrams isomorphic to the tiles of the tile sets (for more detailed explanations on simulation between tile sets, see [LW08a, DRS08]).

To start the simulations of these machines in our tilings, we first force that the only computation tile which exists in the squares of level 1 is the tile, say t_0, representing the initial state of M, N and P. By synchronization, this means that any middle tile of the first line of any square of this construction corresponds also to this tile, and therefore, the computation will begin in any square. We now allow the completion of the first line of any square with tiles representing any letter from the alphabet $\{0, 1, \$\}$. We obtain a tiling where any first line of any square represents the central subword of a bi-infinite input $w \in \{0, 1, \$\}$ and all of these subwords contain in their middle the tile t_0 representing the initial states of the Turing machines.

In all squares of our construction, the computation on the same infinite input is carried out. If one of these machines reaches a final state, then the tiling remains incomplete. Therefore, if the tiling is complete, then M, N and P compute on a word of the form: $i_1\$i_2\$\ldots i_m\$i'_m\$i_{m+1}\$\ldots$, as stated before. Then P simulates any of the tile set i_j, and thus, $\mathfrak{Y}$ totally simulates any tile set τ_{i_j}. Since the index of $\mathfrak{Y}$ is given also in input, then P also simulates a $\mathfrak{Y}$-tiling. In fact, by transitivity of the simulation, it self-simulates infinitely many times. Each time $\mathfrak{Y}$ self-simulates, it also simulates a set of tile sets $\{\tau_{i'_j}\}_{j>0}$ since it simulates a

$\mathfrak{Y}$-tiling that simulates this set. So, a $\mathfrak{Y}$-tiling simulates an infinite number of tile sets of n tiles for any n. Since the set of tile sets of n tiles is finite, and *a fortiori* the set of tile sets of n tiles that tiles the plane, then a $\mathfrak{Y}$-tiling must simulate infinitely many times some tile sets.

4.2 The Construction Gives an Almost Totally Universal Tile Set

We have obtained a tile set $\mathfrak{Y}$ such that any $\mathfrak{Y}$-tiling simulates, for any n, with repetitions, an infinity of tile sets composed of n tiles. We now show that this tile set $\mathfrak{Y}$ is almost totally universal.

Theorem 1. *The tile set $\mathfrak{Y}$ is almost totally universal.*

Proof. Let A be the set of $\mathfrak{Y}$-tilings that simulate at least one tiling for any tile set and $B = \mathcal{T}_{\mathfrak{Y}} \setminus A$, where $\mathcal{T}_{\mathfrak{Y}}$ is the set of $\mathfrak{Y}$-tilings. A is totally universal. We show that A is residual in $\mathcal{T}_{\mathfrak{Y}}$ in both topologies:

We first show that A is residual in $\mathcal{T}_{\mathfrak{Y}}$ (in the Cantor topology) by showing that Player II has a winning strategy in the game $G(A, \mathcal{T}_{\mathfrak{Y}})_C$. In this game, Player I first chooses a $\mathfrak{Y}$-pattern centered around the origin. Player II extends this pattern and so on. Player II wants to obtain a final $\mathfrak{Y}$-tiling that simulates any tile set that tiles the plane. Player I wants to obtain a final tiling such that at least one tile set is never simulated.

Let $\Omega = \{\tau_1, \tau_2, \ldots\}$ be the set of tile sets that tile the plane, and ordered by the number of their tiles first, and then by a lexicographic order of the colors of the tiles. The following strategy is, of course, not recursive since Ω is Π_1. At step n, Player II wants to force the simulation of the n^{th} tile set of Ω. Let m_n be the $\mathfrak{Y}$-pattern played by Player I. Player II wants to force the code of τ_n to appear somewhere in the tiling. When done, by synchronization the final tiling has to simulate a τ_n-tiling. If the code of the tile set τ_n can be written on the input word, then Player II writes this code and forces the simulation of τ_n. Otherwise, we know that $\mathfrak{Y}$ self-simulates infinitely many times, which means that there exists, for any $\mathfrak{Y}$-tiling and for any s, an integer $m > s$ such that $\mathfrak{Y}$ self-simulates with squares of size m. Any of these self-simulations represents a $\mathfrak{Y}$-tiling which simulates other tile sets depending on the infinite input on which it is computing. Therefore, it is enough for Player II to look for the smallest self-simulation where it is possible to write the code of τ_{n+1}. Such a self-simulation always exists. By transitivity of the simulation, this guarantees that the final tiling will simulate τ_{n+1}.

By induction, Player II builds a tiling which simulates at least one tiling for any tile set in Ω. Therefore this tile set is in A, and A is residual in $\mathcal{T}_{\mathfrak{Y}}$ with the Cantor topology.

We now show that B is meager in $\mathcal{T}_{\mathfrak{Y}}$, in the Besicovitch topology. The first move of Player I consists in playing a $\mathfrak{Y}$-tiling P_1 and an integer n_1 to define the open ball $\mathcal{B}_B(P_1, 1/n_1)$. P_1 simulates at least one tile set of one tile. Without loss of generality, we can suppose that this tile set is the first of our enumeration of tile sets that tile the plane (it can be reordered if necessary). Player II wants to be sure that after he has played, the code of τ_1 cannot be removed. This code appears regularly in the tilings which means that there exists an m such that all bits of τ_1 appear in all squares of size m in P_1. If Player II chooses an integer m_1 bigger than m^2 then he is sure that no bits of τ_1 can be changed by Player I since any tiling that has at least one bit of the code τ_1 changed is at least at distance $1/m^2$ of P_1 (by synchronization, changing one bit corresponds to changing one bit in any square of size m).

We now suppose that Player II has already chosen a tiling P_i that simulates all the tile sets in $\{\tau_1, \tau_2, \ldots, \tau_{i-1}\}$ and has chosen an integer big enough to force Player I to play a tiling which simulates also $\{\tau_1, \tau_2, \ldots, \tau_{i-1}\}$. Player I chooses a tiling P_i and an integer n_i. We show that Player I can choose a tiling $Q_i \in \mathcal{B}_B(P_i, 1/n_i)$ that simulates all of $\{\tau_1, \tau_2, \ldots, \tau_{i-1}\} \cup \{\tau_i\}$.

We first make some remarks. If a $\mathfrak{Y}$-tiling P simulates a tile set τ, it means that there exists a level of squares j in P where the simulation of the tiles of τ is made. Of course, not all the tiles of P are concerned by this simulation. We can bound the proportion of tiles that are concerned by this simulation. Indeed, only the tiles which are in squares of level j, and the tiles which are in communication zones between these squares are influenced by this simulation. Therefore the bound is close to 3/4. The exact proportion is not important, since we just need the proportion of tiles concerned by a simulation to be strictly less than 1.

Let S be a $\mathfrak{Y}$-tiling. $\mathfrak{Y}$ self-simulates, therefore S simulates a $\mathfrak{Y}$-tiling, say S_1. By the previous remark, at most 3/4 of the tiles of S are used to simulate S_1. Since S_1 is also a $\mathfrak{Y}$-tiling, then S_1 simulates a $\mathfrak{Y}$-tiling, say S_2. 3/4 of the tiles of S_1 are used to simulate S_2, and by transitivity, $(3/4)^2$ of the tiles of S are used to simulate S_2. By induction, we obtain a sequence $\{S, S_1, S_2, \ldots\}$ of $\mathfrak{Y}$-tilings such that $(3/4)^n$ of the tiles of S are used to simulate S_n.

Because of this remark, Player I can modify P_i such that it simulates a new $\mathfrak{Y}$-tiling S_t by changing a proportion of tiles in P_i smaller than $1/n_i$. This tiling S_t has the particularity of having the code of the i^{th} tile set of Ω in its input and thus, simulates τ_i. Player II plays this tilings Q_i which is at distance less than $1/n_i$ of P_i and which simulates τ_i. Any index of the different tile sets of $\{\tau_1, \tau_2, \ldots, \tau_{n+1}\}$ appears, or is simulated, regularly in the tiling P_{2n+2}: there exists an integer m such that any bit of these indexes appears in all squares of size m. As before, if Player II chooses an integer greater than m^2, he guarantees that none of these tiles can be changed, and therefore, the only possibility for Player I is to choose a $\mathfrak{Y}$-tiling that simulates any tile set of the set $\{\tau_1, \tau_2, \ldots, \tau_{n+1}\}$.

By induction, the tiling obtained at the end of the game is a tiling that simulates all tile sets of Ω. Therefore this tile set is in A, and A is residual in $\mathcal{T}_{\mathfrak{Y}}$ within the Besicovitch topology. □

Acknowledgements

We warmly thank Guillaume Theyssier who, in the first place, convinced one of the co-authors of the possible existence of the previously described tile set. We are also indebted to Bruno Durand for his pertinent remarks on previous work that undoubtedly encouraged us in pursuing in this direction.

References

[AD96] Allauzen, C., Durand, B.: Appendix A: Tiling problems. The classical decision problem, 407–420 (1996)

[BDJ08] Ballier, A., Durand, B., Jeandel, E.: Structural aspects of tilings. In: Proceeding of the Symposium on Theoretical Aspects of Computer Science, pp. 61–72 (2008)

[Ber66] Berger, R.: The undecidability of the domino problem. Mem. Amer. Math Soc. 66, 1–72 (1966)

[DLS01] Durand, B., Levin, L.A., Shen, A.: Complex tilings. In: STOC, pp. 732–739 (2001)

[DLS04] Durand, B., Levin, L.A., Shen, A.: Local rules and global order. Mathematical Intelligencer 27(1), 64–68 (2004)

[DRS08] Durand, B., Romashchenko, A.E., Shen, A.: Fixed point and aperiodic tilings. In: Developments in Language Theory, pp. 276–288 (2008)

[Han74] Hanf, W.P.: Nonrecursive tilings of the plane. I. J. Symb. Log 39(2), 283–285 (1974)

[LW07] Lafitte, G., Weiss, M.: Universal tilings. In: Thomas, W., Weil, P. (eds.) STACS 2007. LNCS, vol. 4393, pp. 367–380. Springer, Heidelberg (2007)

[LW08a] Lafitte, G., Weiss, M.: Simulations between tilings. In: Beckmann, A., Dimitracopoulos, C., Löwe, B. (eds.) Logic and Theory of Algorithms, 4th Conference on Computability in Europe, CiE 2008, Athens, Greece, June 2008, University of Athens (2008)

[LW08b] Lafitte, G., Weiss, M.: Computability of tilings. In: International Federation for Information Processing, Fifth IFIP International Conference on Theoretical Computer Science, vol. 273, pp. 187–201 (2008)

[LW08c] Lafitte, G., Weiss, M.: A topological study of tilings. In: Agrawal, M., Du, D.-Z., Duan, Z., Li, A. (eds.) TAMC 2008. LNCS, vol. 4978, pp. 375–387. Springer, Heidelberg (2008)

[Mye74] Myers, D.: Nonrecursive tilings of the plane. II. J. Symb. Log 39(2), 286–294 (1974)

[Oxt57] Oxtoby, J.C.: Tilings: recursivity and regularity. Contribution to the theory of games III(39), 159–163 (1957)

[Rob71] Robinson, R.M.: Undecidability and nonperiodicity for tilings of the plane. Inv. Math. 12, 117–209 (1971)

[Wan61] Wang, H.: Proving theorems by pattern recognition II. Bell Systems Journal 40, 1–41 (1961)