Math. Struct. in Comp. Science (2010), *vol.* 20, *pp.* 813–850. © Cambridge University Press 2010
doi:10.1017/S0960129510000228

Tilings: simulation and universality

GREGORY LAFITTE[†] and MICHAEL WEISS[‡]

[†]*Laboratoire d'Informatique Fondamentale de Marseille (LIF), CNRS – Aix-Marseille Université, 39 rue Joliot-Curie, F-13453 Marseille Cedex 13, France*
Email: Gregory.Lafitte@lif.univ-mrs.fr
[‡]*Laboratoire I3S, CNRS – Université de Nice Sophia-Antipolis, Les Algorithmes, Bât. Euclide B, 2000 route des Lucioles, BP 121, F-06903 Sophia Antipolis Cedex, France*
Email: Michael.Weiss@unice.fr

Received 1 October 2009; revised 7 May 2010

Wang tiles are unit-size squares with coloured edges. In this paper, we approach one aspect of the study of tiling 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*) that *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 that do not satisfy the universality condition is meagre in the set of $\mathfrak{y}$-tilings.

1. Introduction

Wang was the first to introduce the study of tilings with coloured tiles where a tile is a unit size square with coloured edges (Wang 1961). Two tiles can be assembled if their common edge has the same colour. The process of tiling consists of assembling tiles from a tile set (a finite set of tiles) on the grid $\mathbb{Z}^2$.

It has been known since Berger (1966) that Wang tilings can simulate Turing machines. As a model of computation, tilings raise computability questions. One of the first, which is related to most models of computation, is the existence of universality.

To approach a problem like this we first need a proper notion of reduction. Lafitte and Weiss (2007) presented a first approach to reduction, and by extension, universality. This notion of reduction is inspired by notions of bulking in cellular automata, which were introduced by Ollinger and Theyssier (see Delorme *et al.* (2010)). Intuitively, a tiling P *strongly simulates* a tiling Q if the tiles of Q can be encoded with macro-tiles of P. By generalising this notion to tile sets, we obtain a notion of simulation that is a pre-order on the set of tile sets. Maxima for this pre-order are said to be *complete*, that is, a tile set is complete if it has the ability to encode in macro-tiles all tiles of any tile set and thus, can simulate any tilings. Completeness can be seen as a first large notion of universality. We will give examples of such complete tile sets using the self-similar hierarchical Robinson construction, which we present in detail, and, in particular, the notion of synchronisation, which enables the carrying out in tilings of computations on infinite inputs.

[†] This work was partly supported by French ANR grants Sycomore, NAFIT and EMC.

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

The notion of simulation was then improved in Lafitte and Weiss (2008b) (a closely related definition was also introduced in Durand *et al.* (2008)) to obtain finer 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 the normal simulation, the second condition is not needed). A notion of simulation between tile sets is a first step in the direction of having a universality notion and instance. The total simulation extends the notion of completeness and yields a pre-order on tile sets. The questions that naturally arise are: What can we do with total simulations? Which classes of tile sets have a totally universal instance?

Lafitte and Weiss (2008b) proved that there exists a tile set that totally simulates any periodic tile set, that is, 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 posed: does there exist a tile set totally simulating any tile set that tiles the plane? Because of the number of properties that such a tile set would have (maximal Kolmogorov complexity (Durand *et al.* 2001), non-recursivity (Hanf 1974; Myers 1974), self-similarity of any kind (Durand *et al.* 2008), Turing universality, invariance by recursive modification (Lafitte and Weiss 2008a), ...), Lafitte and Weiss (2009) conjectured that it does not exist. In the current paper, we show that it is indeed the case that such a tile set does not exist, because its existence would bound the quasi-periodicity function (which is in contradiction with a result of Cervelle and Durand (2004)).

Knowing this, our goal is to obtain something that is stronger than completeness for tile sets, but, of course, not as strong as total universality. Our idea, in this paper, is to combine some of the most complex constructions on tilings to build a tile set ע that is almost totally universal: almost all ע-tilings simulate at least one tiling for any tile set (by *almost* we mean that the subset of ע-tilings that do not satisfy this property is a meagre set in the set of ע-tilings). And, therefore, ע has the particular property of *almost always*[†] having all the properties enumerated above. The construction of ע uses various technical tools, but is mainly based on Robinson's aperiodic and self-similar tile set. A detailed explanation of this construction can be found in Robinson (1971), Hanf (1974), Myers (1974), Allauzen and Durand (1997), Durand *et al.* (2001) and Durand *et al.* (2004). Both Lafitte and Weiss (2008b) and Durand *et al.* (2008) (which ingeniously avoided the use of Robinson's tilings construction) showed how a Turing machine can be used to simulate a tile set, in the sense that the Turing machine produces space×time diagrams isomorphic to the tiles of a tile set. This simulation of Turing machines enables the simulation of tile sets in Robinson's tiling. The other tool used for the construction is synchronisation. This principle was first used by Hanf and Myers to build a non-recursive tile set, that is, a tile set that generates only tilings that cannot be defined by a recursive function (Hanf 1974; Myers 1974). We show how to make a synchronisation between squares in Robinson's construction in a new way. Using all these various tools, we are able to construct an almost totally universal tile set.

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

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

The final step is to prove the *almost* part. One of the main tools for proving the meagreness of a set is the use of topological games, such as Banach–Mazur games (Oxtoby 1957). This perfect information game is played by two players on a topological space. A classical result for Banach–Mazur games shows that if Player II has a winning strategy, then A is meagre (or, equivalently, the set $X \setminus A$ is residual).

Lafitte and Weiss (2008c) p1resented a topological study of tilings, and introduced games *à la* Banach–Mazur on them. These games are played on two topological spaces:

— the Besicovitch space, where the distance between two tilings is defined as the asymptotic proportion of different tiles between them; and
— the Cantor space, where the distance between two tilings is related to the biggest pattern centred 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 prove, using these games, that the tile set $\mathfrak{Y}$ is almost totally universal.

In Section 2, we recall the basic notions for tilings, and the notions of simulation between tile sets. In Section 3, we explain Robinson's self-similar construction and how we can synchronise its squares. In Section 4, we consider the notions of universality and completeness, and build two complete tile sets. In Section 5, we define a more restrictive notion of simulation between tile sets: the total simulation. To obtain an almost totally universal tile set, we define topological spaces on tilings in Section 6, prove some of their properties and introduce Banach–Mazur games for tilings. In the final section, we build the tile set $\mathfrak{Y}$ and show that it is almost totally universal.

This paper is an expanded version of the TAMC 2009 conference paper Lafitte and Weiss (2009), and includes all the nooks and crannies of the tiling notions required for the construction of an almost totally universal tile set. These notions were originally introduced, but in a less detailed and sometimes less precise way, in the conference papers Lafitte and Weiss (2007; 2008b; 2008c; 2009).

2. Notions of the completeness and universality of tilings

2.1. *Basic definitions and notions of simulation*

First, we give the definitions for the basic notions of tilings. A *tile* is an oriented unit size square with coloured edges from C, where C is a finite set of colours. A *tile set* is a finite set of tiles. The process of *tiling* consists of placing the tiles of a given tile set on the grid $\mathbb{Z}^2$ such that two adjacent tiles share the same colour along their common edge. Since a tile set can be described using a finite set of integers, we can enumerate the tile sets, and τ_i will designate 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 with a tiling function f_P where $f_P(x, y)$ gives the tile at position (x, y) in P. In this paper, when we say 'Let P be a tiling', we mean that there exists a tile set τ such that P is a τ-tiling of the plane. The set of all tilings is denoted by $\mathscr{T}$. The set of tile sets generated by a given tile set τ is denoted by $\mathscr{T}_\tau$.

Some basic properties of tilings can be defined as follows. We say that a tiling P is *periodic* if there exist two independent vectors $u, v \in \mathbb{Z}^2$ such that $f_P(x) = f_P(x + u) =$

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

$f_P(x+v)$ for any $x \in \mathbb{Z}^2$. If a tile set generates at least one periodic tiling, the tile set is said to be *periodic*. If it tiles the plane but none of its tilings is periodic, it is said to be *aperiodic* or *non-periodic*. Finally, a tiling P is *non-recursive* if its tiling function f_P is non-recursive. A tile set that generates only non-recursive tilings is said to be *non-recursive* (see Hanf (1974), Myers (1974) and Durand *et al.* (2001) for examples of non-recursive tile sets).

A pattern m is a finite tiling. If it is generated by a tile set τ, we say it is a τ-pattern. A pattern m is associated with a tiling partial function f_m, defined on a finite subset of $\mathbb{N}^2$ such that $f_m(x,y)$ gives the tile at the position (x,y) in m.

Aperiodicity can be quantified by the quasiperiodicity function. For a tiling P, we define the quasiperiodicity function G_P as follows: for an integer n, $G_P(n)$ is equal to the smallest m such that any pattern of P of size n appears at least once in each pattern of P of size m. If no such m exists, $G_P(n)$ is not defined. Therefore, the quasiperiodicity function quantifies the regularity of a tiling. A tiling that has a quasiperiodicity function defined for any integer n is said to be quasiperiodic. A result that has now acquired classical status states that if a tile set tiles the plane, then it generates at least one quasiperiodic tiling (Durand 1999).

A subset L of $\mathbb{Z}^2$ is said to be a *lattice* of $\mathbb{Z}^2$ of vectors u and v if there exist $x,u,v \in \mathbb{Z}^2$, with $|x| < |u|$ and $|x| < |v|$, such that $L = \{x + k_1u + k_2v \mid k_1,k_2 \in \mathbb{Z}\}$. We say that L is a lattice of $\mathbb{Z}^2$ of sizes a and b when L is a lattice of vectors $(a,0)$ and $(0,b)$. We use $(c,d)_L$ to denote the point of the lattice L with coordinates $(x + c.u + d.v) \in \mathbb{Z}^2$. The *cut* of a tiling P with respect to a lattice L of sizes a and b is the set of $a \times b$ patterns $M_{(c,d)}$ of P of domain $[c,c+a] \times [d,d+b]$, that is, the pattern tiling function $f_{M_{(c,d)}}$ of $M_{(c,d)}$ is the restriction of the tiling function f_P to $[c,c+a] \times [d,d+b]$. One can see that the intersection of the domains of two different elements of the cut is empty and that the union of all the domains of the cut is $\mathbb{Z}^2$. The *cut function* $f_{P,L}$ of a tiling P with respect to a lattice L is the function that takes as input a pair of integer (c,d) and outputs the pattern $M_{(c,d)}$ of P.

The first simulation notion we will introduce is strong simulation. The intuitive idea is to say that a tiling Q *reduces strongly* to a tiling P, or that P *strongly simulates* Q, if there exist two integers a and b, a function R, called a *reduction function*, from the $a \times b$ patterns of P to the tiles of Q and a lattice L of sizes a and b such that if we replace the patterns of the cut of P with respect to L by their corresponding tiles given by R, then we obtain Q. One can see that P does with *macro-tiles*, that is, rectangular patterns that represent the tiles of another tile set, what Q does with tiles.

This construction can be extended to any possible lattice. Since there are an infinite number of different lattices, a tiling P can be cut in an infinite number of different ways. To each of these cuts, we can apply a reduction function and the tiling P simulates another tiling. Therefore, a tiling simulates an infinite number of other tilings, and it is from this infinite number of different simulations that we expect to obtain a notion of universality for tilings.

We formalise the notion of reduction between tilings in the following definition.

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

Definition 2.1. Let P be a τ-tiling and Q be a τ'-tiling. Q *reduces strongly* to P, or P *simulates* Q, denoted by $Q \trianglelefteq P$, if there exist two integers a and b, a lattice L of sizes a and b, and a reduction function R from the $\{a \times b\}$ patterns of P to the tiles of Q such that

$$f_Q = R \circ f_{P,L},$$

where f_Q is the tiling function of Q and $f_{P,L}$ is the cut function of P with respect to L.

With the same notation, we say that Q *reduces weakly* to P, denoted by $Q \preccurlyeq P$, if for any pattern M of Q there exists a vector $c \in \mathbb{Z}^2$ such that

$$f_M(x) = R \circ f_{P,L}(x + c), \ \forall\, x \in \mathrm{Dom}(M),$$

where f_M is the pattern tiling function of M defined on $\mathrm{Dom}(M) \subset \mathbb{Z}^2$.

When we want to specify R, we use $\trianglelefteq^R$ or $\preccurlyeq^R$ to denote the reduction.

As seen above, if $Q \trianglelefteq P$, we can cut P with respect to a lattice L such that the patterns of the cut represent the tiles of Q. The important thing is that different patterns of the cut of P can represent the same tile of Q. But the converse is impossible by definition (a pattern of the cut of P cannot represent different tiles of Q). For the weak reduction, if $Q \preccurlyeq P$, we can cut P and replace its patterns by tiles of Q such that the tiling obtained contains any pattern of Q. This definition can be extended naturally to reductions between patterns. Those reductions are both preorders on $\mathscr{T}$, where $\mathscr{T}$ is the set of all possible tilings, since transitivity and reflexivity are respected but antisymmetry is not.

2.2. *Completeness and universality*

The notion of simulation described in the previous section deals with tilings. It would also be useful, and even more interesting, to have a notion of simulation between tile sets. The following definition is a first step in this direction; we will define a more restrictive notion of simulation between tile sets in Section 5.

Definition 2.2. A tile set τ simulates a tile set τ' if there exist two integers a, b and a reduction R from a set of τ-patterns of size $a \times b$ to the tiles of τ' such that:

(1) For any τ' tiling P, there exists a τ-tiling Q such that $P \trianglelefteq^R Q$.
(2) There exist two integers c, d such that if a τ-tiling Q contains a pattern A of size $c \times d$ that simulates a τ'-pattern B, then there exists a τ'-tiling P containing B such that $P \trianglelefteq^R Q$.

We use $\tau' \trianglelefteq \tau$ to denote this reduction[†].

This definition needs some explanation. The first condition states that a tile set τ simulates another tile set τ' if there is a set of τ-patterns that behave as the tiles of τ'. Therefore, τ can generate tilings that simulate τ'-tilings. However, we expect this simulation

[†] The same symbol '$\trianglelefteq$' is used for the simulation between tilings and between tile sets, but no confusion should arise since one use is only between tilings and the other is only between tile sets, and is a generalisation of the previous one.

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

to be *natural*; if the simulation is just defined with the first condition, then a trivial tile set that encodes colours would simulate any tile set. The second condition eliminates these *artificial* simulations. If a τ-tiling Q contains a pattern big enough to simulate a τ'-pattern, then Q simulates a τ'-tiling. This means that if a simulation begins somewhere in the tiling, then the simulation spreads naturally to the whole tiling.

This notion of simulation between tile sets gives a pre-order on the set of tile sets. In the following definition, we define maxima for this pre-order.

Definition 2.3. Let $\mathscr{A}$ be a set of tile sets. A tile set τ is *$\mathscr{A}$-complete* if for any tile set $\tau' \in \mathscr{A}$, there exists a reduction function R such that $\tau \preccurlyeq^R \tau$. If $\mathscr{A}$ corresponds to the set of tile sets that tile the plane, then τ is said to be *complete*.

This completeness notion is natural in the sense that it corresponds to what one would expect: any tiling can be reduced to some instance tiling of our complete tile set in such a way that to answer any question about our tiling it suffices to study the instance tiling of our complete tile set.

We expect that a universal tiling will have in its construction much of the information of all the other tilings. There are several ways of defining the information contained in a given tiling, but we can distinguish two main kinds of information for a tiling:

— The first, and most natural, is the tiling itself (how it is built).
— The second consists of studying the different patterns that appear in the tiling.

These two ways of looking at the information contained in a tiling then lead to two ways of considering universality:

— Does a tiling contain enough information to explain the behaviour of all other tilings (we call this *strong dense universality*), or only the behaviour of a tiling for each tile set (*strong sparse universality*)?
— Does a tiling contain enough information to simulate all the patterns of any tiling (*weak dense universality*) or only the patterns of a tiling for each tile set (*weak dense universality*)?

With these considerations in mind, we have the following definitions.

Definition 2.4. Let P_u be a τ-tiling. P_u is:

— *strongly dense universal* if for any tiling Q, we have $Q \preccurlyeq P_u$;
— *strongly sparse universal* if for all τ', there exists a τ'-tiling Q, such that $Q \preccurlyeq P_u$;
— *weakly dense universal* if for any tiling Q, we have $Q \preccurlyeq P_u$;
— *weakly sparse universal* if for all τ', there exists a τ'-tiling Q, such that $Q \preccurlyeq P_u$.

We now have the following straightforward properties.

Lemma 2.1.

(1) P_u strongly (respectively, weakly) dense universal $\Rightarrow$ P_u strongly (respectively, weakly) sparse universal.
(2) P_u strongly dense (respectively, sparse) universal $\Rightarrow$ P_u weakly dense (respectively, sparse) universal.
(3) $\preccurlyeq$ preserves universalities.

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

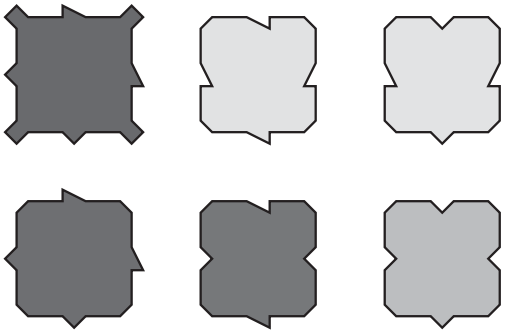


Fig. 1. Robinson's tile set.

(4) $\preccurlyeq$ preserves weak universality.
(5) $\trianglelefteq$ preserves completeness.

We will show the existence or the non-existence of these universality notions later in the paper after we have described Robinson's construction in the following section.

3. Robinson's construction

In this section, we recall the basics of Robinson's construction. This construction is one of the most commonly used for carrying out Turing simulations. We show the aperiodicity of the tilings generated by this tile set and how we can simulate a Turing machine with it. Then we explain in detail the principle of synchronisation that permits the simulation of a Turing machine on an infinite input in a tiling. Synchronisation, which was first used by Hanf and Myers to build a non-recursive tile set (Hanf 1974; Myers 1974), will be one of our main tools for constructing the almost totally universal tile set in Section 7. We propose our own version of synchronisation here, which is adapted for our purposes.

3.1. *Robinson's tile set*

Berger was the first to build an aperiodic tile set (Berger 1966), which was based on Wang tiles. This construction showed the undecidability of the domino problem. Robinson (1971) presented a new aperiodic tile set composed of six different shapes that can be turned and flipped. Figure 1 shows the original six tiles. Robinson proved that his tile set can only generate self-similar tilings. We will only give a brief explanation of the structure of the construction – for a detailed explanation, see Allauzen and Durand (1997). All of Robinson's tilings contain a hierarchy of squares of ever-increasing size. Nowadays, it is more common to represent Robinson's tilings using Wang tiles with arrows, since tiles with arrows underline the self-similar structure of the tilings. Figure 2 shows the hierarchical structure of Robinson's tilings. Some notation will be useful for these tilings. The smallest of the squares appearing in a Robinson tiling are said to be of level 1, and those of level 2 are the squares composed of three tiles on their sides. The construction implies that any central tile of a square of level n defines one at the corner of a square of level $n + 1$. By induction, the central tiles of four squares of level n can define the four

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

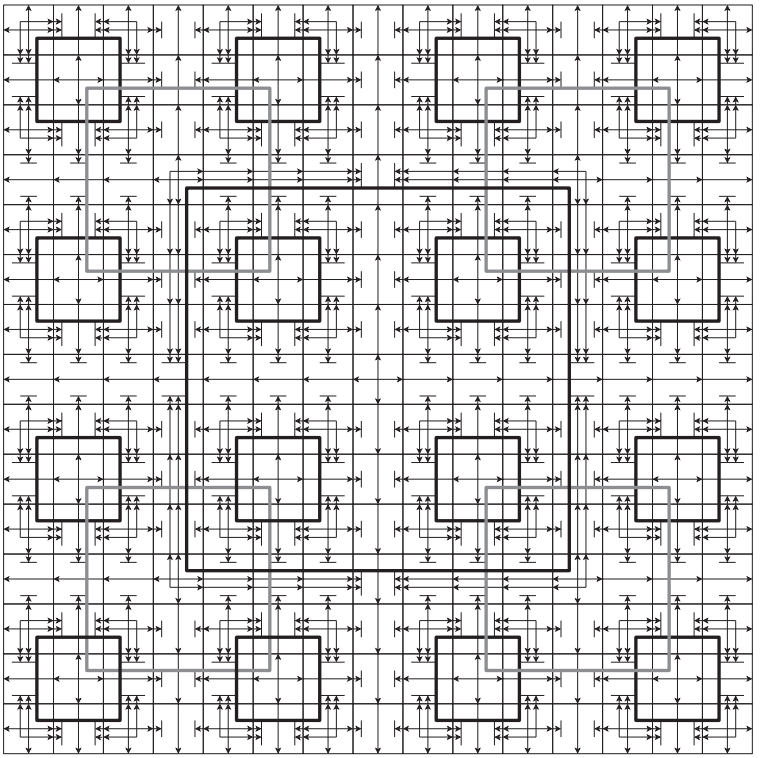


Fig. 2. Robinson's tiling.

corners of a square of level $n+1$. Therefore, one can see that the size of a square of level n is 2^n+1.

We use two different colours for the borders of Robinson's squares. The even-level squares are coloured in black and the odd-level squares are coloured in grey (except the level 1 squares, which only represent the angles of the level 2 squares). Therefore, black squares can only intersect grey squares, and *vice-versa*. This condition can be encoded at the level of tiles. In this way, we obtain the tiling of Figure 2, where black and grey squares alternate. The final improvement in the construction was made in Durand *et al.* (2004), where a robust Robinson tile set was made. In this robust version, the tile set only generates tilings in which all squares of the same level lie on a regular grid (in the non-robust version, some tilings can contain squares that do not match up with another square at the same level).

3.2. *Tilings simulating Turing computation*

Tiles are a natural way to transmit information on the plane. Since one can consider the space×time diagram of a Turing machine as a geometrical object where some special information is computed, we can simulate a Turing machine with a tile set with the following different types of tile:

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

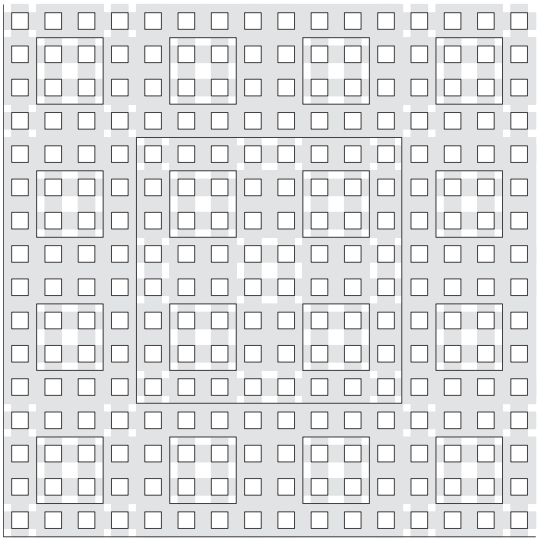


Fig. 3. The obstruction zone (in light-grey) in Robinson's tiling.

— tiles used to transmit a symbol of the Turing machine's alphabet;
— tiles used to show that the state q_i will act on the symbol a_j at the next step;
— tiles used to represent the transitions of the Turing machine;
— a final set of tiles used to encode the initial configuration of the computation, that is, the input word and initial state.

More details of this construction can be found in Allauzen and Durand (1997). Figure 4 shows the different kinds of tile needed to simulate a Turing machine:

— the first tile corresponds to a cell where no modification is made;
— the second and third tiles correspond to the transitions $\lambda(0, q_i) = (1, q_k, R)$ and $\lambda(1, q_j) = (1, q_l, L)$;
— the final two tiles show that a state will act on the cell at the next transition.

Given a Turing machine M and an input w, one can compute the tile set $\tau_{\langle M,w\rangle}$ that simulates the computation of M on w. But since tilings have no origin, we cannot force the tiling to begin in a certain state on a given input. Therefore, we cannot guarantee that a tiling generated by $\tau_{\langle M,w\rangle}$ will respect exactly the computation of M on w.

The main idea is to use the hierarchical pattern construction generated by Robinson's tilings and then compute a given Turing machine in the even-level squares, that is, the black ones, since black squares of different levels do not intersect. This hierarchy of squares can be seen to have spaces where a computation can be made. The main problem is that a level n square also contains lower-level squares. Therefore, if any square contains the simulation of a Turing machine, some tiles will contain arbitrary large superpositions of computation, which is not possible since it would need an infinite set of tiles.

To avoid this problem, and to have at most one computation in any tile, each even-level square sets an obstruction colour outside its border. It is intuitively clear that we can do this with a tiling, but proving it is quite technical. Figure 3 shows the hierarchical

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

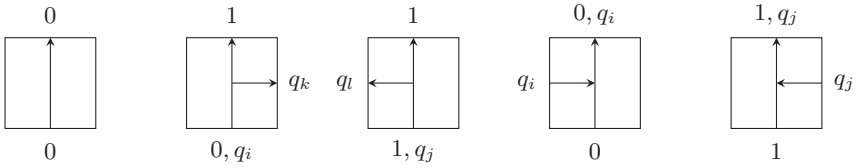


Fig. 4. Turing machine by a tiling.

construction with the obstruction tiles. In this figure, there are some white spaces, that is, areas without the obstruction colour. Intuitively, these spaces can send information vertically and horizontally. These white spaces are the computation spaces, where the Turing machine M will be simulated. On each computation space, we superimpose the tiles representing our Turing machine. The tiles that are obstructed in one direction (horizontally or vertically) transmit the information of the computation of our Turing machine in the other direction. One can see that the tiles that are obstructed in both directions have no information to transmit. Thus, we can use the lowest computation space of a square to start the simulation of our Turing machine on an empty tape with the initial state. Then, by transmission of the information horizontally and vertically, we will simulate in each square of level n, the first steps of our Turing machine.

3.3. *Synchronisation between squares of the same level*

Our first goal in obtaining a synchronisation between all squares is to prove that all the squares of the same level have the same information, that is, all the squares of a certain level in Robinson's construction have the same input word w on their first line. Note that two neighbouring squares, either horizontally or vertically, can share the information they have on their facing sides; indeed, the obstruction colour between them can be used to transmit information and therefore compare the words written on the two facing sides (a more detailed explanation of this fact is given for the construction in Section 4.1). Thus, we need to prove that we can obtain a square that has on its four sides the same input word w. Then, each square will check if its four neighbouring squares are computing using the same input word.

Vertical and horizontal bit transmission is quite easy. We can add the following condition: a tile in a square can be labelled either with a 1 or a 0; all the tiles in the same column must have the same label, and the label of a column must match with the letter of the input word, which is written on the bottom (south) line in binary, that is, if the i^{th} letter of the input word is a 0, then all the tiles of the i^{th} column have to be labelled with a 0. Therefore, the input word is written with labels on the top (north) side. Using the same construction, we can guarantee that the east and west sides have the same input word.

We now need to pass the bits of the input word from the south side to the west side to obtain a square with the same input word on its four sides. The information going from the south to the west side will pass through three kinds of tiles: it first goes through tiles that transmit the information vertically, then turns a corner and, finally, goes through tiles that transmit the information horizontally until it reaches the west side. The only

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

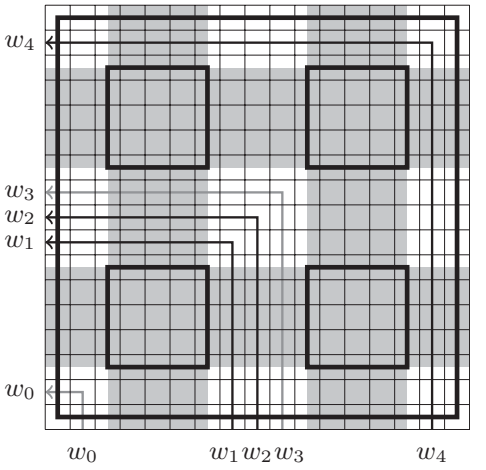


Fig. 5. The transmission of the bits of w from the south side to the west side.

condition we need to add to ensure that all the bits will pass from the south to the west side (as in Figure 5) is to force any tile that is not obstructed (obstructions are coloured in grey in Figure 5) to be one of the three kinds of tiles that transmit information. The obstructed tiles can either transmit information vertically or horizontally, or transmit nothing. Finally, since neighbouring squares of the same level can check if they are computing on the same input word, all squares at the same level compute on the same input word.

3.4. *Synchronisation between levels*

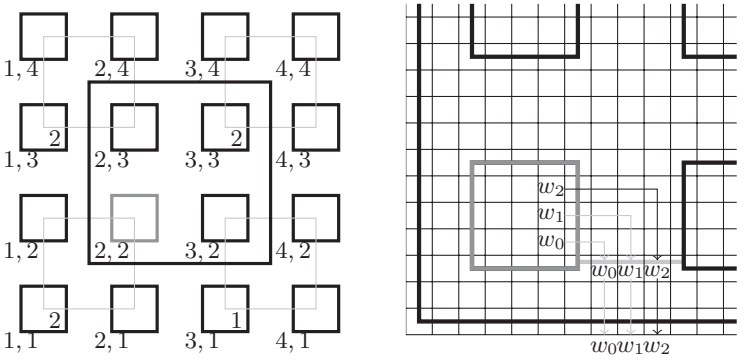
We now want to synchronise the input word between different levels, that is, if w_i is the input word of the level i squares, then w_i is the central word of w_j for any $j > i$, that is, there exist two words w_1 and w_2 of the same length such that $w_j = w_1 w_i w_2$. In this way, all squares at all levels obtain the same computation. Recall that in Robinson's tilings the even-level squares are coloured in black and the odd-level squares are coloured in light grey.

Note that a black level n square intersects exactly with four light-grey level $n-1$ squares, which intersect with sixteen black level $n-2$ squares (four of them are inside the level n square – see Figure 6.a). From the previous section, we know that the sixteen black squares are synchronised and are computing on the same input. Therefore, we can choose one of these sixteen squares to transmit its input word to the higher-level square.

To achieve this goal, we give sixteen different labels to the black squares (one of them is labelled in grey and is the square that transmits its input word to the higher level) and two kinds of label to the light-grey squares. If the labels are made as in Figure 6.a, it is enough to guarantee that any level n black square has a level $n-1$ grey square at its south-west corner. It is this grey square that passes the information from its east side to the south side of the level n square.

To pass the information, we use the induction process of Figure 6.b. We can use the same technique as before because the number of columns between two neighbouring

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press



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

(b) The synchronization between squares of different levels

Fig. 6. Synchronisation between levels.

squares is the same as the number of columns in a square. Then, using an induction process, we will pass all the bits from the east side of the grey square to the middle part of the south side of the black square at the higher level.

Since the synchronisation of squares at the same level and the communication between levels is defined for each level, any square of a final tiling generated by this construction computes on an initial segment of the same infinite input. The main interest in this sort of construction is that it allows us to put a condition on this infinite input to obtain tilings with some 'difficult' properties.

4. Universality and completeness

In this section, we show the existence and non-existence of the various kinds of universality and completeness. We first construct two complete tile sets with the help of Robinson's construction:

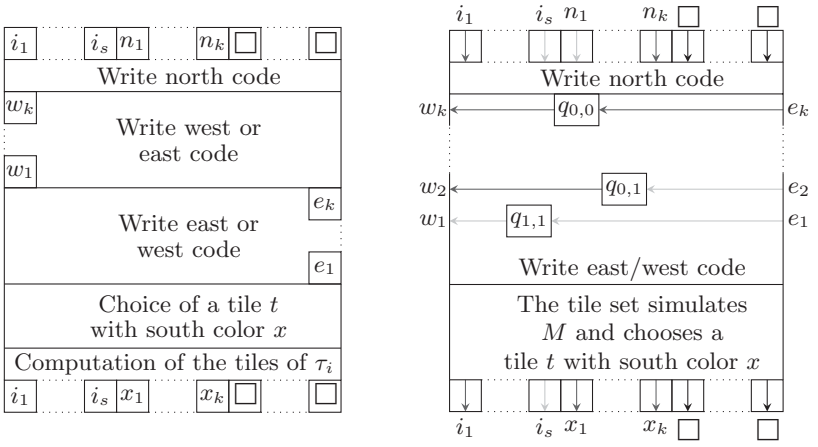
— The first is from Lafitte and Weiss (2007) and uses Robinson's construction only. However, the construction is technical and thus less intuitive.
— The second is a new complete tile set that uses synchronisation in Robinson's construction, and even though it requires synchronisation, it is more basic and gives a better example of completeness.

We will then study the various properties related to universality.

4.1. *Completeness*

4.1.1. *The first complete tile set.* Completeness is an interesting property for tilings, and close to an intuitive notion of universality for tile sets. A tile set is complete if it has the ability to generate enough information to rebuild the behaviour of any other tile set. Therefore, we can expect that complete tile sets are quite difficult to build and require some technical constructions. The first complete tile set was built in Lafitte and Weiss (2007).

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press



(a) Simulation of a tile with a Turing machine

(b) Simulation of a tile with a tile set

Fig. 7. Simulation between tile sets.

In this construction we build a non-deterministic Turing machine M such that the space×time diagrams of the computations of M on inputs of size s give rectangular patterns that simulate tiles from the tile set τ_s (the s^{th} tile set).

Our Turing machine works with a semi-infinite tape. A typical input w is composed of two integers written in binary: i, the code of a tile set, and x, the code of a colour (which belongs to the set τ_i of colours). We know that we can encode a tile set, that is, we can assign a unique number or code in the same way that we do for Turing machines. So the first step for M consists of decoding i to find the different tiles that compose τ_i. Then we check if x is the code of a colour of τ_i. If it is, we choose in a non-deterministic way a tile t of τ_i such that the colour of its south side is x. We can build our Turing machine such that after m steps of computation, the next k steps of the computation are used to write the code of the west/east[†] colour of t (with m and k depending only on $|\omega|$), that is, in the space×time diagram of the Turing machine, the first column from time $m+1$ to $m+k$ represents the code of the colour of the west side of t. The tiles that are not between the $(m+1)^{th}$ and the $(m+k)^{th}$ lines are all blank tiles, (represented by □). We do the same for the east/west side (depending on the first choice we made), that is, in the space×time diagram of the Turing machine, the last column from time $m'+1$ to $m'+k$ represents the code of the colour at the east side of t (with m' depending only on $|\omega|$). The final p steps of the computation are used to write the code of the colour at the north side of t completed with □'s. For later use, we will adopt the convention that our Turing machine M does nothing when the entrance is the empty word. This means that its corresponding space×time diagram will be a rectangular space filled with blank tiles. Figure 7 (a) represents the space×time diagram of our Turing machine.

[†] Our algorithm chooses in a non-determistic way either the west or the east side. This non-determinism is essential for an east/west matching of two diagrams.

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

We now add the condition that M uses the same time to reach a final state if given as its input two valid words containing the same code i of a tile set. With this construction, we are certain that two space×time diagrams have the same size if and only if the inputs have the same lengths. This guarantees that the simulations of two tiles from the same tile set give two diagrams of the same size. So, two diagrams will match on their north/south side if and only if the two tiles that they simulate match on their north/south side. For the east/west border, the match rules are different. During the computation, we can choose whether to write the east or west colour first. So two diagrams will match on their east/west border if and only if the tiles they represent match on their east/west border, and if during the computation, the two Turing machines they represent have made different non-deterministic choices.

The idea is to associate with M the tile set τ_M that simulates M (as seen in Section 3.2), and to inject it into Robinson's construction in such a way that all the squares at the same level contain the simulations of tiles of the same tile set. We begin by improving the tile set τ_M since we do not need to make a non-deterministic choice between west and east sides with tilings. The tile set τ_M simulates M except when M has to make this choice. Indeed, tilings can *send* information in both directions: information is coded in dark grey or light grey arrows, representing the bits 0 or 1, and neighbouring tiles must have the same colour to be assembled. Therefore, one state can be used to write the bits to both the east and west sides, as shown in Figure 7 (b).

We now simulate τ_M in any square of Robinson's construction: the first row of any of these squares can be filled arbitrarily with 0, 1 and the empty letter $\square$. Then, the computation checks if the input is valid. If it is invalid, it reaches a final state represented by a tile that cannot be matched on its south side with another tile; in this case, the tiling remains uncompleted. Therefore, a valid tiling of the plane with this tile set is composed of Robinson's squares containing either the simulation of a tile or nothing (if the input word is empty, then M does nothing). We now have to check that two neighbouring squares at the same level are simulating two tiles of the same tile set that match. To guarantee that the tiles match, the squares need to share the code of the colour they have on their common side. To do this, we can use the obstruction colour. We just need to add labels representing the bits of the codes of the colours to the obstruction colour. Therefore, neighbouring squares can check if they have the same code on their common side. If they do not, the tiling remains uncompleted. Figure 8 shows the communication between two squares. The final thing we need to do is guarantee that all the squares at the same level simulate tiles of the same tile set. In fact, we will show that we can build this tile set such that the level i squares can only simulate tiles of the tile set τ_i. One can see that in a level i square, there is a computation space equivalent to a square of size $2^i - 1$ (we take away the lines and columns containing the obstruction colour). Therefore, to be certain that a level i square can only contain the simulation of a tile of τ_i, it is enough to add the simulation of another Turing machine, say N, that on an input i uses exactly $2^i - 1$ steps to stop, and such that the final state of N is represented by a tile that can match only with a tile on the north side of a square. Therefore, N guarantees that the first part of the input is i if and only if we are in a level i square. In this way, we obtain a tile set μ based on Robinson's construction and such that all the level i squares can

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

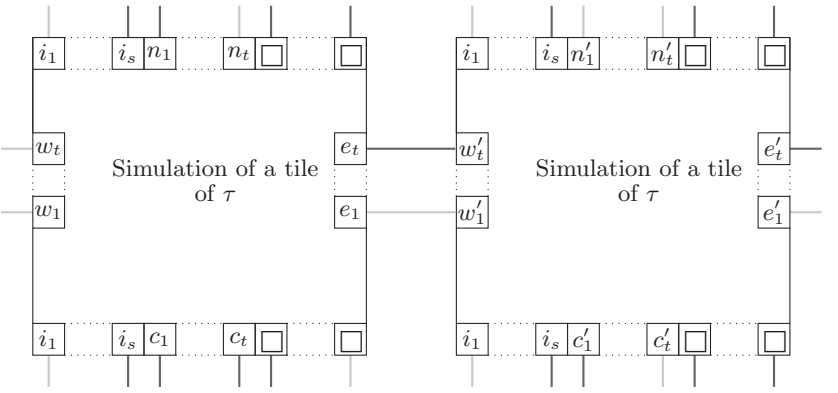


Fig. 8. Communication between Robinson's squares.

simulate a τ_i-tiling only. The simulation is always made *via* the same reduction function. Moreover, if the simulation of a tile set is initiated in a μ-tiling, the simulation is made on the whole tiling. This comes from the fact that if a level i square simulates a tile of τ_i, then all level i squares simulate tiles of τ_i. Therefore, for any tile set τ_i that tiles the plane, we have $\tau_i \trianglelefteq \mu$, which proves the completeness of μ.

4.1.2. *Second complete tile set.* To build our second complete tile set, we use Robinson's construction, but with all the squares at the same level synchronised (as in Section 3.3). This construction is much easier, but uses more technical tools, in particular, synchronisation.

We write the input $i\square t$ on the south side of any square at the same level, where i is the code of a tile set and t is the code of a tile of τ_i (two squares at the same level must have the same code i; the codes t can be different). The tile t is encoded with the cartesian product of four colours from the set of colours of τ_i, that is, $t \in C^4$; where C is the set of colours of τ_i. The first, second, third and fourth components correspond to the codes of the south, north, east and west sides of t, respectively. The code i of the tile set τ_i can be encoded as a sequence of codes t_j representing the tiles of τ_i, that is, $i = t_1 t_2 \dots t_n$ where $t_j \in C^4$ is a tile of τ_i, for all j. Therefore t is a subword of i.

Through synchronisation we can ensure that any square has the same input $i\square t$ on its four sides. Between squares at the same level, only the tile component t can be different (this can be checked by synchronisation of squares at the same level). To guarantee that the local constraint is respected in the simulation, we need to check that two neighbouring squares have the same colour encoded on their common side. The sides of the squares do not send the whole input contained on their side. The component i is always sent totally (for synchronisation reasons), but only one component of t is sent. The south side sends only the first component of t (which corresponds to the south colour of t), the north side sends only the second, the east sends the third and the west sends the fourth. This can be achieved technically by adding two colours to the sides of the square: the first matches with the bits of i, and the second, after the separator $\square$, matches with the bits of t. The border with the first colour sends the whole information, that is, all the bits of i, and the second colour, depending on which side it is on, sends only one of the components of t.

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

Therefore, in a valid tiling of the plane, two neighbouring squares have to share the same colour on their common side.

To guarantee that the level i squares simulate the tiles of τ_i, we use synchronisation between squares at different levels: a level i square *tells* the level $i+1$ square what its level is. By induction, any square *knows* its level and simulates only tiles of the tile set indexed by the number of the level). We also need to add the condition that all the squares of a given level can be marked with an obstruction colour. This obstruction colour prevents simulations of tiles within the squares. Therefore, a level of squares coloured with this obstruction colour does not contain the simulation of a tiling (this condition is necessary since there exist tile sets that do not tile the plane).

To ensure that the input is well written, we simulate a Turing machine that checks that the input is composed of the code of a tile set, a separator $\square$ and the code of a tile. This is enough to obtain a complete tile set since each level i square simulates a tile of τ_i and that the local constraint between squares is respected.

4.2. *Universality*

In this section we will investigate the various universality notions defined in Section 2. We begin by showing a property of universality before constructing our first universal tiling.

Theorem 4.1. Let P be a tiling. If P is strongly universal, then P is non-recursive.

Proof. Let P be a strongly sparse universal tiling. We can enumerate all the tilings that reduce to P by enumerating any possible reduction. If P is recursive, that is, f_P is recursive, we can obtain at least one recursive tiling for any tile set. But this is a contradiction because we know, from results in Myers (1974) and Hanf (1974), that there exist tile sets that only produce non-recursive tilings. $\square$

The first result we get for the various universality notions is the non-existence of strongly dense universal tilings. This follows from a countability argument: we only have a countable set of possible reductions for a given tiling, but an uncountable set of tilings.

Theorem 4.2. Strongly dense universal tilings do not exist.

We will now consider the weak version of universality for tilings. The idea of the construction is to build our tiling in the same way as we can construct a Turing machine simulating every step of all the Turing machines by simulating at step i the first i computing steps of the first i Turing machines. Similarly, it is possible to construct a weakly dense universal tiling that enumerates all possible patterns of all tilings. They are countable, so we can simulate all of them in the same tiling. Thus, we obtain the following result.

Theorem 4.3. There exists a weakly dense universal tiling.

Of course, weak dense universality implies weak sparse universality. There is still one other universality notion to consider: strong sparse universality, that is, a tiling that can simulate at least one tiling for each tile set. In the following theorem, we propose a non-periodic tile set that generates a strongly sparse universal tiling.

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

Theorem 4.4. There exists a non-periodic tile set that generates a strongly sparse universal tiling.

Proof. We show that the complete tile set μ constructed in Section 4.1 can generate a strongly sparse universal tiling. Our tile set μ generates tilings that can either simulate or not simulate a τ_i-tiling, for any tile set τ_i that tiles the plane. In the set of μ-tilings, there are some tilings that simulate at least one tiling for any tile set (since level i squares can either simulate a τ_i tiling or not). These tilings are strongly sparse universal since they simulate at least one tiling for any tile set that tiles the plane. □

4.3. *Classification*

In the previous theorem, we have shown the existence of a strongly sparse universal tiling, which we will denote by P_1. For a given tile set τ_i, we can choose the τ_i-tiling that we simulate in P_1. At a certain step of our computation, our tiling P_1 will simulate its own tile set. We can consider that we will simulate a τ_i-tiling P_2 that is a strongly sparse universal tiling such that P_2 simulates for any tile set τ_i that tiles the plane, another τ_i-tiling than the τ_i-tiling simulated in P_1. Thus, by transitivity, with this construction, P_1 simulates at least 2 tilings for each tile set. At a certain point, P_2 will also simulate its own tile set, and so on. By iterating this process, we can build a tiling that simulates for each tile set τ_i a countable infinity of τ_i-tilings.

The following theorem gives the conditions needed by a tile set to generate a universal tiling that simulates a countable infinity of tilings for each tile set.

Theorem 4.5. Let τ be a tile set. If for any countable set

$$A = \{P_1, P_2, P_3, \ldots \mid P_j \text{ is a } \tau_j\text{-tiling } \forall j\}$$

there exists a τ-tiling P_A such that $P_j \trianglelefteq P_A$ for all $P_j \in A$, then there exists a strongly sparse universal τ-tiling P_u such that for all τ_j there exists a countably infinite set $A_j = \{P_{j_1}, P_{j_2} P_{j_3}, \ldots\}$ of τ_j-tilings such that $P_{j_k} \trianglelefteq P_u$ for all j, k. We say that a tiling has the *universal infinity property* (*UIP*) if it satisfies the conditions of this theorem.

Proof. Let τ be a tile set that satisfies the hypothesis of the theorem. Since τ is a tile set, there exists i such that $\tau = \tau_i$. Consider the set A_1 composed of P_1^1, P_2^1, ... such that, for all j, P_j^1 is a τ_j-tiling and P_i^1 is a τ_i-tiling that simulates all tilings of the set A_2. By induction, we define A_n composed of P_1^n, P_2^n, ... such that for all j, we have that P_j^n is a τ_j-tiling and P_i^n is a τ_i-tiling that simulates all tilings of the set A_{n+1}.

If we choose the sets A_n in such a way that $A_n \bigcap A_m = \varnothing$ for all n, m, then by simulating all tilings of the set A_1, P_u will simulate a countable infinity of tilings for each tile set. □

The following theorem relates this property to universality and completeness.

Theorem 4.6.

(1) If a tile set is complete, it has the universal infinity property.
(2) A tile set has the universal infinite property if and only if it generates a weakly dense universal tiling.

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

Proof.

(1) *Completeness ⇒ UIP*

Let τ be a complete tile set and τ' be a tile set with the universal infinity property. Since τ is complete, for any τ'-tiling P_k, there exists a τ-tiling Q_k such that $P_k \preccurlyeq Q_k$. Since Theorem 4.4 shows the existence of at least one tile set with the universal infinity property, we can reduce all of its tilings to our complete tile set τ, so τ has the universal infinity property.

(2) ($\Rightarrow$):

Let τ be a tile set with the universal infinity property. Thus, for any tiling Q there exists a τ-tiling P such that $Q \preccurlyeq P$. If we take a weakly dense universal tiling Q, there exists a weakly dense universal τ-tiling P since $\preccurlyeq$ preserves weak universality.

($\Leftarrow$):

Let τ be a tile set that generates a weakly dense universal tiling P and τ' be a tile set with the *UIP* property. By definition, any τ'-pattern is simulated in P. Let Q be a τ'-tiling and $\{A_i\}_{i>0}$ be the sequence of patterns of sizes $2i+1$ of Q centred around the origin. There exists a sequence $\{B_i\}_{i>0}$ of patterns of P such that B_i simulates A_i for any i. From this sequence of increasing patterns, we can extract, in the sense of Koenig's Lemma, a tiling P' such that P' simulates Q. Therefore, any τ'-tiling is simulated by a τ-tiling[†]. □

We have shown that the class of complete tile sets, denoted by *Complete*, is the smallest class of our classification. The complete tile sets are contained in the class of tile sets having the *UIP* property, denoted by *UIP*. The notion of completeness is really close to a notion of universality for tile sets.

To clarify this classification, we aim to show that some non-periodic tile sets do not belong to *UIP*. To prove this, we recall the quasiperiodicity function associated with a tiling. For a given tiling P, the quasiperiodicity function G_P gives for each n, the smallest m such that any pattern of size n in P appears at least once in any $m \times m$-square pattern of P. Of course, G_P is not a total function for all P. We can have a tiling Q where a given pattern of size s appears only once in Q, so $G_Q(s)$ will not be defined. Nevertheless, Durand (1999) showed that if a tile set tiles the plane, it can tile it in a quasiperiodic way, that is, the quasiperiodicity function associated with this tiling is total.

We have the following result.

Theorem 4.7. Let P and Q be two tilings of the plane. If there does not exist $c \in \mathbb{N}$ such that $G_Q(cn) > G_P(n)\ \forall n$, then $P \not\preccurlyeq Q$.

Proof. Suppose $P \preccurlyeq Q$ with a reduction of size (a, b). This means that any $c \times d$-pattern of P is simulated by a $ca \times bd$-pattern of Q. By the theorem's condition, there exists at least one n such that $G_Q(a \times b \times n) < G_P(n)$. Thus, there exists a pattern m of P of size n that appears less frequently in P than any pattern of Q of size $a \times b \times n$ appears in Q. Hence, no pattern of Q of size $a \times b \times n$ can represent the pattern m, so $P \not\preccurlyeq Q$. □

[†] This does not mean that τ simulates τ' since we have only proved the first condition of Definition 2.2.

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

From Cervelle and Durand (2004), we know that there exist tile sets that only generate tilings with a non-recursive quasiperiodicity function. Thus, if P is a universal strong tiling, G_P cannot be recursive. Since Robinson's tile set only gives rise to tilings with recursive quasiperiodicity functions, Robinson's tilings are not universal.

5. Simulation between tile sets

The previous section was mainly concerned with completeness, which corresponds to the maximum in the simulation pre-order. This notion can be refined to obtain stronger notions of simulation. In this section we define notions of simulation that make possible the introduction of universality for tile sets. This notion, the *total simulation*, is a finer notion than the simple simulation between tile sets.

5.1. *Definition*

We begin by defining the total simulation. The following definition, which was used in Durand *et al.* (2008) and Lafitte and Weiss (2008b), generalises the simulation to tile sets.

Definition 5.1. Let τ and τ' be two tile sets. We say that τ *totally simulates* 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 the following two conditions are respected:

(1) For any τ'-tiling Q, there exists a τ-tiling P such that $Q \preccurlyeq^R P$.
(2) For any τ-tiling P, there exists a τ'-tiling Q such that $Q \preccurlyeq^R P$.

We denote it by $\tau' \preccurlyeq_t \tau$ (or $\tau' \preccurlyeq_t^R \tau$, to specify the reduction R).

If $\tau' \preccurlyeq_t \tau$, there exists a reduction R such that for any τ'-tiling there exists a τ-tiling that simulates it by R, and, furthermore, τ does not generate tilings that do not simulate a τ'-tiling. This is a strong link between the two tile sets since τ has to simulate all and only the τ'-tilings. Since we do not have conditions on the reduction function R, it is likely that different τ-patterns represent the same tile, which means that a tile set that simulates a periodic tile set is not necessarily also periodic (on the other hand, a periodic tile set can only simulate periodic tile sets).

5.2. *Total simulation of periodic tile sets*

Total simulation offers enough freedom for us to obtain particular tile sets having the property of totally simulating all tile sets with a certain property. In this section, we build a recursive aperiodic tile set that totally simulates each and every periodic tile set, and only them. This tile set is thus totally universal for periodic tile sets. This is an interesting tile set since the problem of knowing whether a tile set is periodic is undecidable. We will then generalise this construction and give sufficient conditions for tile set classes to be totally simulated by a single tile set.

Theorem 5.1. The set of periodic tile sets can be totally simulated by one tile set, that is, there exists a tile set τ such that for any periodic tile set τ', we have $\tau' \preccurlyeq_t \tau$.

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

Proof. First, let f be the projection of a recursive one-to-one function from $\mathbb{N}$ to $\mathbb{N}^2$ such that $f(n) \leqslant n$. Thus, the set $\{f(1), f(2), \ldots\}$ contain any integer infinitely many times. We can make the assumption that f can be computed by a Turing machine M_f in linear time and space (with unary input).

We now build a one-tape Turing machine M with the following behaviour: it takes as input a word written in unary (1^n), computes $1^{f(n)}$ by simulating M_f, and then simulates the tile set $\tau_{f(n)}$ to enumerate the rectangular patterns generated by this tile set. If a periodic pattern is found, the computation stops, otherwise, it never halts.

We now simulate this Turing machine in Robinson's tiling stretched vertically by a factor 2. We have seen that the simulation of the computation of a Turing machine does not depend on the time and space needed to stop. In the square of size $2^n - 1$, we simulate M_f on the greatest input 1^m whose computation fits in this square. It thus outputs $1^{f(m)}$. Since Robinson's tiling is stretched with a factor 2, we still have remaining spaces where we can compute M.

We simulate M in this tiling with the special condition that if a final state is reached, a special colour is sent to the north side of the rectangle in which the computation is made, which forces the whole border of the rectangle to be marked with this special colour. If no final state is reached, the computation keeps on going until it matches with the north side of the rectangle and the special colour cannot be added to the sides of this rectangle.

This special colour triggers the beginning of the computation of a new Turing machine N in this rectangle. The behaviour of N is as follows. An input is an integer x (the index of a tile set) and an integer y (the index of a colour). We check whether y is a valid colour of the tile set τ_x. If it is, we choose in a non-deterministic way a tile of τ_x with south colour y, and we simulate this tile, that is, we build a space×time diagram which is a pattern representing this tile.

We simulate N in our rectangle with the following conditions:

— The computation of N can begin if and only if the rectangle is marked with the special colour.
— The first line of the rectangle marked with the special colour is filled in a non-deterministic way with the letters of the alphabet. The computation of N will say whether the input was valid or not, and if it is not, the computation stops and the tiling cannot be completed to constitute a full tiling of the plane.
— Since the choice of x and y are made in a non-deterministic way, we have to check that x is the code of the tile set that M_f has computed before (even if it was not actually computed *before* since in our tiling the computations are made simultaneously). To do this, the bits of x are sent to the north and have to match with the value of $1^{f(n)}$ to let the computation go on.

Therefore, at the end, the computation of N is carried out in the rectangle if and only if M has stopped, that is, if the tile set $\tau_{f(n)}$ is periodic, and if and only if N has simulated a tile of $\tau_{f(n)}$.

The final stage sends the colours of the simulation of the tiles of $\tau_{f(n)}$ outside the rectangle to ensure that two Robinson rectangles simulating the tiles of $\tau_{f(n)}$ match if and only if the tiles they represent match.

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

Since the index of any tile set is enumerated infinitely many times by f, if a tile set τ is periodic, there exist infinitely many squares big enough to carry out the computation of M to find a periodic τ-pattern and to stop, and thus, to trigger the beginning of the simulation of the tiles of τ. Therefore, the periodic tile sets are totally simulated by our tile set. □

In this proof we have only used the facts that the property is recursively enumerable and that no tile set that does not tile the plane has the property, which here is the property of 'being periodic'. If a tile set that does not tile the plane has the property, then when we simulate the tiles of this tile set, we arrive at a step where no more tiles can be simulated and the tiling process remains unfinished. We say that a property is *decent* if the class of tile sets satisfying it is recursively enumerable and if this property cannot be verified by a tile set that does not tile the plane. This is equivalent to having a recursively enumerable property such that if a tile set can generate a pattern having this property, then it tiles the plane. Thus, we have the following generalisation.

Theorem 5.2. Let P be a decent property. Then the set $L_P = \{i \mid \tau_i \text{ satisfies } P\}$ can be totally simulated by a single tile set.

This is equivalent to saying that there exists a tile set that is totally universal for L_P. We have the following corollary.

Corollary 5.3. The class of tile sets that generate a period using all of its tiles can be totally simulated by one tile set.

5.3. *Total universality*

The construction of a tile set that is totally universal for periodic tile sets raises the question of universality with respect to total simulation. Is it possible to obtain a tile set that totally simulates any tile set that tiles the plane? Since total simulation preserves some properties, such a universal tile set must have a non-negligible number of properties, such as non-recursivity, maximum kolmogorov complexity, self-simulation *via* any pattern...

In the next definition, we define the notion of universality with the total simulation. The first definition defines total universality for a set of tilings; the second defines total universality for a tile set.

Definition 5.2. Let τ be a tile set and $\mathscr{T}_\tau$ be the set of τ-tilings. We say that a set $A \subset \mathscr{T}_\tau$ is *totally universal* if for any tile set τ' that tiles the plane:

(1) For any τ'-tiling Q, there exists a τ-tiling $P \in A$ such that $Q \preccurlyeq^R P$.
(2) For any τ-tiling $P \in A$, there exists a τ'-tiling Q such that $Q \preccurlyeq^R P$.

If $A = \mathscr{T}_\tau$, we say that τ is totally universal.

In the next theorem, we prove that totally universal tile sets do not exist since the existence of a totally universal tile set would imply a bound on the quasiperiodicity functions that can be reached.

Theorem 5.4. Totally universal tile sets do not exist.

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

Proof. Let τ be a totally universal tile set. Let τ' be a tile set that tiles the plane. Then there exist two integers a, b and a reduction function R from the $a \times b$ patterns of τ to the tiles of τ' such that $\tau' \leqslant_t^R \tau$. Let G_τ be the quasiperiodicity function of τ and P be a τ-tiling with quasiperiodicity function G_τ. Since $\tau' \leqslant_t \tau$, there exists a τ'-tiling Q such that $Q \leqslant^R P$. Any tile of Q is represented by a set of $a \times b$-patterns in P and any square pattern A of size n of Q is represented by a set of $na \times nb$-patterns in P. Since P is quasiperiodic, any of these patterns appears at least once in any square of size $G_\tau(max\{a, b\} \times n)$ in P. Such a square of P represents a rectangle of Q of size $G_\tau(max\{a, b\} \times n)/a \times G_\tau(max\{a, b\} \times n)/b$. Without loss of generality, we can suppose that $a = b$, so any square pattern A of size n of Q appears at least once in any square of size $G_\tau(a \times n)/a$. This means we have the following bound for the quasiperiodicity function $G_{\tau'}$ of τ':

$$G_{\tau'}(n) < G_\tau(a \times n)/a.$$

Since Cervelle and Durand (2004) showed that we cannot bound the quasiperiodicity functions, τ cannot simulate any tile set, so τ cannot be totally universal. □

Since totally universal tile sets do not exist, we would like to find a tile set that is close to universality and is more than complete. The idea in the following sections is to build a tile set that is almost totally universal, that is, totally universal up to a meagre subset of its tilings. To define such a notion of meagreness, we will consider tilings from a topological point of view and introduce Banach–Mazur games on tilings.

6. Topological study of tilings

In this section we are interested in the topological spaces on tilings. Two natural metrics can be used on tilings to obtain such spaces. We will then use them to define topological Banach–Mazur games, which are one of the main tools we will use in Section 7 to prove the existence of tile sets close to total universality.

6.1. *Generalisation of tilings*

In this section we will use the following terminology:

— An *initial set* S is an initial subset $\{1, \ldots, n\}$ of $\mathbb{N}$.
— The process of *mapping* consists of placing the numbers of S on the grid $\mathbb{Z}^2$.
— A mapping A generated by S is called an S-*mapping* and is associated with a mapping function $f_A \in S^{\mathbb{Z}^2}$ that gives the tile of S at position (x, y) in A.
— We say $\mathfrak{M}$ is the set of all mappings, that is, $\mathfrak{M} \equiv \{\{1, \ldots, n\}^{\mathbb{Z}^2}\}_{n \geqslant 1}$.
— An S-*pattern* is an S-mapping defined on a finite subset of $\mathbb{Z}^2$.
— An S-mapping $A \in \mathfrak{M}$ is a Wang tiling if there exists a Wang tile set τ, a τ-tiling P and a bijective function $h : S \to \tau$ such that $h \circ f_A(x, y) = f_P(x, y)$.
— We define $\mathfrak{T}$ as the subset of mappings of $\mathfrak{M}$ that are Wang tilings.

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

6.2. *A Besicovitch topology*

The first metric we introduce is a metric *à la* Besicovitch for cellular automata. An $n \times n'$ τ-pattern m can be viewed as a sequence of numbers placed in a rectangle of size $n \times n'$; we have the number k in position (x, y) if $f_m(x, y) = t_k$ where t_k is the k^{th} tile of τ. Then, intuitively, any reordering of the tiles of τ gives the same pattern. Thus, we would like to say that the distance between two patterns m and m' is c if the number of tiles differing between m and m' as a proportion of the total is at most c up to a reordering of the tiles. We formalise this notion in the following definitions by defining a metric for patterns.

Definition 6.1. Let S and S' be two initial subsets of $\mathbb{N}$ such that $|S| \leqslant |S'|$. Let A be an $n \times n$ S-pattern, B be an $n \times n$ S'-pattern, and g be a one-to-one function from S to S'. We define the metric related to g by

$$\delta_g(A, B) = \frac{\#\{(x, y) \mid g \circ f_A(x, y) \neq f_B(x, y)\}}{n^2}.$$

If A is an S-pattern and B is an S'-pattern such that $|S| \geqslant |S'|$, then $\delta_g(A, B)$ is defined as $\delta_g(B, A)$.

We define the absolute metric by $\delta(A, B) = \min_{g \in S'^S}\{\delta_g(A, B)\}$.

Note that this definition of distance between patterns means δ is symmetric and satisfies the triangle inequality.

Now that we have defined a metric between patterns, we can generalise it to tilings of the whole plane, since, by compactness, a tiling can be seen as the infinite union of patterns of ever increasing sizes.

Definition 6.2. Let A be an S-tiling and let B be an S'-tiling. For any function $g \in S'^S$, we define the tiling metric d_g by:

$$d_g(A, B) = \limsup_{i \to \infty} \delta_g(A_i, B_i),$$

where the A_i (*respectively,* B_i) are the $n \times n$ S-patterns (*respectively,* S'-patterns) centred around $(0, 0)$ in A (*respectively,* B).

We define the absolute tiling metric d by $d(A, B) = \min_{g \in S'^S}\{d_g(A, B)\}$.

Since δ satisfies the triangle inequality, and since reflexivity and symmetry are obvious, d is a pseudometric on $\mathfrak{M}$. The natural way to obtain a metric on $\mathfrak{M}$ is to introduce an equivalence relation $\equiv_d$ defined by $A \equiv_d B \Leftrightarrow d(A, B) = 0$. We can now consider the quotient space $\mathfrak{M}/\equiv_d$ where a typical element is the equivalence class $[A] = \{B \mid d(A, B) = 0\}$. By adding the metric d to this space, we obtain a metric space, which we call $\mathfrak{M}_B$. In this paper, we use $[A]$ to denote the equivalence class of the particular mapping A; an element of $\mathfrak{M}_B$ is denoted by a capital letter in bold, for example, $\mathbf{A}$; and we say 'let $A \in \mathfrak{M}_B$' to mean consider a mapping of the equivalence class of A, that is, a mapping in $[A]$. Similarly, we define the space $\mathfrak{T}/_{\equiv_d}$ where a typical element is $[P] = \{Q \mid d(P, Q) = 0\}$. By adding the metric d to this space, we obtain the metric space $\mathfrak{T}_B$. Of course, $\mathfrak{T}_B \subset \mathfrak{M}_B$. An element of $\mathfrak{T}_B$ is an equivalence class of $\mathfrak{M}_B$ that can contain mappings that are not Wang tilings, but which work 'almost' like Wang tilings since the local constraint is respected almost everywhere.

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

From this definition, we have the following results:

— for any two mappings $A, B \in \mathfrak{M}$, the distance between A and B is in $[0,1[$; and
— for any mapping $C \in \mathfrak{M}$ (*respectively,* any tiling $P \in \mathfrak{T}$) and any $\epsilon \in [0,1[$, there exists a tiling D (*respectively,* Q) such that $d(C,D) \geqslant 1-\epsilon$ (*respectively,* $d(P,Q) \geqslant 1-\epsilon$).

Therefore, for any tiling A, we can build a tiling B such that the distance between A and B is almost one.

To obtain a topological space, since $\mathfrak{M}_B$ is a metric space, we use the natural topology induced by the metric where the open sets are the balls $B(A,r)$. We use the subset topology on $\mathfrak{T}_B$.

The Besicovitch metric is one of two traditional metrics used for tilings. The other is the Cantor metric, which places greater emphasis on the finite patterns centred around the origin.

6.3. *A Cantor metric*

In this section we define and study another *traditional* metric adapted for tilings: a metric *à la* Cantor. The metric discussed in the previous section allows us to investigate the behaviour of a tiling in the whole $\mathbb{Z}^2$ grid. The distance between two tilings is small if their behaviour is close. Another way to measure distance between tilings is to consider the greatest common pattern centred around $(0,0)$ that they share. We first define the function $p : \mathbb{N} \to \mathbb{Z}^2$ such that $p(0) = (0,0)$, $p(1) = (0,1)$, $p(2) = (1,1)$, $p(3) = (1,0)\ldots$, with p continuing to spiral outwards. This function allows us to enumerate the tiles of a given tiling. We define the *prefix-patterns* of a tiling P as follows.

Definition 6.3. Let P be a tiling. The *prefix-pattern* of size n of P is the pattern m defined on the finite subset $D = \{p(0),\ldots,p(n)\} \subsetneq \mathbb{Z}^2$ with the pattern function $f_m(x,y) = f_P(x,y)$ if $(x,y) \in D$. We denote the prefix-pattern m by a finite sequence of tiles ordered by the function p:

$$m = \{f_m \circ p(0), f_m \circ p(1), \ldots, f_m \circ p(n-1)\}.$$

If m is a prefix-pattern of size n, then $m.t_1$, the concatenation of m and the tile t_1, is the prefix-pattern of size $n+1$ such that

$$m.t_1 = \{f_m \circ p(0), f_m \circ p(1), \ldots, f_m \circ p(n-1), t_1\}.$$

The set of prefix-patterns can be enumerated by a tree T. The rules for constructing T are as follows:

— at level 0 in T, we have the empty prefix-pattern;
— at level 1, we have a unique prefix-pattern $\{1\}$;
— and so on.

So a pattern m at level i, composed of j different tiles, has $j+1$ sons, which are $m.1, m.2, \ldots, m.(j+1)$.

One can see that for any tile set S and any prefix-pattern $m = \{m_1,\ldots,m_n\}$ generated by S, there exists a unique bijective function $e_m : S \to \{1,\ldots,|S|\}$ such that the prefix-pattern $e_m(m) \stackrel{\text{def}}{=} \{e_m(m_1), e_m(m_2), \ldots, e_m(m_n)\}$ is an element of T. We say $e_m(m)$ is the

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

canonical form of m. In T, an infinite branch corresponds to a mapping of the plane. Thus, to any S-mapping A there exists a unique bijective function e_A such that the set $e_A(A) \stackrel{\text{def}}{=} \{e_A \circ f_A \circ p(0), e_A \circ f_A \circ p(1), \ldots\}$ corresponds to an infinite branch of T. We say $e_A(A)$ is the *canonical form* of A. We also say that m is a prefix-pattern of A if $e_m(m)$ is a prefix-pattern of $e_A(A)$. We can now define a metric *à la* Cantor.

Definition 6.4. Let A be an S-mapping and B be an S'-mapping. We define the Cantor metric δ_C by $\delta_C(A, B) = 2^{-i}$ where i is the size of the greatest common prefix-pattern of $e_A(A)$ and $e_B(B)$, that is, the highest level in T where $e_A(A)$ and $e_B(B)$ are equal.

If $e_A(A) = e_B(B)$, then $\delta_C(A, B) = 0$.

We can see that δ_C is a pseudometric on $\mathfrak{M}$. In fact, d_C is a hypermetric, that is, a metric such that for any three mappings A, B, C, $d_C(A, C) \leqslant \max\{d_C(A, B), d_C(B, C)\}$. This is a stronger version of the triangle inequality. Note that in a hypermetric space, any point of an open ball is the centre of the ball.

To obtain a metric on $\mathfrak{M}$, we say that two tilings P and Q are equivalent, $P \equiv_C Q$, if their Cantor distance is null. Thus, two tilings are equivalent if they represent the same tiling up to a colour permutation. We use $\mathfrak{M}_C$ to denote the space of equivalence classes $\mathfrak{M}/_{\equiv_C}$ equipped with the metric δ_C.

We define $\mathfrak{T}/_{\equiv_C}$ similarly, and denote it by $\mathfrak{T}_C$. The metric δ_C is a metric on $\mathfrak{T}_C$. From this, we can define a topology on $\mathfrak{T}_C$: we say that the set

$$U_m = \{P \mid m \text{ is a prefix-pattern of } P\}$$

is a clopen set for any prefix-pattern m. This topology gives rise to a different understanding of the topology of tilings compared with the topological space $\mathfrak{M}_C$ since it gives more importance to the local structure centred around $\{0, 0\}$. Since there is a finite set of patterns of a given size, we can cover $\mathfrak{M}_C$ and $\mathfrak{T}_C$ with a finite set of open sets. Therefore, $\mathfrak{M}_C$ and $\mathfrak{T}_C$ are precompact, that is, for all r, there exists a finite set of open balls of radius r that cover these spaces.

Since we have a Hausdorff space, because $\mathfrak{M}_C$ is a metric space, and since we have a basis of clopen sets, $\mathfrak{M}_C$ is a 0-dimensional space.

To complete the definition of the Cantor space, we will just state some obvious facts about the Cantor metric. If A and B are two mappings, then $d_C(A, B) \in [0, 1/2]$, and for any mapping A, there exists a mapping B such that $d_C(A, B) = 1/2$.

6.4. *Basic properties*

The space $\mathfrak{M}_C$ is well defined since two mappings at distance 0 are in fact the same tiling up to a reordering of the tiles, or, in other words, they have the same canonical form. The space $\mathfrak{M}_B$ is slightly different, since two tilings at distance 0 can be different. We have to redefine properties for the mappings of $\mathfrak{M}_B$: if a tiling class contains a tiling with a certain property, then the whole class has this property, since, in fact, all other tilings of the class 'almost' have the property. Thus, we can define the following tiling classes – if

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

$\mathbf{P} \in \mathfrak{T}_B$ is a tiling class, we say that $\mathbf{P}$ is:

— *periodic* if $\mathbf{P}$ contains a periodic tiling;
— *quasiperiodic* if $\mathbf{P}$ contains a quasiperiodic tiling;
— *finite* if $\mathbf{P}$ contains a finite tiling (that is, a pattern);
— *universal* if $\mathbf{P}$ contains a universal tiling and a tiling with a quasiperiodicity function f if $\mathbf{P}$ contains a tiling with a quasiperiodicity function f and if $\mathbf{P}$ does not contain a tiling with a quasiperiodicity function $g < f$.

We will now consider a space that works almost like the space of all Wang tilings. We can see how the different classes work. The following proposition states that the distance between two periodic tilings of $\mathfrak{T}_B$ is a rational number and gives us a characterisation of the classes of periodic tilings.

Proposition 6.1. If P and Q are two different periodic tilings, then there exist $n, m \in \mathbb{N}^*$ such that $d(P, Q) = n/m$. Therefore, if $\mathbf{P} \in \mathfrak{M}_B$ is periodic, it contains one and only one periodic tiling up to a reordering of the tiles.

Proof. Let P be a periodic tiling of period (a, b) and let Q be a periodic tiling of period (c, d). The tiles of P that differ from Q appear in a periodic way, and this period is $\mathrm{lcm}(a, c) \times \mathrm{lcm}(b, d)$. Thus, in any rectangle of size $\mathrm{lcm}(a, c) \times \mathrm{lcm}(b, d)$, there will be a certain number n of different tiles. So,

$$d(P, Q) = \frac{n}{\mathrm{lcm}(a, c) \times \mathrm{lcm}(b, d)}.$$

□

The following proposition shows that two quasiperiodic tilings belonging to the same equivalence class have the same quasiperiodicity function.

Proposition 6.2. If P and Q are two quasiperiodic tilings such that $P \equiv_d Q$, then $g_P = g_Q$, where g_P and g_Q are the quasiperiodicity functions of P and Q.

Proof. Let P and Q be two quasiperiodic tilings such that $P \equiv_d Q$. Suppose $g_P \neq g_Q$. So there exists a number x such that $g_P(x) \neq g_Q(x)$. This means that any pattern of size x appears in any square of size $g_P(x)$ in P and in any square of size $g_Q(x)$ in Q. Thus, there exist patterns of Q that have no equivalent in P, and since those patterns appear in a regular way in Q, the distance between P and Q is strictly greater than 0. □

We will now recall a basic notion of tilings, that of *extraction*. Consider an infinite set of τ-patterns $\{m_1, m_2, \ldots\}$ of ever increasing sizes. We can see them as an infinite tree with the root representing the empty pattern and where a pattern m is a direct son of a pattern n if n is a subpattern of m, and if there does not exist a pattern $m' \neq m$ such that n is a subpattern of m' and m' is a subpattern of m. Thus, we obtain an infinite tree with finite degree. Therefore, by Koenig's lemma, we have at least one infinite branch. In our tree, this branch represents a tiling Q of the plane. This tiling Q is said to be an extraction of the set $\{m_1, m_2, \ldots\}$. Now, we say that Q is extracted from P if there exists an infinite set $\{m_1, m_2, \ldots\}$ of patterns of P such that Q is an extraction of $\{m_1, m_2, \ldots\}$.

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

We will now consider the different distances that we can obtain between Wang tilings and mappings in our two spaces $\mathfrak{M}_B$ and $\mathfrak{T}_B$.

Theorem 6.3. We have the following properties:

(1) There exists a mapping $A \in \mathfrak{M}_B \setminus \mathfrak{T}_B$ and $\epsilon > 0$ such that the ball $B(A, \epsilon)$ does not contain any Wang tilings.
(2) For any n, there exists an infinite subset H of $\mathfrak{T}_B$ such that for any two tilings P and Q of H, $d(P, Q) \geqslant 1 - 1/n^2$.

Proof.

(1) We can construct a pattern composed of six tiles such that its construction implies that this pattern cannot be represented by Wang tiles (at least one of its tiles cannot be represented by a Wang tile). We then consider the periodic tiling A of the plane that has this pattern as a period. So $1/6$ of the tiles cannot be represented by Wang tiles, and thus, for any Wang tiling P, we have $d(A, P) \geqslant 1/6$.
(2) We construct H by iteration. The first step is to build the first two tilings P_0 and P_1 of our iteration. P_0 is the τ_0-unicolour-tiling, that is, P_0 is tiled with a unique unicolour tile. P_1 is the deterministic periodic τ_1-tiling of period $n \times n$ – we use the phrase 'deterministic periodic tiling' to mean τ_1 is composed of n^2 tiles that can be assembled in a unique way to form a periodic $n \times n$ pattern that can be repeated indefinitely to tile the plane. In any square of size $n \times n$, P_0 and P_1 can have at most one tile similar. Thus, $d(P_1, P_2) = 1 - 1/n^2$.
Suppose we have built the first $i-1$ tilings. We build the deterministic periodic τ_i-tiling P_i, where τ_i is a tile set composed of n^{2i} tiles that can be assembled in a unique way to form a periodic $n^i \times n^i$ pattern that can be repeated indefinitely to tile the plane. In any square of size $n^i \times n^i$, P_i can have at most $n^{i-1} \times n^{i-1}$ tiles similar to P_{i-1}, and $n^{i-2} \times n^{i-2}$ tiles similar to $P_{i-2}, \ldots,$ and 1 tile similar to P_0.
Thus,

$$\begin{aligned} d(P_i, P_{i-1}) &= 1 - (n^{i-1} \times n^{i-1})/(n^i \times n^i) \\ &= 1 - 1/n^2 \\ d(P_i, P_{i-2}) &= 1 - 1/n^4 \\ &\vdots \\ d(P_i, P_0) &= 1 - 1/n^{2i}. \end{aligned}$$

□

As a corollary, we have that the spaces $\mathfrak{M}_B$ and $\mathfrak{T}_B$ are not precompact. Note that there exist prefix-patterns that cannot be represented by the local constraint of a Wang tile set. From this, we obtain the following proposition.

Proposition 6.4. There exists an open set in $\mathfrak{M}_C$ that does not contain any Wang tiling.

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

6.5. *Properties of our topologies*

6.5.1. *Properties of the metric spaces.* First, since we have metrisable spaces, our spaces are completely Hausdorff and there are no isolated points.

Proposition 6.5. $\mathfrak{M}_B$, $\mathfrak{T}_B$, $\mathfrak{M}_C$ and $\mathfrak{T}_C$ are all perfectly normal Hausdorff and perfect.

The set of tilings is uncountable. But there exist tile sets that generate an uncountable set of tilings but only a countable set of equivalence classes in $\mathfrak{M}_B$. From this, and the fact that any equivalence class contains an uncountable set of mappings, the question of the cardinality of $\mathfrak{M}_B$ and $\mathfrak{T}_B$ arises. The following proposition shows that even the equivalence classes of $\mathfrak{T}_B$ are uncountable.

Proposition 6.6. $\mathfrak{T}_B$ has the cardinality of the continuum.

To prove this result, we show the following lemma, which is almost equivalent to our proposition.

Lemma 6.7. Let P_0 be the unicolour tiling. The following function is continuous and onto:

$$\begin{aligned} g : \mathfrak{T}_B &\longrightarrow [0,1[\\ A &\longmapsto d(P_0, A). \end{aligned}$$

Proof. To prove the continuity of the function, we use the classical $\epsilon - \delta_\epsilon$ definition. Let $\epsilon > 0$ and choose $\delta_\epsilon < \epsilon$. Let P_x be a tiling of $\mathfrak{T}_B$ and P_y be a tiling in $B(P_x, \delta_\epsilon)$. Such a P_y always exists for all P_x because of Proposition 6.5. If we consider the set $\mathbf{R}$ with the classical metric, we have

$$|g(P_y) - g(P_x)| = |d(P_y, P_0) - d(P_x, P_0)|.$$

Without loss of generality, we can assume that $d(P_y, P_0) < d(P_x, P_0)$, so we have the inequality

$$\begin{aligned} |d(P_y, P_0) - d(P_x, P_0)| &= d(P_y, P_0) - d(P_x, P_0) \\ &\leqslant d(P_y, P_x) + d(P_x, P_0) - d(P_x, P_0) \\ &= d(P_y, P_x) \\ &= \epsilon \leqslant \delta_\epsilon. \end{aligned}$$

To show it is an onto function, we have to show that for any $x \in [0,1[$, there exists a tiling P_x such that $d(P_x, P_0) = x$. For any $x \in [0,1[$, there exists an infinite sequence $s \in \mathbb{N}^{\mathbb{N}}$ such that $s(i+1) - s(i) \in \{0,1\}$ and $\lim_{i\to\infty} s(i)/i = x$. Let n be the smallest natural number such that $x \leqslant 1 - 1/n$. Let τ be the Wang tile set composed of a unicolour tile and $2n$ tiles that can be assembled in a unique way to form a vertical pattern of size $1 \times 2n$ with a unicolour border.

We now build our τ-tiling P_x by iteration . The first column, that is, all the tiles that have abscissa 0, is filled with the unicolour tile if $s(0) = 0$, and with the $1 \times 2n$ pattern if $s(0) = 1$. Then, by iteration, the i^{th} column, that is, all the tiles that have abscissa $(-1)^i \times \lfloor (i+1)/2 \rfloor$, is filled with the unicolour tile if $s(i) - s(i-1) = 0$, and with the $1 \times 2n$ pattern if $s(i) - s(i-1) = 1$.

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

We can now prove that $d(P_x, P_o) = x$.

$$d(P_x, P_o) = d_g(P_x, P_o) = \limsup_{i\to\infty} \delta_g(A_i, B_i)$$

Where the A_i (*respectively,* B_i) are the $i \times i$-patterns of P_x (*respectively,* P_0) centred around $(0,0)$. Since $x < 1 - 1/n$, we have that in each $i \times i$-pattern there will be at least i^2/n unicolour tiles and at most $i^2 \times (n-1)/n$ of the $2n$ other tiles of τ. Thus, any other tile of τ will be represented at most $(i^2 \times (n-1)/(n \times 2n)) + 1$ times. This is less than i^2/n, so $\delta(A_i, B_i) = \delta_g(A_i, B_i)$ where g is defined by

$$\begin{aligned} g : \text{unicolour tile} &\longrightarrow \tau \\ \text{unicolour tile} &\longmapsto \text{unicolour tile.} \end{aligned}$$

We now have

$$\begin{aligned} \limsup_{i\to\infty} \delta_g(A_i, B_i) &= \limsup_{i\to\infty} \frac{\text{number of } \textit{unicolour tile} \text{ in } A_i}{i^2} \\ &= \limsup_{i\to\infty} \frac{s(i)\times i}{i^2} \\ &= \limsup_{i\to\infty} \frac{s(i)}{i} \\ &= x \end{aligned}$$

□

The next theorem is important for understanding Wang tilings. The metric used is strongly related to the information contained in the tiling. Thus, two tilings are close if the information contained in them is similar. So Theorem 6.8 can be stated as follows: a countable set of Wang tilings cannot approach all the information that Wang tilings can generate. Then, by generalising this theorem, we can obtain some nice corollaries.

Theorem 6.8. There does not exist a countable set of Wang tilings that is dense in $\mathfrak{T}_B$.

Corollary 6.9.

(1) For any countable set of Wang tilings H, there exists a tiling P and a natural number n such that $d(P, Q) \geqslant 1/n$ for any $Q \in H$.
(2) There exists a Wang tiling P and an integer n such that for any tile set τ, there exists at least one τ-tiling Q such that $d(P, Q) \geqslant 1/n$.

The next proposition shows how different the two topological spaces $\mathfrak{T}_B$ and $\mathfrak{T}_C$ are. This comes from the fact that the first takes a glimpse at the whole tiling while the second just looks at it with blinkers.

Proposition 6.10. There exists a countable set of Wang tilings that is dense in $\mathfrak{T}_C$.

Proof. Consider the different possible prefix-patterns that can be made with tile sets. By studying the tree T of all patterns, we can see that we have only one prefix-pattern of size one. We call the set that contains this prefix-pattern M_1. Now a prefix-pattern of size i contains at most i different tiles. So there are at most i^i different patterns. Some of them are the same up to a reordering of the tiles, and some are prefix-patterns that cannot respect a local tiling constraint. We use M_i to denote the subset of prefix-patterns of size i that represent a valid Wang pattern, that is, such that a local constraint is respected.

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

Let $M = \bigcup_{i \geqslant 1} M_i$ be the set of all possible prefix-patterns. M is countable, so we can enumerate its elements: $m_1, m_2, \ldots$. To each m_i, we associate a tiling P_i such that m_i is a prefix-pattern of P_i. Then the countable set $D = \{P_i\}_{i \geqslant 1}$ is dense in $\mathfrak{T}_C$.

This is true since any tiling Q can be seen as the limit of a sequence of prefix-patterns of ever-increasing size. Therefore, Q is the limit of a sequence of elements of D. □

From this, we have that $\mathfrak{T}_C$ is separable, and since it is completely metrisable, we have that $\mathfrak{T}_C$ is a Polish space (the same result applies to $\mathfrak{M}_C$). The next theorem shows that for any two tilings $A, B \in \mathfrak{T}_B$, there exists a continuous path $c : [0,1] \to \mathfrak{T}_B$ such that $c(0) = A$ and $c(1) = B$.

Theorem 6.11. $\mathfrak{T}_B$ is a path-connected space.

Proof. We use almost the same proof as in the previous lemma. For any tiling P, we build a continuous and bijective function $i : [0,1] \to \mathfrak{T}_B$ such that $i(0) = P_0$ and $i(1) = P$ where P_0 is the unicolour tiling.

For any $x \in [0,1]$ we build the tiling $P_x = i(x)$. The real x can be written as a sequence of power 2: $x = a_1 2^{-1} + a_2 2^{-2} + \ldots$. Let $A_x = \{a_1, a_2, \ldots\}$. We build P_x by iteration. We begin by explaining how to fill a column of P_x. If we fill the j^{th} column of P_x with tiles of P, that means that $f_{P_x}|_{\{j \times \mathbb{N}\}} = f_P|_{\{j \times \mathbb{N}\}}$ (which is the restriction of f_P to the domain $\{j \times \mathbb{N}\}$); if we do not fill the j^{th} column of P_x with tiles of P, we fill it with:

(1) unicolour tiles if the $j - 1^{th}$ and the $j + 1^{th}$ columns are also tiled with unicolour tiles;
(2) oherwise, special matching tiles that allow us to make the transition colours between the tiles of the $j - 1^{th}$ and the $j + 1^{th}$ columns.

The first step is to consider the columns with abscissas -1 and 1. If $a_1 = 1$, we fill the column -1 with tiles of P. At step i, we consider the columns with abscissas

$$\{-2^{i-1}, -2^{i-1} + 1, \ldots, -1, 1, \ldots, 2^{i-1} - 1, 2^{i-1}\}.$$

By iteration, the columns

$$\{-2^{i-2}, -2^{i-2} + 1, \ldots, -1, 1, \ldots, 2^{i-2} - 1, 2^{i-2}\}$$

have already been filled, and we still have 2^{i-1} columns to fill. We define b_i as follows:

$$b_i = 2^i \times \sum_{j=1}^{i} a_j \times \frac{1}{2^j} - 2^{i-1} \times \sum_{k=1}^{i-1} a_k \times \frac{1}{2^k}.$$

In this equation:

— b_i corresponds to the number of columns we have to fill with tiles of P in order to respect the binary development of x.
— The

$$2^i \times \sum_{j=1}^{i} a_j \times \frac{1}{2^j}$$

corresponds to the number of columns we still need to fill with tiles of P.

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

— The

$$2^{i-1} \times \sum_{k=1}^{i-1} a_k \times \frac{1}{2^k}$$

corresponds to the number of columns already filled with the tiles of P.

Now, for the columns with abscissas

$$\{-2^{i-1}, -2^{i-1}+1, \ldots, -1, 1, \ldots, 2^{i-1}-1, 2^{i-1}\},$$

we fill the first b_i columns free of tiles with the tiles of P.

By iteration, we build a tiling P_x. We have to check that the function i is continuous and bijective, and that $i(0) = P_0$ and $i(1) = P$. By construction, since $0 = \sum_{i \geqslant 1} 0 \times (1/2^i)$, none of the columns of $i(0)$ are filled with tiles of P, so $i(0) = P_0$.

Now, $i(1) = P_1$, and since $1 = \sum_{i \geqslant 1} 1/2^i$, all but one of the columns are tiled with the tiles of P. So, using the identity function id, we have

$$\begin{aligned} d(P_1, P) = d_{id}(P_1, P) &= \limsup_{i \to \infty} \delta_{id}((P_1)_i, (P)_i) \\ &= \lim_{i \to \infty} \tfrac{1}{n} \\ &= 0 \end{aligned}$$

Therefore, $i(1) = [P]$.

We will now prove the continuity of i. Let $x, y \in [0, 1]$. We will use the traditional metric $d_{\mathbf{R}}$ on $\mathbf{R}$. We use $(x)_i$ to denote the first i decimals of x in binary. By construction, we have

$$d((P_x)_i, (P_y)_i) = d_{\mathbf{R}}((x)_i, (y)_i).$$

Indeed, if $(x)_i = a/2^i$ and $(y)_i = b/2^i$, then P_x will have $a - b$ different columns from P_y, so,

$$d(P_x, P_y) = \frac{a-b}{2^i}.$$

So

$$\begin{aligned} d(x, y) &= \lim_{i \to \infty} d_{\mathbf{R}}((x_i), (y)_i) \\ &= \lim_{i \to \infty} d((P_x)_i, (P_y)_i) \\ &= d(P_x, P_y), \end{aligned}$$

which proves the continuity of i. □

6.5.2. *Topological properties.* In this section we consider the topological structure of our spaces. Since they are metric spaces, natural topologies are induced on them by the metrics. We define the following sets:

$$\begin{aligned} \text{Mapping}(S) &= \{[A] \mid A \text{ is an } S\text{-mapping}\} \\ \text{Wang}(S) &= \{[P] \mid P \text{ is a Wang } S\text{-tiling}\} \end{aligned}$$

And for any tile set τ, we define the set $Wang(\tau) = \{[P] \mid P \text{ is a } \tau\text{-tiling}\}$.

The following theorem shows that the set of mappings or tilings generated by a tile set is a closed set. Then we give a characterisation of the set of Wang tilings that can produce a tile set.

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

Theorem 6.12. Let S and τ be two tile sets. Then, in $\mathfrak{M}_B$, Wang(τ) and Mapping(S) are closed sets and Wang(τ) is either a closed discrete set or a closed non-discrete nowhere-dense set.

Proof.

(1) To prove that Wang(τ) is a closed set, we prove that any infinite sequence in Wang(τ) has its limit in Wang(τ). Let $s \in \text{Wang}(\tau)^{\mathbb{N}}$ and let $[P]$ be the limit of this sequence. Let

$$Q = \{P_1, P_2, \dots \mid \exists i,\ s(i) = [P_i] \text{ and } P_i \text{ is a } \tau\text{-tiling}\}.$$

Q corresponds to the set of Wang tilings, while $\{s(i)\}_{i \geqslant 0}$ corresponds to the set of classes of Wang tilings. We are going to build a tiling P_∞ such that

$$\lim_{n \to \infty} d(P_\infty, P_i) = 0.$$

We define the set Q_i by

$$Q_i = \{P_{1,i}, P_{2,i}, \dots \mid\ P_{j,i} \text{ is the } i \times i \text{ pattern of } P_j \text{ centred around } (0,0)\}.$$

Q_i is infinite, but since τ is finite, the number of different patterns of size i is finite. Thus, at least one $P_{j,i}$ appears infinitely many times. We define the $i \times i$ pattern centred around $(0,0)$ in P_∞ as $P_{j,i}$. We build Q_{i+1} by iteration as follows:

$$Q_{i+1} = \{P_{1,i+1}, P_{2,i+1}, \dots \mid\ P_{j,i} \text{ is the } i \times i \text{ pattern of } P_j \text{ centred around } (0,0) \text{ and } P_{j,i} \in Q_i\}.$$

Q_{i+1} is infinite, so there exists at least one pattern $P_{j,i+1}$ that appears infinitely many times. We define the $(i+1) \times (i+1)$ pattern centred around $(0,0)$ in P_∞ as $P_{j,i+1}$. By iteration, we build a tiling P_∞ such that:

(1) $\forall\, \epsilon > 0$, there exists an infinite subset $I \subset \mathbb{N}$ such that $d(P_\infty, s(i)) < \epsilon\, \forall\, i \in I$).

(2) P_∞ is a τ-tiling.

Therefore, P_∞ is the limit of the sequence s and $P_\infty \in [P]$. Hence, $[P] \in Wang(\text{S})$.

(2) We show that a tiling that is not in Mapping(S) cannot be the limit of a sequence of S-mappings. Let S' be a tile set such that $|S'| \geqslant |S|$. Indeed, if $|S'| \leqslant |S|$, then Mapping(S') $\subset$ Mapping(S). Let Q be an S'-mapping that uses all the tiles S' such that $d(P, Q) > 0$ where P is the S-tiling that is built by replacing all the tiles of $S' \setminus S$ in Q by tiles of S.

This condition guarantees that $Q \notin [P]$. We just have to note now that if $d(P, Q) > \epsilon$ for $\epsilon > 0$, there can be no S-tiling P' such that $d(P', Q) < \epsilon$. Therefore, no tiling outside tiling (S) can be a limit of a sequence of S.

(3) From the two previous points, it is obvious that Wang(τ) is either a closed discrete set, or a closed non-discrete nowhere-dense set. □

Corollary 6.13.

(1) $\mathfrak{T}_B$ is meagre in $\mathfrak{M}_B$.

(2) $\mathfrak{M}_B \setminus \mathfrak{T}_B$ is dense in $\mathfrak{M}_B$.

We will now show that our spaces are Baire spaces.

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

Theorem 6.14. $\mathfrak{T}_B$, $\mathfrak{T}_C$, $\mathfrak{M}_B$ and $\mathfrak{M}_C$ are complete metric spaces, and thus, are Baire spaces.

Proof.

(1) To prove this result, we show that any Cauchy sequence of $\mathfrak{T}_B$ converges in $\mathfrak{T}_B$. Let c be a Cauchy sequence. Thus, for any ϵ, there exists n_0 such that for any $n, m \geqslant n_0$, we have $d(c(n), c(m)) = d(P_n, P_m) < \epsilon$. The same argument as in the proof of Theorem 6.12.1 can be used here, so $\mathfrak{T}_B$ is a complete metric space.

(2) Let s be a Cauchy sequence of mappings of $\mathfrak{T}_C$. Thus, for any ϵ, there exists an n_0 such that for any $n, m \geqslant n_o$, we have $d_c(s(n), s(m)) \leqslant \epsilon$. For mappings, this can be reformulated as the statement: for any 2^{-i}, there exists an n_0 such that for any $n, m \geqslant n_0$, the mappings $s(n)$ and $s(m)$ have a same prefix-pattern m_i of size i. So we have a set of prefix-patterns $\{m_i\}_{i \geqslant 1}$. Therefore, we can extract from the m_i's a mapping B such that $\lim_{i \to \infty}(d(s(i), B)) = 0$. □

In the next section, we will introduce games on our topological spaces as tools for studying the structure of tilings.

6.6. *Topological games*

Now that we have defined some notions for the 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.

Several kinds of infinite games exist and are used in many different fields. Here we will give the definitions of Banach–Mazur games (Oxtoby 1957), played on the topological structure of a space. Different definitions of these games exist; we propose the following for our purposes.

Definition 6.5. Let X be a topological space and a Y family of subsets of X such that:

(1) Any member of Y has non-empty interior.
(2) Any non-empty open subset of X contains a member of Y.

Let C be a subset of X. The game (X, C) proceeds as follows:

— Player I chooses a subset Y_1 of Y;
— Player II chooses a subset Y_2 of Y such that $Y_2 \subseteq Y_1$;
— then Player I chooses a subset Y_3 of Y such that $Y_3 \subseteq Y_2$;
— and so on

At the end of the infinite game, we obtain a decreasing sequence of sets

$$X \supseteq Y_1 \supseteq Y_2 \supseteq \dots$$

such that Player I has chosen the sets with odd indexes and Player II has chosen the sets with even indexes. Player II wins the game if $\bigcap_{n \geqslant 1} Y_i \subseteq C$.

The main application of Banach–Mazur games is to determine the different subsets of X such that Player II has a winning strategy. Of course, if $C = X$, Player II has a winning strategy. The question is: how big does C have to be to allow Player II to have a winning

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

strategy? This gives rise to a classical theorem for Banach–Mazur games on topological spaces, which states that a subset C of X is meagre, that is, is the the union of countably many nowhere dense subsets, if and only if Player II has a winning strategy for the game on $\{X, X \setminus C\}$. Meagre is thus a topological definition of *small* or *negligible* subsets. We propose the following two Banach–Mazur games on the set of tilings generated by a tile set.

Definition 6.6. Let X be a set of tilings generated by a tile set and C be a subset of X.

(1) The first game $\{X, C\}_B$ is played on the Besicovich 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 and an integer $n_2 > n_1$ such that

$$\mathscr{B}_B(P_2, 1/n_2) \subset \mathscr{B}_B(P_1, 1/n_1),$$

where $\mathscr{B}_B(P, \epsilon)$ is the open ball in Besicovich topology with centre P and radius ϵ;

— and so on.

Player II wins the game if $\bigcap_{n>1} \mathscr{B}_B(P_i, 1/n_i) \in C$.

(2) The second game $\{X, C\}_C$ is played on the Cantor topology and has the following rules:

— Player I chooses a square pattern A_1 centred around the origin;

— Player II chooses a square pattern A_2 that is an extension of A_1, that is, 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 then obtain an infinite tiling P. Player II wins the game if $P \in C$.

In both games, players have to choose an open set that is included in the open set played previously by their opponent. Since these games are played on topological spaces, they are Banach–Mazur games, so Player II has a winning strategy in the game (X, C) if and only if $X \setminus C$ is meagre in X.

We can now define almost total universality.

Definition 6.7. A tile set τ is *almost totally universal* if it contains a totally universal subset A (see Definition 5.2) residual, or in an equivalent way, that $\mathscr{T}_\tau \setminus A$ is meagre.

Therefore, an almost totally universal tile set is a tile set that is totally universal modulo a meagre set: just a small subset prevents it from being totally universal. In the following section, we construct an almost totally universal tile set using the various techniques introduced in the paper.

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

7. Almost totally universal tile set

7.1. *Description of the construction*

In this section we construct an almost totally universal tile set $\mathfrak{V}$. We use three Turing machines M, N and P that we simulate in the synchronised construction explained in Sections 3.3 and 3.4. The three machines work 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. The first machine, M, checks whether the input is well written, and if it is not, M stops. We add another restriction to M: we want the code of $\mathfrak{V}$ to appear as an input. The tile set $\mathfrak{V}$ will have access to its own code – we can prove, using Kleene's recursion theorem, that a tile set can have access to its own code (see Lafitte and Weiss (2008a) and Durand *et al.* (2008)). Let m be the number of tiles of $\mathfrak{V}$. M checks that the input contains two codes of tile sets of m tiles, and checks that one of them is the code of $\mathfrak{V}$. Therefore, the input has to be of the form $i_1\$i_2\$\ldots i_m\$i'_m\$i_{m+1}\$\ldots$, where i'_m is the code of $\mathfrak{V}$.

The second machine, N, checks for any n if any of the tile sets τ_{i_j}'s, with $i_j < n$, 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 final machine, P, is a non-deterministic machine that simulates the tile sets of the input, that is, it generates space×time diagrams isomorphic to the tiles of the tile sets (as we have already done in the previous sections). Therefore, each level of squares simulates a tile set given as input, and any tile set of the input is simulated in a level of squares.

Note that if one of the Turing machine reaches a final state, the tiling cannot be completed. This can be done by representing the final state of a machine by a special tile that cannot be associated with another tile.

To start the simulations of these machines in our tilings, we first ensure that the only computation tile that exists in the level 1 squares is the tile, say t_0, representing the initial state of M, N and P. By synchronisation, this means that any middle tile of the first line of any square of this construction also corresponds to this tile, so 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, 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 earlier. P simulates any of the tile sets i_j, so $\mathfrak{V}$ totally simulates any tile set τ_{i_j}. Since the index of $\mathfrak{V}$ is also given in the input, P also simulates an $\mathfrak{V}$-tiling. In fact, by transitivity of the simulation, it self-simulates infinitely many times. Each time $\mathfrak{V}$ self-simulates, it also simulates a set of tile sets $\{\tau_{i'_j}\}_{j>0}$ since it simulates an $\mathfrak{V}$-tiling that simulates this set. So, an $\mathfrak{V}$-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, an $\mathfrak{V}$-tiling must simulate some tile sets infinitely many times.

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

7.2. *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 7.1. 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 $B = \mathscr{T}_{\mathfrak{Y}} \setminus A$, where $\mathscr{T}_{\mathfrak{Y}}$ is the set of $\mathfrak{Y}$-tilings. A is totally universal. We show that A is residual in $\mathscr{T}_{\mathfrak{Y}}$ in both topologies.

We first show that A is residual in $\mathscr{T}_{\mathfrak{Y}}$ (in the Cantor topology) by showing that Player II has a winning strategy in the game $G(A, \mathscr{T}_{\mathfrak{Y}})_C$. In this game, Player I first chooses an $\mathfrak{Y}$-pattern centred 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, ordered first by the number of their tiles, and then by a lexicographic order of the colours 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 (it is its n^{th} move). Player II wants to force the code of τ_n to appear somewhere in the tiling. If he can write this code, then, by synchronisation, 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 an $\mathfrak{Y}$-tiling that 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 in which 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 that simulates at least one tiling for any tile set in Ω. Therefore, this tile set is in A, and A is residual in $\mathscr{T}_{\mathfrak{Y}}$ with the Cantor topology.

We now show that B is meagre in $\mathscr{T}_{\mathfrak{Y}}$, in the Besicovitch topology. The first move of Player I is to play an $\mathfrak{Y}$-tiling P_1 and an integer n_1 to define the open ball $\mathscr{B}_B(P_1, 1/n_1)$. 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, 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 synchronisation, 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

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

a tiling that also simulates $\{\tau_1, \tau_2, \ldots, \tau_{i-1}\}$. Player I chooses a tiling P_i and an integer n_i. We will show that Player I can choose a tiling $Q_i \in \mathscr{B}_B(P_i, 1/n_i)$ that simulates all of $\{\tau_1, \tau_2, \ldots, \tau_{i-1}\} \cup \{\tau_i\}$.

We begin with some remarks. If an $\mathfrak{Y}$-tiling P simulates a tile set τ, 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 in this simulation, and we can bound the proportion of tiles that are concerned with it. Indeed, only those tiles that are in level j squares, and the tiles that 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 in a simulation to be strictly less than 1.

Let S be an $\mathfrak{Y}$-tiling. $\mathfrak{Y}$ self-simulates, therefore S simulates an $\mathfrak{Y}$-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 an $\mathfrak{Y}$-tiling, then S_1 simulates an $\mathfrak{Y}$-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 particular property of having the code of the i^{th} tile set of Ω in its input, so it simulates τ_i. Player II plays this tiling Q_i, which is at distance less than $1/n_i$ from 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 an $\mathfrak{Y}$-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 $\mathscr{T}_{\mathfrak{Y}}$ within the Besicovitch topology. □

Acknowledgements

We woould like to express our warm thanks to Guillaume Theyssier who, in the first place, convinced one of the co-authors of the possible existence of the tile set described here. We are also indebted to Bruno Durand for his pertinent remarks on previous work, which undoubtedly encouraged us to pursue this direction of research.

References

Allauzen, C. and Durand, B. (1997) Appendix A: Tiling problems. In: Börger, E., Grädel, E. and Gurevich, Y. (eds.) *The classical decision problem*, Springer-Verlag 407–420.

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

Cervelle, J. and Durand, B. (2004) Tilings: recursivity and regularity. *Theoretical Computer Science* **310** (1-3) 469–477.

Delorme, M., Mazoyer, J., Ollinger, N. and Theyssier, G. (2010) Bulking I: an abstract theory of bulking. Available at `http://hal.archives-ouvertes.fr/hal-00451732/fr/`.

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press

Durand, B. (1999) Tilings and quasiperiodicity. *Theoretical Computer Science* **221** (1-2) 61–75.

Durand, B., Levin, L. A. and Shen, A. (2001) Complex tilings. In: *Proceedings of the thirty-third annual ACM Symposium on Theory of Computation (STOC)*, ACM 732–739.

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

Durand, B., Romashchenko, A. E. and Shen, A. (2008) Fixed point and aperiodic tilings. In: Ito, M. and Toyama, M. (eds.) Proceedings of the conference Developments in Language Theory. *Springer-Verlag Lecture Notes in Computer Science* **5257** 276–288.

Hanf, W. P. (1974) Nonrecursive tilings of the plane. I. *Journal of Symbolic Logic* **39** (2) 283–285.

Lafitte, G. and Weiss, M. (2007) Universal tilings. In: Thomas, W. and Weil, P. (eds.) Twenty-Fourth International Symposium on Theoretical Aspects of Computer Science (STACS'2007). *Springer-Verlag Lecture Notes in Computer Science* **4393** 367–380.

Lafitte, G. and Weiss, M. (2008a) Computability of tilings. In: Ausiello, G., Karhumäki, J., Mauri, G. and Ong, C.-H. L. (eds.) Proceedings of the Fifth IFIP International Conference on Theoretical Computer Science (IFIP-TCS'2008). *IFIP Conference Proceedings*, Springer-Verlag **273** 187–201.

Lafitte, G. and Weiss, M. (2008b) Simulations between tilings. In: Beckmann, A., Dimitracopoulos, C. and Löwe, B. (eds.) Logic and Theory of Algorithms, Fourth Conference on Computability in Europe (CiE'2008), University of Athens 264–273.

Lafitte, G. and Weiss, M. (2008c) A topological study of tilings. In: Agrawal, M., Du, D., Duan, Z. and Li, A. (eds.), Fifth Annual Conference on Theory and Applications of Models of Computation (TAMC'2008). *Springer-Verlag Lecture Notes in Computer Science* **4978** 375–387.

Lafitte, G. and Weiss, M. (2009) An almost totally universal tile set. In: Chen, J. and Cooper, S. B. (eds.) Sixth Annual Conference on Theory and Applications of Models of Computation (TAMC'2009). *Springer-Verlag Lecture Notes in Computer Science* **5532** 271–280.

Myers, D. (1974) Nonrecursive tilings of the plane. II. *Journal of Symbolic Logic* **39** (2) 286–294.

Oxtoby, J. (1957) The Banach–Mazur game and Banach category theorem. In: Contribution to the theory of games, Volume III. *Annals of Mathematical Studies*, Princeton University Press **39** 159–163.

Robinson, R. M. (1971) Undecidability and nonperiodicity for tilings of the plane. *Inventiones Mathematicae* **12** (3) 177–209.

Wang, H. (1961) Proving theorems by pattern recognition II. *Bell System Tech. Journal* **40** (1) 1–41.

https://doi.org/10.1017/S0960129510000228 Published online by Cambridge University Press