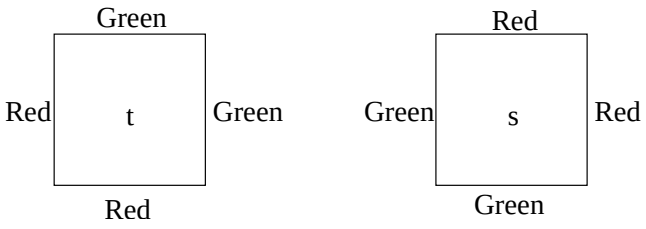
the exception of type $3 \cdot 3 \cdot 3 \cdot 3 \cdot 6$, any isometry that takes vertex $P_1$ and its incident polygons onto $P_2$ and its incident polygons is a symmetry of the tiling. The archimedean tiling of type $3 \cdot 3 \cdot 3 \cdot 3 \cdot 6$ comes in two enantiomorphic forms that are congruent with each other only by odd isometries.

Archimedean tilings are also called uniform, which refers to the fact that they are vertex transitive: the entire tiling looks exactly the same from each vertex. This is a stronger property than the property we started with: that the tiling looks locally the same at each vertex, as each vertex is of the same type.

# 4 Wang tiles

Wang tiles are unit square tiles with colored edges. Hence each tile can be represented as a 4-tuple $(N, E, S, W)$ where $N, E, S$ and $W$ are the colors of the north, east, south and west sides of the square. Tilings with a finite number of prototiles are only considered. In Wang tilings copies of the prototiles are placed at integer lattice points, without rotating or flipping the tiles, so that all tiles are congruent to the given prototiles by translations only. A tiling can then be represented as a function $f : \mathbb{Z}^2 \to \mathcal{P}$ where $\mathcal{P}$ is the set of prototiles and $f(i, j)$ gives the tile at position $(i, j) \in \mathbb{Z}^2$. The tiling rule is that in a valid tiling the shared edge between any two tiles that are edge neighbors must have the same color.

For example, set $\mathcal{P} = \{$(Green, Green, Red, Red), (Red, Red, Green, Green)$\}$ consists of two prototiles



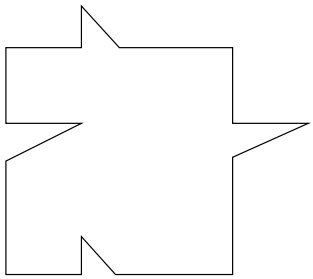

that admit the checkerboard-tiling

| | | | | | | |
|---|---|---|---|---|---|---|
| t | s | t | s | t | s | t |
| s | t | s | t | s | t | s |
| t | s | t | s | t | s | t |
| s | t | s | t | s | t | s |

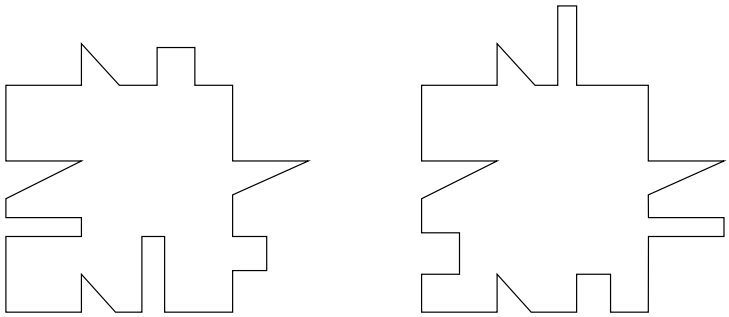
Wang tiles provide a discrete abstraction of tilings that allows us to study tilings using tools of discrete mathematics rather than geometry. This is especially useful when investigating computational properties and problems related to tilings. At first, Wang tiles may seem very restricted as the tilings are on a square lattice only. Nevertheless, the computational problems on Wang tiles are as hard as on more general types of tiles. By using Wang tiles we avoid problems related to representations of tiles (e.g. irrational coordinates of vertices) on computers, and we transform geometric problems into more manageable symbolic problems.

Our first observation is that Wang tiles fit our original definition of tiles as topological disks. We can namely represent Wang tiles as polygons as follows: The basic shape is a unit square. The middle of the north and east sides of each tile contain triangular "bumps" and the south and west sides have

"dents" that exactly fit the bumps. The bump/dent pairs are different in the horizontal and the vertical directions, and they are asymmetric so that flipped and non-flipped tiles do not match:



It should be clear that these tiles can only tile the plane in such a way that all tiles are aligned, and rotations and flips are not possible. To simulate the colors, we introduce an additional bump/dent pair on the sides of the tiles. Each color has its own bump/dent shape that does not fit with any other color. For example, our sample protoset of two tiles could look like this:



It should be obvious from this construction that any tiling by such polygons is congruent to a tiling where the tiles are positions at integer lattice points, without rotations and flips. Such tilings are clearly "isomorphic" to Wang tilings.

In the following we consider all tilings that given prototiles admit. In particular, we are interested to know when do given prototiles admit at least some tiling and when do they admit a periodic tiling. As it turns out that even among Wang tiles these questions can not be algorithmically answered (they are undecidable), so it follows that the questions are undecidable also among tiles that are polygons.

In the following two subsections we prove two preliminary results that will be needed in the algorithmic considerations that follow: First we show that if a finite set of Wang prototiles admits a tiling whose symmetry is a frieze group then it automatically admits also a tiling with a wallpaper symmetry. Then we prove that if one can tile arbitrarily large squares then one can also tile the entire infinite plane.

## 4.1 Periodic tilings

A tiling is called non-periodic if its symmetry group is finite, that is, if there is no translation that keeps the tiling invariant. A tiling is two-way periodic, or simply periodic, if its symmetry group is a wallpaper group, that is, if there are translations in non-parallel directions that keep the tiling invariant. A tiling whose symmetry group contains some non-trivial translation will be called called one-way periodic.

Vector $(a, b) \neq (0, 0)$ is called a period of a tiling, if $\tau_{(a,b)}$ is a symmetry of the tiling. In the case of a Wang tiling $f : \mathbb{Z}^2 \longrightarrow \mathcal{P}$ this means that $f(x, y) = f(x + a, y + b)$ for all $(x, y) \in \mathbb{Z}^2$.

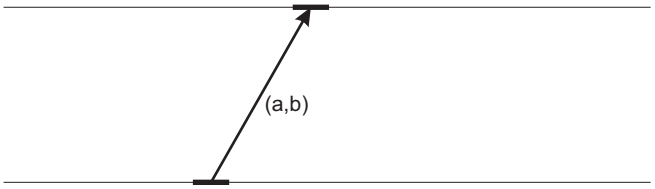
Notice that any two-way periodic tiling with Wang tiles has horizontal and vertical periods of equal lengths. Namely, if $f : \mathbb{Z}^2 \longrightarrow \mathcal{P}$ is periodic with non-parallel periods $(a, b)$ and $(c, d)$ then it is also

periodic with the horizontal period $d(a,b) - b(c,d) = (ad - bc, 0)$ and the vertical period $a(c,d) - c(a,b) = (0, ad - bc)$. Note that $ad - bc \neq 0$ as vectors $(a,b)$ and $(c,d)$ are not parallel. In other words, a two-way periodic Wang tiling consists of a periodic repetition of a square pattern.

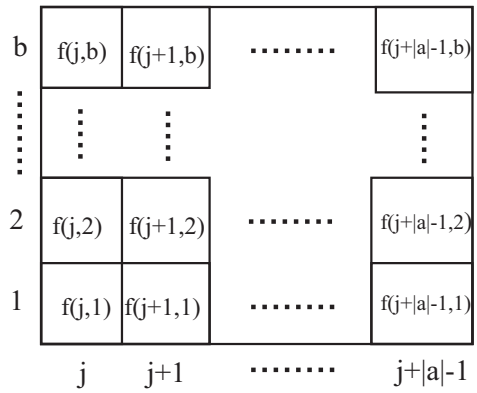
The next theorem states that a set of Wang tiles that admits a one-way periodic tiling also admits a periodic tiling:

**Theorem 4.1** *Let $\mathcal{P}$ be a finite set of Wang prototiles that admits a tiling $f : \mathbb{Z}^2 \longrightarrow \mathcal{P}$ that is one-way periodic. Then there exists also a two-way periodic tiling $g : \mathbb{Z}^2 \longrightarrow \mathcal{P}$.*

*Proof.* Let $(a,b) \neq (0,0)$ be a period of tiling $f$. Without loss of generality we may assume that $b > 0$. Consider a horizontal strip of height $b$ extracted from tiling $f$, e.g., the tiles $f(x,y)$ for $1 \leq y \leq b$. The sequences of horizontal colors on the top and the bottom of this strip are identical, with the horizontal offset $a$:

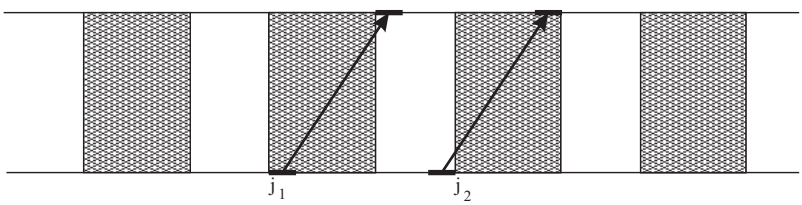



Within this strip, consider the rectangular $|a| \times b$ blocks

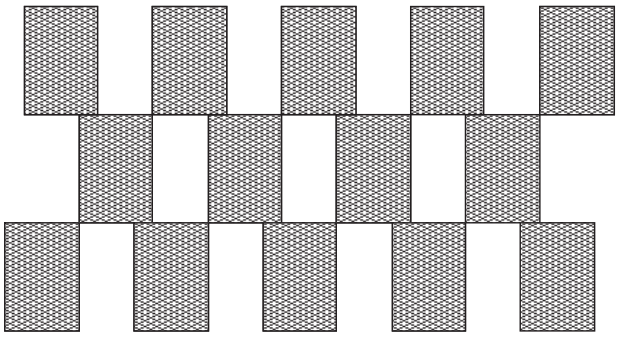



of tiles extracted from $f$ with the bottom-left corner in position $(j,1)$, for all $j \in \mathbb{Z}$. (And if $a = 0$ consider just the sequences of vertical colors on the $b$ rows.) Since there are only a finite number of tiles in the protoset, there are only a finite number of such blocks. This means that for two different values of $j$, say $j_1$ and $j_2$, the blocks are identical.

Now we can construct a valid periodic tiling of an infinite horizontal strip of height $b$ by repeating the pattern between positions $j_1$ and $j_2$. Note that the sequences of horizontal colors on the top and the bottom of this strip are again identical, with the horizontal offset $a$:



The tiling of the strip is valid and it has a horizontal period of length $j_2 - j_1$. A valid, two-way periodic tiling of the plane can now be obtained by stacking copies of the strip on top of each other, with the horizontal offset $a$:



□

## 4.2 Compactness principle

In later chapters we'll introduce a metric on tilings that induces a compact topology. This will imply several interesting results, but for the main algorithmic questions that follow next we just need to know that if a Wang set admits tilings of arbitrarily large squares then it admits a tiling of the whole infinite plane. This is a direct consequence of the compactness, but we state the result here without a direct reference to topology.

Let $\mathcal{P}$ be a finite set of Wang prototiles. Let us call any function

$$c : \mathbb{Z}^2 \to \mathcal{P}$$

a *configuration*, and let us denote by

$$\mathcal{P}^{\mathbb{Z}^2} = \{c : \mathbb{Z}^2 \to \mathcal{P}\}$$

the set of all configurations over the tile set $\mathcal{P}$. Note that configurations are arbitrary assignments of tiles on integer lattice points. i.e., the color constraints are not checked. Valid tilings are particular types of configurations.

Consider an infinite sequence $c_1, c_2, \ldots$ of configurations, each $c_i \in \mathcal{P}^{\mathbb{Z}^2}$. We say that the sequence *converges* and $c \in \mathcal{P}^{\mathbb{Z}^2}$ is its *limit* if for every $(x, y) \in \mathbb{Z}^2$ there exists some $k \geq 0$ such that $c_i(x, y) = c(x, y)$ for all $i \geq k$. In other words: if we look at an arbitrary position and browse through a converging sequence $c_1, c_2, \ldots$ then from some moment on we always see the same tile in that position. It is obvious that if a limit exists it is unique, and we denote this limit by

$$\lim_{i \to \infty} c_i.$$

A *subsequence* of $c_1, c_2, \ldots$ is another sequence $c_{i_1}, c_{i_2}, \ldots$ where $i_1 < i_2 < \ldots$. A subsequence is hence obtained by picking infinitely many elements of the sequence, preserving their relative order. Obviously every subsequence of a converging sequence also converges and has the same limit.

The following theorem states the compactness of the configuration space:

**Theorem 4.2** *Every sequence of configurations has a converging subsequence.*

*Proof.* Let $c_1, c_2, \ldots$ be an arbitrary sequence, $c_i \in \mathcal{P}^{\mathbb{Z}^2}$. Let $\vec{r}_1, \vec{r}_2, \ldots$ be some (arbitrary) enumeration of elements of $\mathbb{Z}^2$. In the following we show that there is a subsequence $c_{i_1}, c_{i_2}, \ldots$ such that for every

$n \geq 1$, if $j \geq n$ then $c_{i_j}(\vec{r}_n) = c_{i_n}(\vec{r}_n)$, i.e., the subsequence has a constant value in the $n$'th position $\vec{r}_n$ starting from the $n$'th element of the subsequence. Then clearly the subsequence converges.

Let us choose indices $i_0 < i_1 < i_2 < i_3 < \ldots$ inductively as follows: $i_0 = 0$ and $i_1 \geq 1$ is the smallest positive index such that there are infinitely many elements in $c_1, c_2, \ldots$ that agree with $c_{i_1}$ in the first position $\vec{r}_1$. Such $c_{i_1}$ exists because there are only finitely many different tiles that can appear in position $\vec{r}_1$.

Suppose then that $i_{k-1}$ has been chosen and we want to choose $i_k$ for $k \geq 2$. We choose $i_k$ to be the smallest integer that satisfies the following three conditions:

$(A_k)$ $i_k > i_{k-1}$,

$(B_k)$ $c_{i_k}(\vec{r}_j) = c_{i_{k-1}}(\vec{r}_j)$ for all $j = 1, 2, \ldots k-1$.

$(C_k)$ There exist infinitely many indices $i$ such that $c_i(\vec{r}_j) = c_{i_k}(\vec{r}_j)$ for all $j = 1, 2, \ldots k$.

Numbers $i_k$ that satisfy $(A_k)$–$(C_k)$ always exist for the following reasons: Because condition $(C_{k-1})$ was satisfied when $i_{k-1}$ was chosen, we have infinitely many choices of $i_k$ that satisfy $(B_k)$. Set $\mathcal{P}^k$ is finite so there is a finite number of combinations of tiles that can appear in positions $\vec{r}_1, \ldots, \vec{r}_k$. Consequently, among the infinitely many indices $i_k$ that satisfy $(B_k)$ there are infinitely many choices that also satisfy $(C_k)$. Some of them hence satisfy all requirements $(A_k)$–$(C_k)$.

It follows from properties $(B_k)$ that $c_{i_1}, c_{i_2}, \ldots$ converges: For an arbitrary $\vec{r}_n \in \mathbb{Z}^2$ all $c_{i_j}$ for $j \geq n$ have the same tile in position $\vec{r}_n$. □

Note: The proof is essentially the same as the proof of weak Kőnig's lemma which states that an infinite binary tree contains an infinite path. The proof did not require the axiom of choice. (The same result could also be easily proved using Tychonoff's theorem, but that is equivalent to the axiom of choice.)

Let us say that a configuration $c : \mathbb{Z}^2 \to \mathcal{P}$ tiles correctly at position $(x, y) \in \mathbb{Z}^2$ if $c(x, y)$ matches in color with its neighbors $c(x, y-1), c(x, y+1), c(x-1, y), c(x+1, y)$. A configuration is then a valid tiling iff it tiles correctly at each position.

The following corollary of the compactness principle states that if $\mathcal{P}$ can be used to properly tile arbitrarily large squares then it admits a valid tiling of the plane:

**Corollary 4.3** *Let $\mathcal{P}$ be a finite set of Wang tiles. Suppose that for each finite set $F \subset \mathbb{Z}^2$ of positions there is a configuration that tiles correctly at each $(x, y) \in F$. Then $\mathcal{P}$ admits a valid tiling.*

*Proof.* Let $\vec{r}_1, \vec{r}_2, \ldots$ be an enumeration of elements of $\mathbb{Z}^2$, and for each $n \geq 1$ denote

$$F_n = \{\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_n\}.$$

By the hypotheses of the corollary there exists for each $n$ a configuration $c_n$ that tiles correctly at positions $\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_n$. By Theorem 4.2 the sequence $c_1, c_2, \ldots$ has a converging subsequence. Let $c \in \mathcal{P}^{\mathbb{Z}^2}$ be its limit. Then $c$ tiles correctly at every position $\vec{r}_k$ because there are arbitrarily large indices $i$ such that $c$ and $c_i$ assign the same tile to position $\vec{r}_k$ and its neighbors. □

## 4.3 Robinson's aperiodic tile set

It is easy to construct Wang tiles that admit non-periodic tilings. For a long time it was thought that any finite set of prototiles that admits a non-periodic tiling must also admit a periodic one. This conjecture was refuted by R.Berger in 1966 when he constructed a set of Wang prototiles that only admit non-periodic tilings.

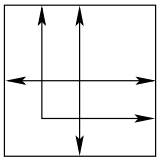
A finite set of prototiles is called aperiodic if

(i) it admits valid tilings, and

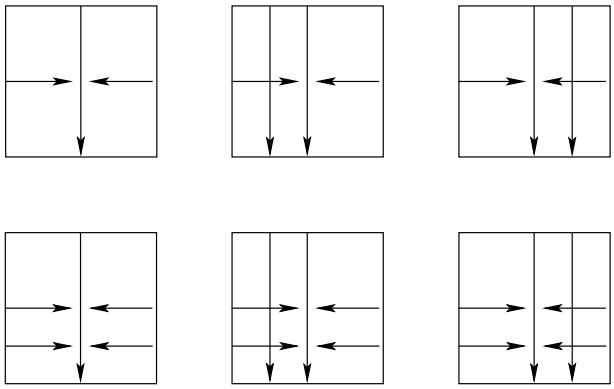
(ii) it does not admit any periodic valid tilings.

As an example of an aperiodic tile set we next describe a set of 56 Wang tiles due to R.M.Robinson. This set will be also useful later in our undecidability proofs. Instead of colors we use arrows to describe the matching rules between tiles. In valid tilings arrow heads and tails in neighboring tiles must match. This formalism can be easily converted into a color-based matching simply by assigning a different color for each orientation and positioning of arrows.

Robinson's tile set consists of tiles



called "crosses" and tiles



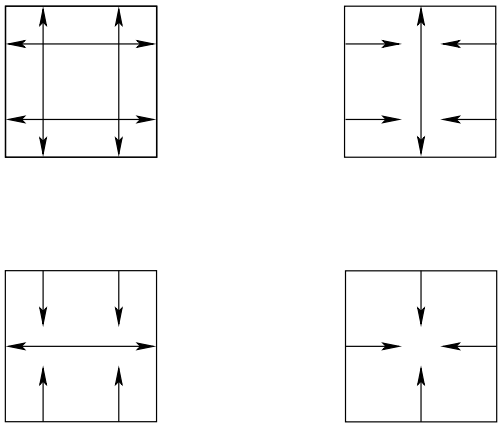
called arms. All tiles may be rotated so each tile comes in four orientations. Hence the total number of such tiles is 28.

The following terminology will be used:

- Every tile has central arrows at the centers of all four sides, and possibly some side arrows.
- A cross is said to face the directions of its side arrows.
- The arrow that runs through an arm is called the principal arrow of the arm, and the direction of the principal arrow is called the direction of the arm.

All six arms above are drawn in the north-to-south orientation. An important fact about arms is that if there are side arrows perpendicular to the principal arrow then these side arrows are towards the head of the principal arrow. Otherwise, all combinations of side arrows are allowed, as shown in the figure above.

We want to enforce a cross in the intersections of every other row and column. This can be established by forming the cartesian product ("sandwich tiles") with the parity tiles



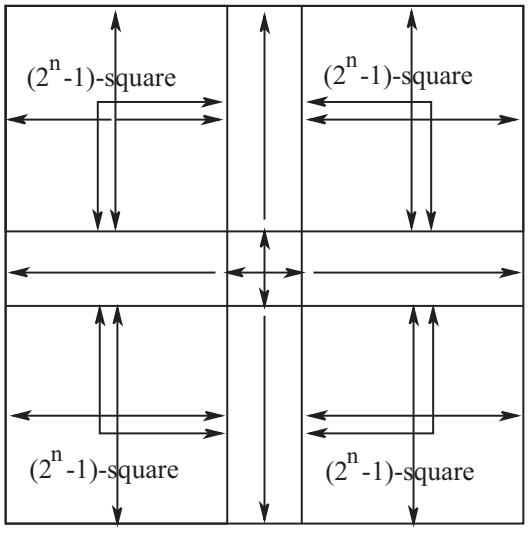
and by forbidding arms from the first parity tile. Since the only way the parity tiles tile the plane is by alternating the tiles on even and odd rows and columns, the first parity tile is forced at the intersections of every other row and column, and hence a cross is forced to appear in those locations. By numbering the rows and columns suitably we can assume from now on that all odd-odd positions of the plane contain a cross.

Note that between two crosses can only appear an arm, and the orientation of the arm has just two possible choices as it cannot point towards either cross. This means that the second parity tile only needs to be paired with north-to-south or south-to-north oriented arms, and the third parity tile is only paired with east-to-west or west-to-east oriented arms. The fourth parity tile is paired with any of the 28 tiles. So the final set contains $4 + 12 + 12 + 28 = 56$ different tiles.

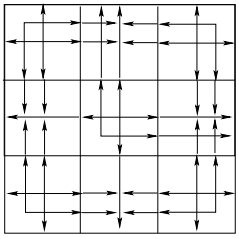
Next we investigate valid tilings admitted by Robinson's tiles, and we show that the tile set is aperiodic. Specific patterns called 1-, 3-, 7-, 15-, ..., $(2^n - 1)$-squares are defined recursively as follows:

(i) A 1-square is a cross at the odd-odd position,

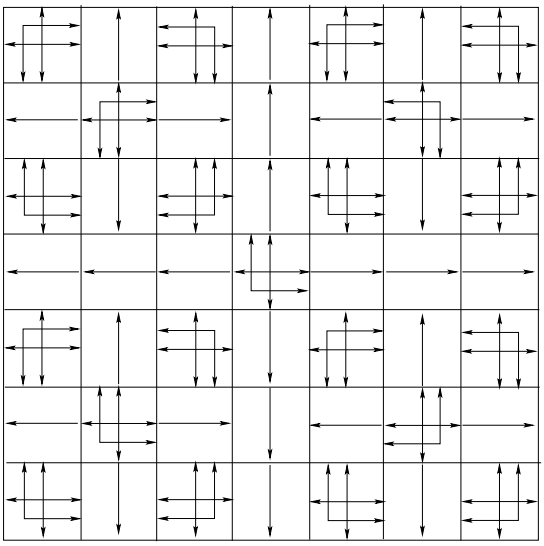
(ii) A $(2^{n+1} - 1)$-square consists of a cross in the middle (in an even-even position), sequences of arms radiating out of the center and four copies of $(2^n - 1)$-squares facing each other at the four quadrants:




Note that for every $n$ there are actually four different $(2^n - 1)$-squares as the cross at the center may be in any of the four possible orientations. For example, the following figure illustrates the 3-square facing north and east:

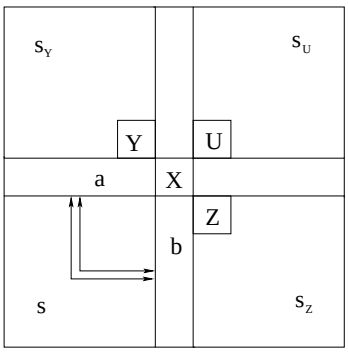


and the following figure shows the 7-square facing north and east. (For clarity, only the central, principal arrows of the arms are shown. The other arrows are uniquely determined by the orientations of the crosses.)



Inductively one easily gets the following properties of $(2^n - 1)$-squares: (1) The tiling is valid within the square, (2) all edges on the border of the square have arrow heads pointing out of the square, so all edge neighbors of $(2^n - 1)$-squares are forced to be arms, and (3) the only side arrows on the border are in the middle of the borders in the directions where the center cross of the square faces.

Consider an arbitrary valid tiling of the plane by Robinson's tiles. Let us show, using mathematical induction on $n$, that every cross in odd-odd position belongs to a unique $(2^n - 1)$-square, for every $n = 1, 2, \ldots$. The case $n = 1$ is trivial, as by definition 1-squares are themselves the crosses at odd-odd positions. Suppose then the claim is true for $n$ and let $C$ be an arbitrary cross in an odd-odd position. By the inductive hypothesis $C$ belongs to a unique $(2^n - 1)$-square $s$. There are four possibilities for the orientation of this square, but they are all symmetric. Let us assume without loss of generality that $s$ faces north and east. In the following discussion we refer to symbols indicating positions in the following figure:



First we prove that tile $X$, outside the north-east corner of square $s$, must be a cross. Suppose the opposite: $X$ is an arm. Then it has an incoming arrow on all but one side, so one of its edge neighbors in regions $a$ or $b$ must be an arm directed towards $X$. By continuing this reasoning we see that all tiles in one of the regions $a$ or $b$ must be arms directed towards $X$. But this means that the tile at the center of region $a$ or $b$ is an arm with an incoming side arrow at the wrong end of the principal arrow: the side arrows are only possible towards the head of the principal arrow. Hence the assumption that $X$ is an arm must be incorrect, and $X$ must be a cross.

Consider then tile $Y$ that is a cornerwise neighbor of $X$. It is in an odd-odd position and therefore $Y$ is a cross. According to the inductive hypothesis $Y$ belongs to a $(2^n - 1)$-square $s_Y$. This square cannot overlap with square $s$ because then the tiles in the overlap region would belong to two different $(2^n - 1)$-squares which contradicts the uniqueness property. Also the tile north of $X$ cannot belong to $s_Y$ because $X$ is a cross. Hence $Y$ has to be at the south-east corner of $s_Y$. Analogously, tiles $Z$ and $U$ are corners of disjoint $(2^n - 1)$-squares $s_Z$ and $s_U$, respectively. Tiles between these $(2^n - 1)$-squares are forced to be arms radiating out from $X$. The side arrows at the middle of $a$ and $b$ force the center crosses of $s_Y$ and $s_Z$ to face squares $s$ and $s_U$, so the squares of $s, s_Y, s_Z, s_U$ and the tiles between them form a $(2^{n+1} - 1)$-square that contains tile $C$.

We have proved the existence of a $(2^{n+1} - 1)$-square that contains $C$. The uniqueness is obvious as the orientation of the (unique) $(2^n - 1)$-square $s$ that contains $C$ determines the location of the center of the $(2^{n+1} - 1)$-square that contains $C$.

We have proved that every 1-square belongs to a 3-square, which belongs to a 7-square, which belongs to a 15-square and so on. Based on this observation we can state:

**Lemma 4.4** *Robinson's tiles form an aperiodic protoset.*

*Proof.* The $(2^n - 1)$-squares are valid tilings of arbitrarily large squares, so a valid tiling of the plane exists (Corollary 4.3).

The centers of the quadrants of any $(2^n - 1)$-square are crosses separated by $(2^{n-1} - 1)$ arms. As every valid tiling contains $(2^n - 1)$-squares for every $n$, the tiling contains horizontally aligned crosses separated by arbitrarily long sequences of arms. So there can be no horizontal period, and a periodic tiling is not possible.

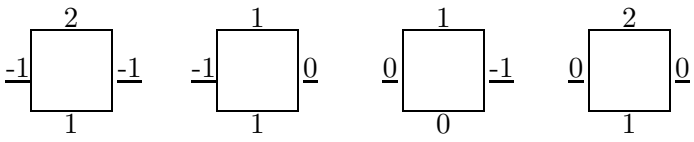
□

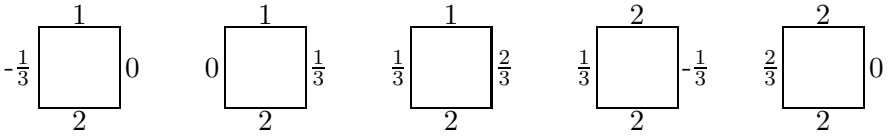
## 4.4 An aperiodic set of 14 Wang tiles

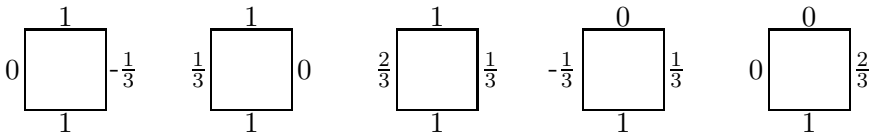
We have learned the aperiodic set of 56 Wang tiles by Robinson. In this section we learn a very different method of constructing aperiodic tile sets that yields a set with only 14 tiles, shown in the figure below. But note that even smaller aperiodic sets exist: E. Jeandel and M. Rao have an aperiodic Wang tile set that contains just 11 tiles, and they proved that 11 is the smallest possible size.

In our 14 tile set, the edges are labeled with rational numbers. Each number represents one color, so in valid tilings neighboring tiles must match in the numbers at the abutting edges. Notice also the the labels of the vertical edges of the first four tiles are underlined: This means that those numbers represent a different color than the same numbers without a line underneath.

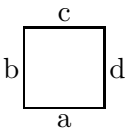
The set consists of two parts: the first four tiles form the set $\mathcal{P}_2$ and the set of the last ten tiles is called $\mathcal{P}_{2/3}$. The aperiodic set $\mathcal{P}$ is the union of these two sets. As the vertical sides of the elements of the two parts have different labels, it is clear that on any valid tiling of the plane by $\mathcal{P}$, each horizontal row is tiled by tiles that come from $\mathcal{P}_2$ or $\mathcal{P}_{2/3}$ only.




The tiles perform arithmetic operations in the following sense: We say that tile




multiplies by $q$ if $qa + b = c + d$. In other words, the tile multiplies the "input" number $a$ on its bottom edge by $q$, adds the "carry forward" $b$ from the left edge, and splits the result between the "output" $c$ at the top edge and the "carry forward" $d$ to the right. It is easy to verify that the tiles in $\mathcal{P}_2$ all multiply by 2, and the tiles in $\mathcal{P}_{2/3}$ multiply by $\frac{2}{3}$.

Consider a horizontal segment of $n$ tiles that all multiply by the same number $q$. Let $a_i, b_i, c_i$ and $d_i$ be the numbers on the $i$'th tile so that $qa_i + b_i = c_i + d_i$, for all $i = 1, 2, \ldots n$. Summing up over all $n$ tiles we get

$$q\sum_{i=1}^{n} a_i + \sum_{i=1}^{n} b_i = \sum_{i=1}^{n} c_i + \sum_{i=1}^{n} d_i.$$

If the tiling constraint is satisfied then we have $d_i = b_{i+1}$ for all $i = 1, 2, \ldots, n-1$, and if the segment also starts and ends with the same carry forward $d_n = b_1$ we have that

$$\sum_{i=1}^{n} d_i = \sum_{i=1}^{n} b_i.$$

This happens if the segment is extracted from a periodic tiling with horizontal period $n$. Then

$$q\sum_{i=1}^{n} a_i = \sum_{i=1}^{n} c_i.$$

**Theorem 4.5** *The set $\mathcal{P}$ of 14 Wang prototiles above is aperiodic.*

*Proof.* We have two facts to prove: (i) no periodic tiling is possible, and (ii) some valid tiling exists.

(i) Suppose the opposite is true: there exists a periodic tiling $f : \mathbb{Z}^2 \longrightarrow \mathcal{P}$. Then we know that such a periodic tiling must have a horizontal period $h$ and a vertical period $v$, for some $h, v > 0$. (In fact we could choose these two numbers to be identical.)

Let $a_{i,j}, b_{i,j}, c_{i,j}$ and $d_{i,j}$ be the colors on the south, west, north and east edges of the tile $f(i,j)$ in position $(i,j) \in \mathbb{Z}^2$. It follows from the tiling rule that $d_{i,j} = b_{i+1,j}$ and that $c_{i,j} = a_{i,j+1}$. As discussed above, we have

$$q_j \sum_{i=1}^{h} a_{i,j} = \sum_{i=1}^{h} c_{i,j} = \sum_{i=1}^{h} a_{i,j+1},$$

where $q_j = 2$ or $\frac{2}{3}$ depending on whether the tiles on row $j$ come from set $\mathcal{P}_2$ or $\mathcal{P}_{2/3}$. By combining these equations for rows $j = 1, 2, \ldots, v$ we get the result that

$$q_1 q_2 q_3 \ldots q_v \sum_{i=1}^{h} a_{i,1} = \sum_{i=1}^{h} a_{i,v+1} = \sum_{i=1}^{h} a_{i,1}.$$

It is clear from the tiles that we cannot have a horizontal row of tiles such that the bottom edges all have value 0, so we have $\sum_{i=1}^{h} a_{i,1} > 0$. Hence we can divide $\sum_{i=1}^{h} a_{i,1}$ from the equation, which leaves

$$q_1 q_2 q_3 \ldots q_v = 1.$$

But each $q_j$ is either 2 or $\frac{2}{3}$, and any product of these numbers is some power of 2 divided by a power of 3. Numbers 2 and 3 are relative primes, so no such product can equal 1, a contradiction.

(ii) It is enough to construct one valid tiling. The tiling will of course be non-periodic. We use the following notations and concepts. For any real number $x$, the floor $\lfloor x \rfloor$ of $x$ is the largest integer not greater than $x$, that is, $\lfloor x \rfloor$ is the unique integer that satisfies $x - 1 < \lfloor x \rfloor \leq x$. Analogously, the ceiling $\lceil x \rceil$ is the smallest integer that is not smaller than $x$. By the *balanced representation* $B(x)$ of real number $x$ we mean the bi-infinite sequence $\ldots B(x)_{-1}, B(x)_0, B(x)_1, B(x)_2, \ldots$ whose $i$'th term is $B(x)_i = \lfloor ix \rfloor - \lfloor (i-1)x \rfloor$. Notice that the elements of the sequence are integers, and

$$\begin{aligned} B(x)_i &= \lfloor ix \rfloor - \lfloor (i-1)x \rfloor < ix - ((i-1)x - 1) = x + 1, \text{ and} \\ B(x)_i &= \lfloor ix \rfloor - \lfloor (i-1)x \rfloor > ix - 1 - (i-1)x = x - 1. \end{aligned}$$

Hence each element of the sequence $B(x)$ is either $\lfloor x \rfloor$ or $\lceil x \rceil$.

Consider an arbitrary real number $x \in \left[\frac{1}{2}, 1\right]$. We have that $0 \leq x \leq 1$ and $1 \leq 2x \leq 2$. The symbols in the balanced sequences for $x$ and $2x$ are 0's and 1's, and 1's and 2's, respectively. Let us show that the prototiles of $\mathcal{P}_2$ admit a tiling of an bi-infinite horizontal strip whose bottom labels read the sequence $B(x)$ and the top labels read $B(2x)$. Let

$$\begin{aligned} a_i &= B(x)_i, \\ b_i &= 2\lfloor (i-1)x \rfloor - \lfloor (i-1)(2x) \rfloor, \\ c_i &= B(2x)_i, \text{ and} \\ d_i &= 2\lfloor ix \rfloor - \lfloor i(2x) \rfloor \end{aligned}$$

be the labels on the south, west, north and east edges of the tile in position $i \in \mathbb{Z}$ of the strip. It is clear from this definition that $b_i = d_{i-1}$ so the labels match on the tiling of the strip. Let us analyze values of $a_i, b_i, c_i$ and $d_i$ to prove that the tile with these labels is in our set $\mathcal{P}_2$.

From the properties of the balanced sequences we know that $a_i \in \{0, 1\}$ and $c_i \in \{1, 2\}$. Clearly,

$$\begin{aligned} d_i - b_i &= 2\lfloor ix \rfloor - \lfloor i(2x) \rfloor - (2\lfloor (i-1)x \rfloor - \lfloor (i-1)(2x) \rfloor) \\ &= 2(\lfloor ix \rfloor - \lfloor (i-1)x \rfloor) - (\lfloor i(2x) \rfloor - \lfloor (i-1)(2x) \rfloor) \\ &= 2a_i - c_i, \end{aligned}$$

so the tiles of the strip multiply by number 2. We also have

$$d_i = 2\lfloor ix \rfloor - \lfloor i(2x) \rfloor < 2ix - (2ix - 1) = 1,$$

and

$$d_i = 2\lfloor ix \rfloor - \lfloor i(2x) \rfloor > 2(ix-1) - 2ix = -2.$$

Because $d_i$ is an integer, the only possible values of $d_i$ (and hence also $b_i$) are $-1$ and $0$. The following possibilities remain:

$$\begin{array}{lcll}
a_i = 0, c_i = 2 & \Longrightarrow & d_i - b_i = 2a_i - c_i = -2, & \text{not possible,} \\
a_i = 0, c_i = 1 & \Longrightarrow & d_i - b_i = 2a_i - c_i = -1 & \Longrightarrow \ d_i = -1, b_i = 0, \\
a_i = 1, c_i = 2 & \Longrightarrow & d_i - b_i = 2a_i - c_i = 0 & \Longrightarrow \ d_i = b_i = -1 \text{ or } d_i = b_i = 0, \\
a_i = 1, c_i = 1 & \Longrightarrow & d_i - b_i = 2a_i - c_i = 1 & \Longrightarrow \ d_i = 0, b_i = -1.
\end{array}$$

Only four possibilities exist, and these are precisely the four tiles in $\mathcal{P}_2$.

Next we analyze $\mathcal{P}_{2/3}$ in a similar way. Let $x \in [1,2]$, so that $1 \le x \le 2$ and $\frac{2}{3} \le \frac{2}{3}x \le \frac{4}{3}$. The balanced representations of $x$ and $\frac{2}{3}x$ consist of 1's and 2's, and 0's, 1's and 2's, respectively. Let us show that there is a tiling by $\mathcal{P}_{2/3}$ of a bi-infinite strip such that the labels on the bottom and the top of the strip read the balanced representations $B(x)$ of $x$ and $B(\frac{2}{3}x)$ of $\frac{2}{3}x$. For brevity, let us denote $q = \frac{2}{3}$. The tile in position $i$ of the strip has labels

$$\begin{array}{rcl}
a_i & = & B(x)_i, \\
b_i & = & q\lfloor (i-1)x \rfloor - \lfloor (i-1)(qx) \rfloor, \\
c_i & = & B(qx)_i, \text{ and} \\
d_i & = & q\lfloor ix \rfloor - \lfloor i(qx) \rfloor.
\end{array}$$

The consecutive tiles of the strip match as $b_i = d_{i-1}$. We know that $a_i \in \{1,2\}$ and $c_i \in \{0,1,2\}$. As above, we also have

$$\begin{array}{rcl}
d_i - b_i & = & q\lfloor ix \rfloor - \lfloor i(qx) \rfloor - (q\lfloor (i-1)x \rfloor - \lfloor (i-1)(qx) \rfloor) \\
& = & q(\lfloor ix \rfloor - \lfloor (i-1)x \rfloor) - (\lfloor i(qx) \rfloor - \lfloor (i-1)(qx) \rfloor) \\
& = & qa_i - c_i.
\end{array}$$

We also have

$$d_i = q\lfloor ix \rfloor - \lfloor i(qx) \rfloor < qix - (qix - 1) = 1,$$

and

$$d_i = q\lfloor ix \rfloor - \lfloor i(qx) \rfloor > q(ix-1) - qix = -q.$$

Because $d_i$ is an integer multiple of $\frac{1}{3}$, the only possible values of $d_i$ (and hence also $b_i$) are $-\frac{1}{3}, 0, \frac{1}{3}$ and $\frac{2}{3}$. The following possibilities remain:

$$\begin{array}{lcl}
a_i = 1, c_i = 2 & \Longrightarrow & d_i - b_i = qa_i - c_i = -\frac{4}{3}, \\
& & \text{not possible,} \\
\\
a_i = 1, c_i = 1 & \Longrightarrow & d_i - b_i = qa_i - c_i = -\frac{1}{3} \\
& \Longrightarrow & d_i = -\frac{1}{3}, b_i = 0 \text{ or } d_i = 0, b_i = \frac{1}{3} \text{ or } d_i = \frac{1}{3}, b_i = \frac{2}{3}, \\
\\
a_i = 1, c_i = 0 & \Longrightarrow & d_i - b_i = qa_i - c_i = \frac{2}{3} \\
& \Longrightarrow & d_i = \frac{1}{3}, b_i = -\frac{1}{3} \text{ or } d_i = \frac{2}{3}, b_i = 0, \\
\\
a_i = 2, c_i = 2 & \Longrightarrow & d_i - b_i = qa_i - c_i = -\frac{2}{3} \\
& \Longrightarrow & d_i = -\frac{1}{3}, b_i = \frac{1}{3} \text{ or } d_i = 0, b_i = \frac{2}{3}, \\
\\
a_i = 2, c_i = 1 & \Longrightarrow & d_i - b_i = qa_i - c_i = \frac{1}{3} \\
& \Longrightarrow & d_i = 0, b_i = -\frac{1}{3} \text{ or } d_i = \frac{1}{3}, b_i = 0 \text{ or } d_i = \frac{2}{3}, b_i = \frac{1}{3}, \\
\\
a_i = 2, c_i = 0 & \Longrightarrow & d_i - b_i = qa_i - c_i = \frac{4}{3}, \\
& & \text{not possible.}
\end{array}$$

The ten possibilities are exactly the tiles of $\mathcal{P}_{2/3}$.

We have proved that for every $x \in \left[\frac{1}{2}, 1\right]$ we can tile a strip whose bottom and top edges read the sequences $B(x)$ and $B(2x)$, respectively, and for every $x \in [1, 2]$ we can tile a strip whose bottom and top edges read the sequences $B(x)$ and $B(\frac{2}{3}x)$. Define the following real function $f : \left(\frac{2}{3}, 2\right] \longrightarrow \left(\frac{2}{3}, 2\right]$:

$$f(x) = \begin{cases} 2x, & \text{if } x \leq 1, \text{ and} \\ \frac{2}{3}x, & \text{if } x > 1. \end{cases}$$

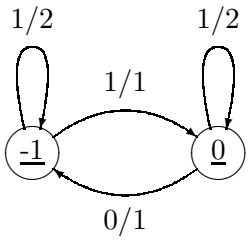
It is easy to see that the range of $f$ is the half-open interval $\left(\frac{2}{3}, 2\right]$, so function $f$ is surjective. (In fact, function $f$ is a bijection, but this fact is not relevant to the reasoning below.) It follows from the surjectivity of $f$ that there exist bi-infinite sequences $\ldots x_{-1}, x_0, x_1, x_2, \ldots$ of real numbers such that $x_{j+1} = f(x_j)$ for all $j \in \mathbb{Z}$. In fact, since one element $x_0$ of the sequence can be chosen arbitrarily from the half-open interval $\left(\frac{2}{3}, 2\right]$, the number of such sequences is uncountably infinite.

As proved above, for each $j \in \mathbb{Z}$ we can tile an infinite strip whose edges read $B(x_j)$ and $B(x_{j+1})$. By stacking these strips on top of each other we obtain a tiling of the plane. In fact, we proved there exist uncountably many different valid tilings.

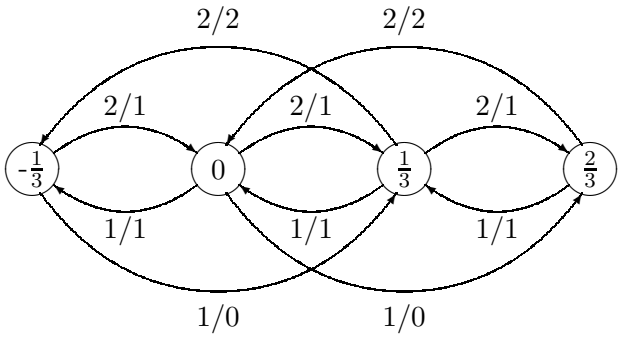
□

The following diagram illustrates the tiles as a directed graph whose nodes are labeled by the vertical colors, and the edges are labeled by pairs of "input"/"output" symbols. Each edge corresponds to a tile: the tile with labels $a$, $b$, $c$ and $d$ is the edge from node $b$ to node $d$ that is labeled by $a/c$. Any bi-infinite path through the diagram that follows the edges gives a valid tiling of one bi-infinite strip. Such a diagram is called a finite state transducer.

$M_2$ :




$M_{2/3}$ :




Analogously, it is easy to construct for any given rational number $q$ a finite set of tiles that multiply balanced sequences representing numbers of a given interval by $q$. The requested tiles have edge labels

$$\begin{aligned} a &= B(x)_i, \\ b &= q\lfloor (i-1)x \rfloor - \lfloor (i-1)(qx) \rfloor, \\ c &= B(qx)_i, \text{ and} \\ d &= q\lfloor ix \rfloor - \lfloor i(qx) \rfloor. \end{aligned}$$

where $x$ is a number in the desired interval and $i \in \mathbb{Z}$. A simple analysis shows that we always have $-q < b < 1$ and $-q < d < 1$, so there only is a finite number of such tiles.

The balanced sequences we used in the proof have many interesting properties. Balanced sequences of rational numbers are periodic, but if $x$ is irrational then $B(x)$ is non-periodic, but only "barely so": it is a so-called Sturmian sequence, which means that the number of different subsegments of length $n$ is $n+1$, for every $n$. This is the smallest possible number of different subsegments of length $n$ in any non-periodic infinite sequence.

# 5 Undecidable problems concerning tiles

The following question (known as the domino problem or the tiling problem) arises naturally: How can one determine if a given finite set of Wang prototiles admits a tiling ? Does there exist some simple (or even complicated) properties that one can use to develop a computer program to determine if a tiling is possible. The input to the program should be an arbitrary finite set of Wang tiles, and the output should be "yes" or "no" depending on whether the input admits a tiling. In this section we show that such a computer program does not exist. The non-existence is a mathematical fact that cannot be overcome by building more powerful computers or by developing new programming languages or tools. The undecidability will be deduced from Turing's result on the undecidability of the halting problem of Turing machines, using the reduction technique.

The tiling problem is an example of a decision problem. A decision problem is a problem that has an input parameter, and the answer to the problem is always "yes" or "no". When we fix the value of the input parameter we get an instance of the problem. An instance is called a "yes"-instance or a "no"-instance depending on whether the answer to the decision problem is "yes" or "no", respectively. For example, the problem "Does a given quadratic polynomial have a real root ?" is a decision problem. Quadratic polynomials are instances (we always use the keyword "given" in the decision problem statement to indicate the input). For example, $x^2-2x+2$ is a "no"-instance, while $x^2+2x-1$ is a "yes"-instance to this problem. In the tiling problem, instances are finite sets of Wang tiles, and an instance is a "yes"-instance iff the protoset admits a tiling of the plane. The complement of a decision problem is the decision problem where the "yes" and the "no" answers have been switched: for example, the complement of the tiling question asks whether the given protoset does not admit a tiling. "Yes" means now that no tiling is possible.

An algorithm can be formally defined in various ways. In order to keep the discussion simple, we are going to leave it undefined. For our purposes it is sufficient to understand a (decision) algorithm to be a computer program that takes some input and returns a "yes" or a "no" answer on each input. We say that the algorithm solves a decision problem if the algorithm returns the correct yes/no -answer on every instance of the problem. If such an algorithm exists then the decision problem is called decidable and if no such algorithm exists then the problem is undecidable.

Strictly speaking, the input of a computer program is a string of bits (or a string over some other alphabet) so the instance of a decision problem has to be encoded into such a string. For example, Wang tiles could be encoded as sequences of four colors, each color represented as a unique binary string. There are of course many ways to do such an encoding, but all encodings are equivalent in the sense that the decidability status of the decision problem is not affected by the encoding. In our discussion encodings of inputs will be irrelevant as we are not going to write any actual programs – rather algorithms will be defined by describing in plain English the steps that the algorithm executes. The idea is that we all are sufficiently familiar with computer programming so that such a description (when detailed enough) will convince everyone that a computer program exists for solving the program. Notice also that when describing an algorithm we do not need to worry about things related to computing resources such as memory space etc. We are always supposed to have unlimited amounts of such resources.

For example, the following algorithm solves the question whether a given quadratic polynomial has a real root: Let $ax^2+bx+c$ be the input to the algorithm. The algorithm starts by computing the discriminant $D=b^2-4ac$. Then it checks whether $D \geq 0$ or not. If $D \geq 0$ then the algorithm returns

"yes", and if $D < 0$ then the algorithm returns "no". This level of description should convince everyone that the problem is decidable.

Notice that from the decidability point of view any problem that has no input instance or has only a finite number of possible instances is trivially decidable. For example, a question like "Is the Riemann hypothesis true ?" is decidable. There is a trivial algorithm that solves this problem, we just do not know what that algorithm is. (The algorithm is either the algorithm that only types "yes" or the algorithm that only types "no".) In the same way, any problem with a finite number of possible input instances is solved by a program that simply looks-up from a finite table the correct answer corresponding to the given input. For example, for any fixed number $N$ it is decidable if a given set of $N$ Wang tiles admits a valid tiling, as there are only finitely many such sets, up to renaming of the colors. So decidability questions are only relevant in connection to decision problems with an unlimited number of possible input instances.

A semi-algorithm is a weaker concept than an algorithm: it is a computer program that halts and returns "yes" if the input is a positive instance of the problem, but it may run forever, without ever halting, on negative input instances. So a semi-algorithm (semi-) solves a decision problem if on every "yes" -instance of the problem it returns the correct "yes" -answer, but when the input is a "no" -instance it may run forever without ever returning an answer. If an answer is returned, it has to be the correct answer: semi-algorithms never return wrong answers. If a decision problem has a semi-algorithm then it is called semi-decidable. Clearly every decidable problem is also semi-decidable as an algorithm is also a semi-algorithm. As an example of a semi-algorithm that is not an algorithm consider the following process of determining if a given Wang protoset does not admit a tiling.

**Lemma 5.1** *The complement of the tiling problem "Does a given finite set $\mathcal{P}$ of Wang prototiles admit a tiling ?" is semi-decidable.*

*Proof.* The semi-algorithm enumerates positive integers $n = 1, 2, 3, \ldots$ one-by-one. For each $n$ it tries all possible ways of tiling the $n \times n$ square by the given prototiles. It can simply try (in the lexicographic order) each sequence of $n^2$ tiles, write the tiles inside the $n \times n$ square row-by-row and check whether the tiles match or not. For each $n$ there are only a finite number of sequences to try, and an algorithm can easily go through all of them one-by-one. If we find a valid tiling of the $n \times n$ square we increment $n$ and repeat. If we do not find a tiling of the $n \times n$ square then a valid tiling of the plane does not exist, so the semi-algorithm returns answer "yes" to indicate that no tiling is admitted. Notice that if the given instance is a "no"-instance (i.e. admits a tiling) the process will never end as we keep on tiling larger and larger squares. But on every "yes" -instance (i.e. no tiling exist) the process will terminate with the correct output, because by Corollary 4.3 some $n \times n$ square cannot be tiled. We conclude that the complement of the tiling problem is semi-decidable. □

We make the following observations:

**Theorem 5.2** *A decision problem is decidable if and only if the complement problem is decidable. A decision problem is decidable iff the problem and its complement are both semi-decidable.*

*Proof.* If decision problem $P$ is decidable then there exists an algorithm $A$ that solves $P$. We get an algorithm for the complement problem if we simply switch the output of $A$, or more precisely, make a new algorithm $A'$ that (i) calls $A$ as a subroutine with the original input of $A'$, and (ii) if $A$ returns "yes" algorithm $A'$ returns "no" and if $A$ returns "no" then $A'$ returns "yes". This proves the first claim.

Algorithm is also a semi-algorithm so if $P$ is decidable it is also semi-decidable and, since the complement problem is decidable, the complement problem is also semi-decidable. Conversely, suppose that problem $P$ has a semi-algorithm $A$ and the complement of $P$ has a semi-algorithm $A'$. Here is a description of an algorithm that solves $P$: With a given input we execute both $A$ and $A'$ at the same time. This

can be arranged by, for example, alternating between the semi-algorithms by executing one step of each semi-algorithm in turn. Eventually one of the two semi-algorithms will return the "yes"-answer. If the instance is a "yes" -instance then it will be $A$ that gives the answer and if the instance is a "no" -instance then $A'$ will give the answer. In the first case our algorithm returns "yes", in the second case we return "no".

□

In view of the previous theorem and lemma: if the tiling problem were semi-decidable then it would be also decidable (because we know that the complement problem is semi-decidable). Equivalently: Once we prove that the tiling problem is undecidable, it implies that the tiling problem is not semi-decidable either.

In order to prove from the scratch that a problem is undecidable we would need a more precise definition of an algorithm. Here, to avoid too deep and time consuming involvement in the computation theory (which a topic of another course) we are not going to prove undecidability results from the scratch. Rather, we take the classic result by Alan Turing without a proof. This result provides us with one decision problem (the halting problem of Turing machines without input) that is known to be undecidable. Turing's proof of this result is a very nice — and not very difficult — diagonal argument similar to Cantor's proof for the uncountability of real numbers.

Once we have one undecidable decision problem available, we use the technique of reduction to prove other problems undecidable. Reduction is an indirect proof technique that works as follows: Suppose $P$ is a known undecidable problem (e.g. the halting problem of Turing machines), and we want to prove that problem $Q$ is also undecidable. We make the assumption (indirect proof!) that there exists an algorithm $A$ that solves $Q$. Then we describe an algorithm that solves $P$, using $A$ as a subroutine. Since $P$ is undecidable, no such algorithm can exist, so algorithm $A$ cannot exist either. In the reduction technique we design an algorithm for $P$ in order to prove that no algorithm exists that solves $Q$.

## 5.1 Turing machines

Let us start by defining Turing machines. A Turing machine is a simple computing device that consists of a bi-infinite tape that serves as the memory and a finite state processor that moves over the tape. The tape consists of a sequence of memory locations, indexed by $\mathbb{Z}$, each of which contains an element of a finite set $\Gamma$, called the tape alphabet. So the content of the tape at any given time is given by a function $f : \mathbb{Z} \longrightarrow \Gamma$ where $f(i)$ is the symbol at location $i$. One element $b \in \Gamma$ is specified as the blank symbol, and in the beginning of the computation all tape locations contain symbol $b$.

At all times, the processor (also called the control unit) of the machine accesses one tape location $i \in \mathbb{Z}$. The control unit is in some state $q$ that is an element of a finite state set $S$. Depending on the current state $q$ and the current tape symbol $f(i) \in \Gamma$ at the current location $i$ of the control unit, the Turing machine changes the state of the control unit, replaces the tape symbol $f(i)$ with a new symbol, and moves the control unit one position to the left or right on the tape. This action of the machine is specified by its transition function

$$\delta : S \times \Gamma \longrightarrow S \times \Gamma \times \{L, R\}.$$

The interpretation of $\delta(q, x) = (p, y, d)$ is that if the current state is $q$ and the tape symbol at the current location $i$ is $x$ then the machine changes the state into $p$, replaces $x$ by $y$ on the tape and moves one position left or right on the tape depending on whether $d = L$ or $d = R$.

The configuration of the machine is an element of $S \times \mathbb{Z} \times \Gamma^{\mathbb{Z}}$. Configuration $(q, i, f)$ specifies the current state $q$ of the machine, its current position $i$ on the tape, and the current content $f$ of the entire tape. Formally we can now define one move of the machine: Configuration $(q, i, f)$ is transformed in one move into the configuration $(p, j, g)$, where $\delta(q, f(i)) = (p, y, d)$, $g(i) = y$, $g(k) = f(k)$ for all $k \neq i$, and

$j = i + 1$ if $d = R$ and $j = i - 1$ if $d = L$. We denote this move by

$$(q, i, f) \vdash (p, j, g).$$

In the beginning of the computation the Turing machine is in one specific state $s_0 \in S$ called the initial state, and another state $s_h \in S$ is specified as the halting state. The Turing machine halts when the control unit enters state $s_h$. The Turing machine then can be understood as a dynamical system where the transformation $\vdash$ is applied repeatedly starting from the initial configuration $(s_0, 0, f_b)$ where $f_b(k) = b$ for all $k \in \mathbb{Z}$ until (if ever) the machine reaches and halts in some configuration $(s_h, i, f)$, where $i$ and $f$ can be arbitrary.

To specify a Turing machine one needs to provide six items. We say that a Turing machine is a six-tuple $M = (S, \Gamma, \delta, s_0, s_h, b)$ where $S$ and $\Gamma$ are finite sets, $s_0, s_h \in S$ and $b \in \Gamma$ are elements of those sets, and $\delta : S \times \Gamma \longrightarrow S \times \Gamma \times \{L, R\}$ is a function. Note that in the most common terminology in the literature one also specifies a third finite set, the input alphabet, but we can ignore it here because we only discuss Turing machines without input.

**Example 9.** Consider the following Turing machine $M = (\{s, t, h\}, \{a, b\}, \delta, s, h, b)$ where

$$\begin{aligned} \delta(s, a) &= (t, a, L) \\ \delta(s, b) &= (t, a, R) \\ \delta(t, a) &= (h, a, L) \\ \delta(t, b) &= (s, a, L) \end{aligned}$$

(and the values of $\delta(h, \ldots)$ do not matter as $h$ is the halting state.) The computation by $M$ proceeds as follows:

$$\ldots bbbb \overset{s}{b} bb \ldots \vdash \ldots bbbba \overset{t}{b} b \ldots \vdash \ldots bbb \overset{s}{a} abb \ldots \vdash \ldots bb \overset{t}{b} aab \ldots \vdash$$
$$\ldots b \overset{s}{b} aaab \ldots \vdash \ldots ba \overset{t}{a} aab \ldots \vdash \ldots b \overset{h}{a} aaab \ldots$$

□

Note that at all times only a finite number of tape locations may contain symbols that are different from the blank symbol $b$. Hence configurations $(s, i, f)$ have a finite representation. The following result is given without proof:

**Theorem 5.3 (Turing 1936)** *There is no algorithm to solve the following decision problem: "Does a given Turing machine $M$ eventually halt ?"*

□

Note that the decision problem of the previous theorem is semi-decidable: A simple semi-algorithm for the "yes" instances simply simulates the Turing machine step-by-step until it halts, if ever. Once the halting state $q_h$ is reached, the semi-algorithm returns answer "yes". The "no" answer is never returned: if the Turing machine does not halt then the simulation continues indefinitely.

Based on Theorem 5.3 we can now prove other problems undecidable using the reduction technique discussed earlier.

**Example 10.** As an example of a reduction, let us prove that the following decision problem is undecidable: "Given a Turing machine $M$ and a tape symbol $a$, does the Turing machine $M$ eventually write symbol $a$ somewhere on the tape ?" Let us call this problem $Q$, and let us call the halting problem of Theorem 5.3 problem $P$. Suppose we have an algorithm $A$ that solves $Q$. Then there exists the following algorithm $B$ to solve $P$:

Algorithm $B$ gets as input a Turing machine $M = (S, \Gamma, \delta, q_0, q_h, b)$. In order to determine if $M$ halts, algorithm $B$ creates a new Turing machine $M' = (S \cup \{q'\}, \Gamma \cup \{a\}, \delta', q_0, q', b)$ where $q' \notin S$ is the new

halting state, $a \notin \Gamma$ is a new tape symbol, and $\delta'$ is exactly like $\delta$, except for the following modification: For every $x \in \Gamma$ we set $\delta(q_h, x) = (q', a, R)$. For all $x \in \Gamma \cup \{a\}$ and $q \in S \cup \{q'\}$ the values of the new entries $\delta(q', x)$ and $\delta(q, a)$ of $\delta$ can be chosen arbitrarily. The idea is that the modified $M'$ works exactly like $M$ until (if ever) $M$ enters the halting state $q_h$. Instead of halting in state $q_h$ the new machine writes then the new tape letter $a$ on the tape and halts. Clearly, $M$ halts if and only if $M'$ writes symbol $a$ on the tape.

Algorithm $B$ can easily construct $M'$. Then $B$ gives this $M'$ and symbol $a$ as the input to the algorithm $A$. Algorithm $A$ returns "yes" or "no" depending on whether $M'$ eventually writes $a$ on the tape or not. Algorithm $B$ then simply returns that same answer.

We described an algorithm $B$ that according to Theorem 5.3 does not exist. Therefore the assumption that algorithm $A$ exists must be incorrect.

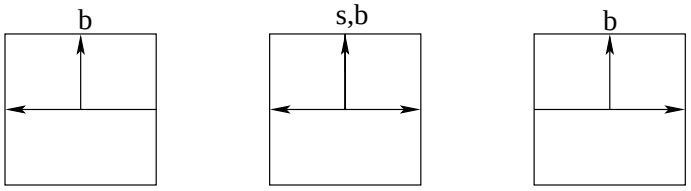
□

## 5.2 The tiling problem with a seed tile

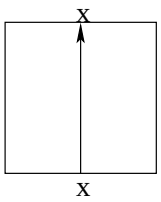
To use the reduction technique in connection to tiling questions we start by designing Wang protosets that simulate Turing machines. A valid tiling will picture the entire computation history by a Turing machine, move-by-move. Instead of colors on the edges of the tiles we use labeled arrows. In a valid tiling, each arrow head and tail must meet a tail and head, respectively, with the same label. The tiling constraints using such arrows can then be easily transformed into color constraints by replacing an arrow with label $L$ and direction $D$ by color $(L, D)$, where $D$ can be North, East, South or West.

The labels of the arrows will be tape symbols (representing a tape location containing that symbol) and state/tape symbol pairs (representing a tape location containing the control unit at the given state). Any given Turing machine $M = (S, \Gamma, \delta, s, h, b)$ will be represented by a set $\mathcal{P}_M$ of Wang tiles that contains

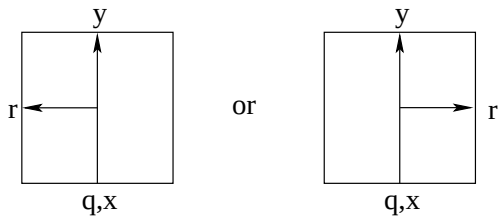
(i) the following three starting tiles to represent the blank tape




(ii) for every tape letter $x \in \Gamma$ an alphabet tile

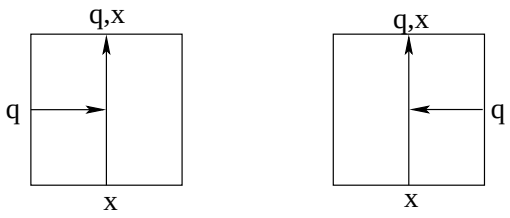



(iii) for every non-halting state $q \in S \setminus \{h\}$ and tape symbol $x \in \Gamma$ one action tile

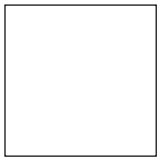



where the left tile is used iff $\delta(q, x) = (r, y, L)$ and the right tile iff $\delta(q, x) = (r, y, R)$,

(iv) for every non-halting state $q \in S \setminus \{h\}$ and tape symbol $x \in \Gamma$ the two merging tiles

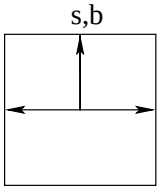



(v) the blank tile



**Theorem 5.4** *The following decision problem is undecidable: "Given a finite set $\mathcal{P}$ of Wang prototiles and one specified seed tile $t \in \mathcal{P}$, does $\mathcal{P}$ admit a valid tiling of the plane that contains at least one occurrence of $t$ ?"*

*Proof.* Suppose an algorithm exists. Then we can solve the halting problem of Turing machines as follows: For any given Turing machine $M$ our algorithm starts by constructing the set $\mathcal{P}_M$ described above. This set can clearly be mechanically constructed based on $M$. Then set $\mathcal{P}_M$ and the seed tile




are given as input to the algorithm that determines if a tiling exists that contains the seed tile. Such a tiling exists if and only if $M$ does not halt: The seed tile uniquely determines the entire tiling. Tiles on the same horizontal row as the seed must be starting tiles of type (i). Horizontal rows above are forced to represent consecutive configurations of the Turing machine. If the machine halts then the tiling becomes impossible as there are no action tiles of type (iii) for the halting state. But if the machine $M$ never halts then the tiling can be continued indefinitely, to fill the entire upper half of the plane. The lower half plane can be filled with the blank tile of type (v).

□

Note that the decision problem in the previous theorem is not the same as the tiling problem. The request for the named seed tile to appear in the tiling makes the proof easy, as it allows us to guarantee the proper initialization of the Turing machine computation.

### 5.3 Finite systems of forbidden patterns

Before moving on to other decision problems concerning tiles, we simplify the discussion by relaxing the requirement to use colors or arrows to specify which tiles may be put next to each other. Rather, correctness of a tiling will be specified by a finite collection of *forbidden patterns*. A configuration is then a valid tiling if and only if it does not contain a forbidden pattern.

More precisely, let us first define a *neighborhood vector*

$$N = (\vec{n}_1, \vec{n}_2, \ldots, \vec{n}_m)$$

where each $\vec{n}_i \in \mathbb{Z}^2$ and $\vec{n}_i \neq \vec{n}_j$ for all $i \neq j$. The elements $\vec{n}_i$ specify the relative locations of the neighbors of each position: Position $\vec{n} \in \mathbb{Z}^2$ has $m$ *neighbors* $\vec{n} + \vec{n}_i$ for $i = 1, 2, \ldots, m$.

Let $T$ be a finite set, the set of prototiles, and let $R \subseteq T^m$ be a relation specifying which patterns are allowed in valid tilings: A configuration $c \in T^{\mathbb{Z}^2}$ is valid at position $\vec{n} \in \mathbb{Z}^2$ if and only if

$$[c(\vec{n} + \vec{n}_1), c(\vec{n} + \vec{n}_2), \ldots, c(\vec{n} + \vec{n}_m)] \in R,$$

that is, the neighborhood of $\vec{n}$ contains an allowed pattern. Configuration $c$ is a valid tiling iff it is valid at at all positions $\vec{n} \in \mathbb{Z}^2$.

Note that the complement of $R$ is the set of forbidden patterns: any configuration that contains such a pattern is not a valid tiling. The triplet $(T, N, R)$ specifies valid tilings, and we call such a triplet a *finite system of forbidden patterns*.

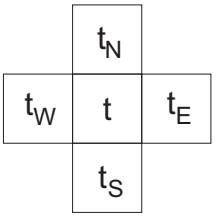
Any Wang tile set can be expressed as a finite system of forbidden patterns with the neighborhood

$$N = [(0, 0), (0, -1), (0, 1), (-1, 0), (1, 0)],$$

and the relation $R$ that contains all those patterns where the colors of the tiles match:

$$(t, t_S, t_N, t_W, t_E) \in R$$

if and only if the colors match between $t$ and the neighboring tiles in




There is also a correspondence to the other direction: for any finite system of forbidden patterns we can effectively (that is, algorithmically) construct a Wang tile set such that there is a natural correspondence between valid tilings in the two tile systems.

**Lemma 5.5**

*(i) For every Wang protoset $\mathcal{P}$ one can effectively construct a finite system $S = (\mathcal{P}, N, R)$ of forbidden patterns over $\mathcal{P}$ such that $c \in \mathcal{P}^{\mathbb{Z}^2}$ is a valid Wang tiling if and only if $c$ is valid according to $S$.*

*(ii) Conversely, for every finite system $S = (T, N, R)$ of forbidden patterns one can effectively construct a Wang protoset $\mathcal{P}$ such that $\mathcal{P}$ admits a (periodic) tiling if and only if $S$ admits a (periodic) tiling.*

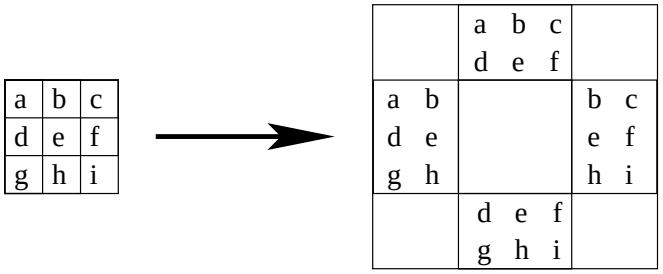
*Proof.* Part (i) is clear and was already explained above. Consider then (ii). Let $S = (T, N, R)$ be a finite system of forbidden patterns. Observe first that we can assume, without loss of generality, that elements of $N$ form a square: Let $m \geq 2$ be an integer such that there is an $m \times m$ square $M$ containing all elements $\vec{n}_i$ of the neighborhood vector $N$. We can add to $N$ the missing elements of $M$ – keeping them irrelevant in the relation $R$ – and obtain a system that admits exactly same valid tilings and whose neighborhood vector consists exactly of the elements of $M$.

Assuming now that $N$ forms an $m \times m$ square, we construct a set $\mathcal{P}$ of Wang tiles as follows: The tiles are the allowed $m \times m$ square patterns over $T$, that is, $\mathcal{P} = R$. Colors of $t \in R$ are obtained by erasing one boundary column or row from it: if $t = [t_{ij}]_{1 \leq i \leq m}^{1 \leq j \leq m}$ is a tile, where each $t_{ij} \in T$, then the left, top, right and bottom colors of $t$ are

$$[t_{ij}]_{1 \leq i \leq m-1}^{1 \leq j \leq m}, \qquad [t_{ij}]_{1 \leq i \leq m}^{2 \leq j \leq m}, \qquad [t_{ij}]_{2 \leq i \leq m}^{1 \leq j \leq m}, \qquad [t_{ij}]_{1 \leq i \leq m}^{1 \leq j \leq m-1},$$

respectively. So the vertical colors are $(m-1) \times m$ blocks and horizontal colors are $m \times (m-1)$ blocks over $T$.

For example, the following figure illustrates the tile corresponding to a $3 \times 3$ allowed pattern:




Note that two adjacent Wang tiles then match if and only if the $m \times m$ patterns they represent have the correct $(m-1) \times m$ or $m \times (m-1)$ overlap when the tiles are placed next to each other.

This construction immediately implies that, if $f \in T^{\mathbb{Z}^2}$ does not contain any forbidden pattern, then the function $g \in \mathcal{P}^{\mathbb{Z}^2}$ is a correct Wang tiling, where for each $(i, j) \in \mathbb{Z}^2$ we set $g(i, j)$ be the $m \times m$ pattern in $f$ whose lower left corner is in position $(i, j)$.

Conversely, let $g \in \mathcal{P}^{\mathbb{Z}^2}$ be a valid Wang tiling. Let $f \in T^{\mathbb{Z}^2}$ be the function such that $f(i, j)$ is the symbol at the lower left corner of the $m \times m$ square $g(i, j)$, for every $(i, j) \in \mathbb{Z}^2$. Because of the overlap property between neighboring Wang tiles, we have that the $m \times m$ blocks extracted from $f$ coincide with the corresponding tiles in $g$. Because the elements of $\mathcal{P}$ are the allowed $m \times m$ patterns we have that $f$ only contains allowed $m \times m$ blocks, and therefore $f$ is a valid tiling.

Note that in the previous reasoning $f$ is periodic if and only if $g$ is periodic.

□

The previous lemma means that in the following decision problems we can describe tiles in any terms that locally determine which tiles are allowed to be next to each other. Such tiles can anyway be effectively converted into an equivalent set of Wang tiles. This substantially simplifies the discussion.

## 5.4 The periodic tiling problem

Next we consider the problem of deciding if a given protoset admits a periodic tiling. There is an obvious semi-algorithm:

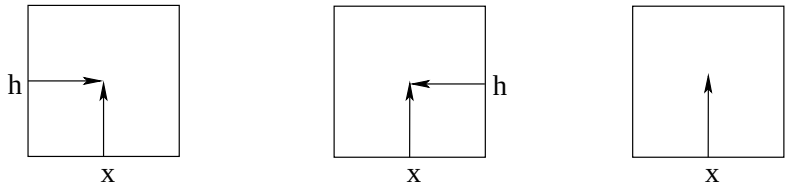
**Lemma 5.6** *The decision problem "Does a given finite set $\mathcal{P}$ of Wang prototiles admit a periodic tiling ?" is semi-decidable.*

*Proof.* The semi-algorithm enumerates positive integers $n = 1, 2, 3, \ldots$ one-by-one, and for each $n$ it constructs all valid tilings of the $n \times n$ square. There are only a finite number of them. For each such tiling, we check if the top and the bottom of the square read the same colors, and if the left and the right side also read the same colors. If they do, we have found a square pattern that can be repeated to form a periodic tiling of the plane. The semi-algorithm returns then "yes". We know that if a periodic tiling exists then some $n \times n$ square forms a period of a periodic tiling, so our semi-algorithm is guaranteed to correctly detect all "yes"-instances. □

Note that if no aperiodic tile sets existed then there would be an algorithm for determining if a periodic tiling exists: We namely have semi-algorithms for detecting that no tiling exists (Lemma 5.1) and that a periodic tiling exists (Lemma 5.6). If these were the only two possibilities then the two semi-algorithms would yield an algorithm (Theorem 5.2) that would determine if a (periodic) tiling exists. However, since aperiodic protosets do exist, this reasoning can not be used. In fact, the admittance of periodic tilings turns out to be undecidable. Not surprisingly, an aperiodic protoset is needed in the proof.

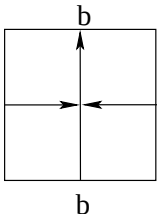
**Theorem 5.7** *The following decision problem is semi-decidable but not decidable: "Does a given finite set $\mathcal{P}$ of Wang prototiles admit a periodic tiling ?"*

*Proof.* Semi-decidability was discussed above in Lemma 5.6. Let us prove undecidability via a reduction from the halting problem of Turing machines. For any given Turing machine $M$, we construct the Wang set $\mathcal{P}_M$ from Theorem 5.4. We add to these the following halting tiles for every tape letter $x \in \Gamma$:



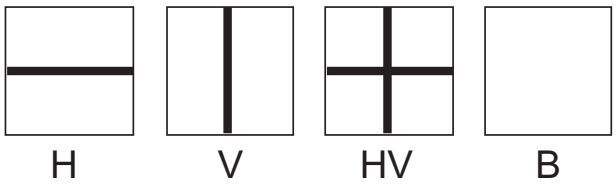

Here $h$ is the halting state. The halting tiles have the effect that a tiling becomes possible even if the Turing machine halts — then the state component simply disappears from the configuration. Using the third tile, the entire configuration can then disappear.

We also add the following tile

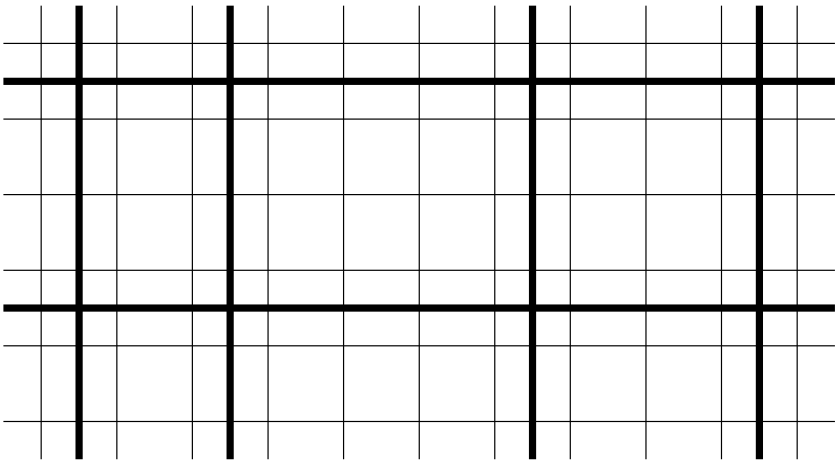


to the start tiles of the Turing machine. This tile allows the same horizontal row to contain several copies of the start configuration of the Turing machine.

In addition to $\mathcal{P}_M$, we take one aperiodic tile set $\mathcal{P}$. This can be, for example, the Robinson's aperiodic tile set. Finally, we also use the following set $\mathcal{Q}$ of prototiles: Set $\mathcal{Q}$ contains tiles with horizontal and/or vertical fault lines, as well as tiles without a fault line:




There are no tiles to end a fault line, so all fault lines cut the plane horizontally or vertically:



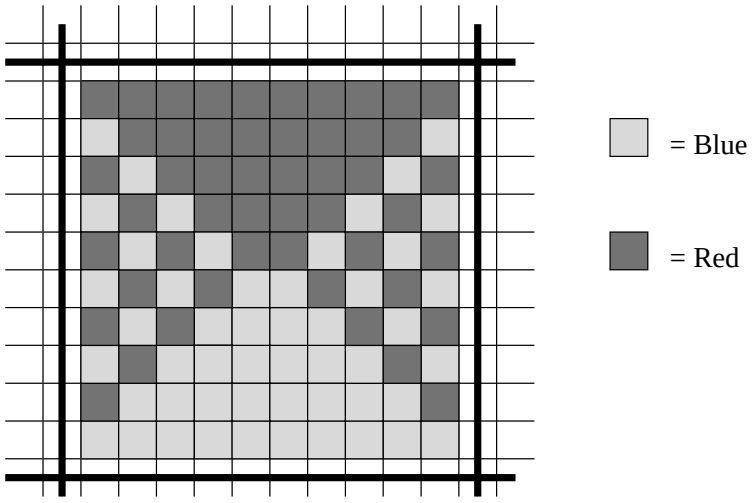
We want the fault line tiles to satisfy the following property:

(*) If a tiling contains at least two parallel fault lines then it also contains at least two fault lines in the perpendicular direction.

To establish this, we make two versions of the empty tile $B$ without fault lines: one called red and the other one called blue. Then we add the following local constraints on validity of tilings:

- The northern neighbor of a horizontal fault line (tile H) must be a blue blank (=blue version of tile B).
- The southern neighbor of a horizontal fault line (tile H) must be a red blank (=red version of tile B).
- The northern neighbor of a blue blank whose horizontal neighbors are blue blanks is a blue blank.
- The southern neighbor of a red blank whose horizontal neighbors are red blanks is a red blank.

These local constraints are satisfied in tilings where the fault lines partition the plane into squares of even size. The insides of the squares consist of a blue and a red triangle and two triangles with the checker-board pattern of blue and red:




Notice that the constraints are local and can be implemented using a finite system of forbidden patterns.

Suppose that a tiling contains two horizontal fault lines at distance $n$ from each other. Suppose there would be a horizontal segment of length $2n$ without a vertical fault line. Then there is a blue segment of length $2n$ on top of the lower fault line. On top of it we have $2n-2$ blue tiles, then $2n-4$ blue tiles, and so on. We see that the horizontal row below the upper fault line must contain blue tiles, which is not allowed by the local constraints above. Conclusion: if a tiling contains at least two horizontal fault lines then it also contains at least two vertical fault lines.

An analogous coloring is also done in the perpendicular direction. Hence the non-fault line tiles come if four varieties, one for each combination of blue/red color in horizontal/vertical direction. Then our set $\mathcal{Q}$ satisfies the required property (*).

After establishing the three tile sets $\mathcal{P}_M$ (simulates the Turing machine), $\mathcal{P}$ (aperiodic set) and $\mathcal{Q}$ (fault lines), the algorithm combines these three sets into a single tile set by taking their cartesian product

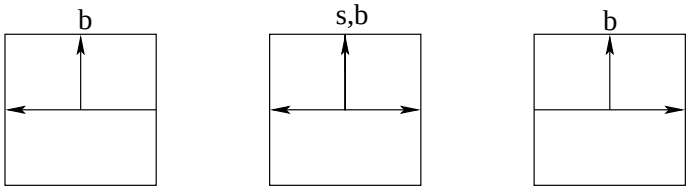
$$\mathcal{P}_M \times \mathcal{P} \times \mathcal{Q}.$$

Each of the three components of any tile must match locally with its neighbors according to the rules of the corresponding tile set. In this way tilings will be "sandwiches" with three layers. For any $(a,b,c) \in \mathcal{P}_M \times \mathcal{P} \times \mathcal{Q}$ we call $a$, $b$ and $c$ the first, the second and the third layer, respectively.

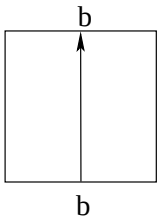
We add the following local constraints on tilings:

(1) In tile $(a,b,c)$, if $c$ contains a fault line then the tiling rule is not enforced on the second layer $b$. The idea is to allow periodic tilings (even though $\mathcal{P}$ is aperiodic) in the presence of fault lines.

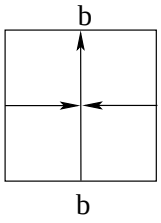
(2) In tile $(a,b,c)$, if $c$ contains only the horizontal fault line then the first component $a$ must be one of the start tiles




(3) In tile $(a,b,c)$, if $c$ contains only the vertical fault line then the first component $a$ must be



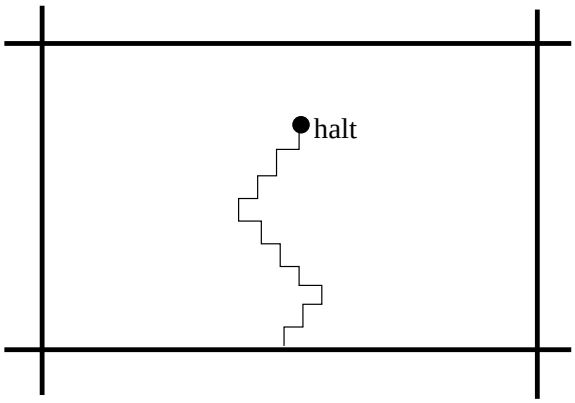

(4) In tile $(a, b, c)$, if $c$ contains both horizontal and vertical fault lines then $a$ must be




These constraints force the lower border of any rectangle surrounded by fault lines to contain (a finite segment of) the blank tape and a single Turing machine in its initial state $s$. The vertical fault lines are forced to contain the black symbol only, and the Turing machine is never allowed to reach a vertical fault line.

The construction of the tile set is now complete. Suppose first that Turing machine $M$ halts in $n$ steps. Then the tiles admit a valid periodic tiling with the horizontal and vertical period $2n$. On the third layer the fault lines partition the space into squares of size $2n \times 2n$. The second layer contains a correctly tiled $2n \times 2n$ square, repeated inside the squares between the fault lines. The tiling of the second layer fails on some tiles along the fault lines, but that is allowed by (1) above.

The first layer consists of the halting simulation of the Turing machine $M$. The start of the simulation begins at the bottom of each $2n \times 2n$ square. The entire simulation fits inside the $2n \times 2n$ square, because the machine halts after $n$ steps. The halting tiles allow the disappearance of the Turing machine configuration before the next vertical line is reached. Hence a periodic tiling is admitted.




Conversely, suppose a periodic tiling exists. Because $\mathcal{P}$ is an aperiodic set, there must be a place on the tiling where the tiling in the second layer is incorrect. This is possible only if there is a fault line in that location. Because the tiling is periodic, this implies the existence of infinitely many parallel fault lines. By property (*) this further implies the presence of a rectangle bordered by fault lines. Consider the first

layer of one such rectangle. The bottom is forced to contain a (finite segment) of the start configuration of the Turing machine. The tiles in $\mathcal{P}_M$ force the following rows to simulate the Turing machine moves one-by-one. If the Turing machine does not halt then the simulation continues without a limit and the Turing machine never disappears. But the next horizontal fault line is possible only if the Turing machine disappears. Hence the Turing machine must halt before the simulation reaches the upper border of the rectangle.

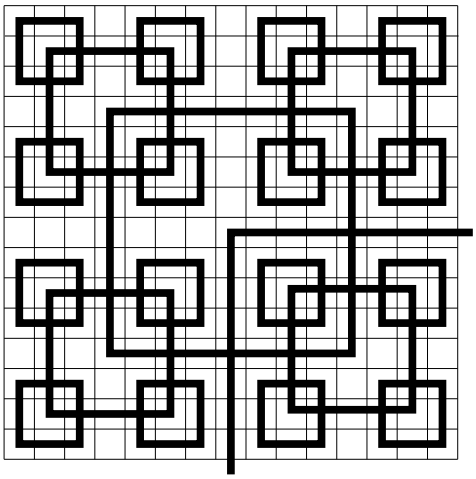
The tile set we constructed is given as input to our hypothetical algorithm that determines if the set admits a periodic tiling. (More precisely, the input is the corresponding set of Wang tiles, obtained by the conversion from the system of forbidden blocks as described in Lemma 5.5). We know that a periodic tiling is possible if and only if $M$ halts, so we get the answer to the halting problem, a contradiction.

□

## 5.5 The tiling problem

Now we turn to the general tiling problem: "Does a given Wang tile set admit a valid tiling?" To prove undecidability we make a reduction from the tiling problem with a seed tile, proved undecidable in Theorem 5.4. Now we have no specified seed tile required to be used, so the main problem is how to force the presence of the seed (=the beginning state of the Turing machine) in every valid tiling. Note that if it is possible to have arbitrarily large squares without the seed, then it is also possible to make the entire tiling without the seed. This is a consequence of the compactness of the tiling space (Corollary 4.3). Therefore the seed must be enforced inside all $n \times n$ squares for some $n$. This on the other hand would seem to be contradictory to the possibility that tilings with only a single seed tile may occur. (Indeed, in our proof of Theorem 5.4 only a single seed tile occurs, starting an infinite, non-halting computation of a Turing machine.) A solution is to partition the space using Robinson's aperiodic tile set into "nested boards", each containing a copy of piece of a valid tiling around a seed tile.

This takes us back to the Robinson's tile set. Recall the special $(2^n - 1)$-squares that necessarily exist in every valid tiling. We define nested boards using the side arrows of Robinson's tiles. The following figure shows only the side arrows of a $(2^n - 1)$-square:



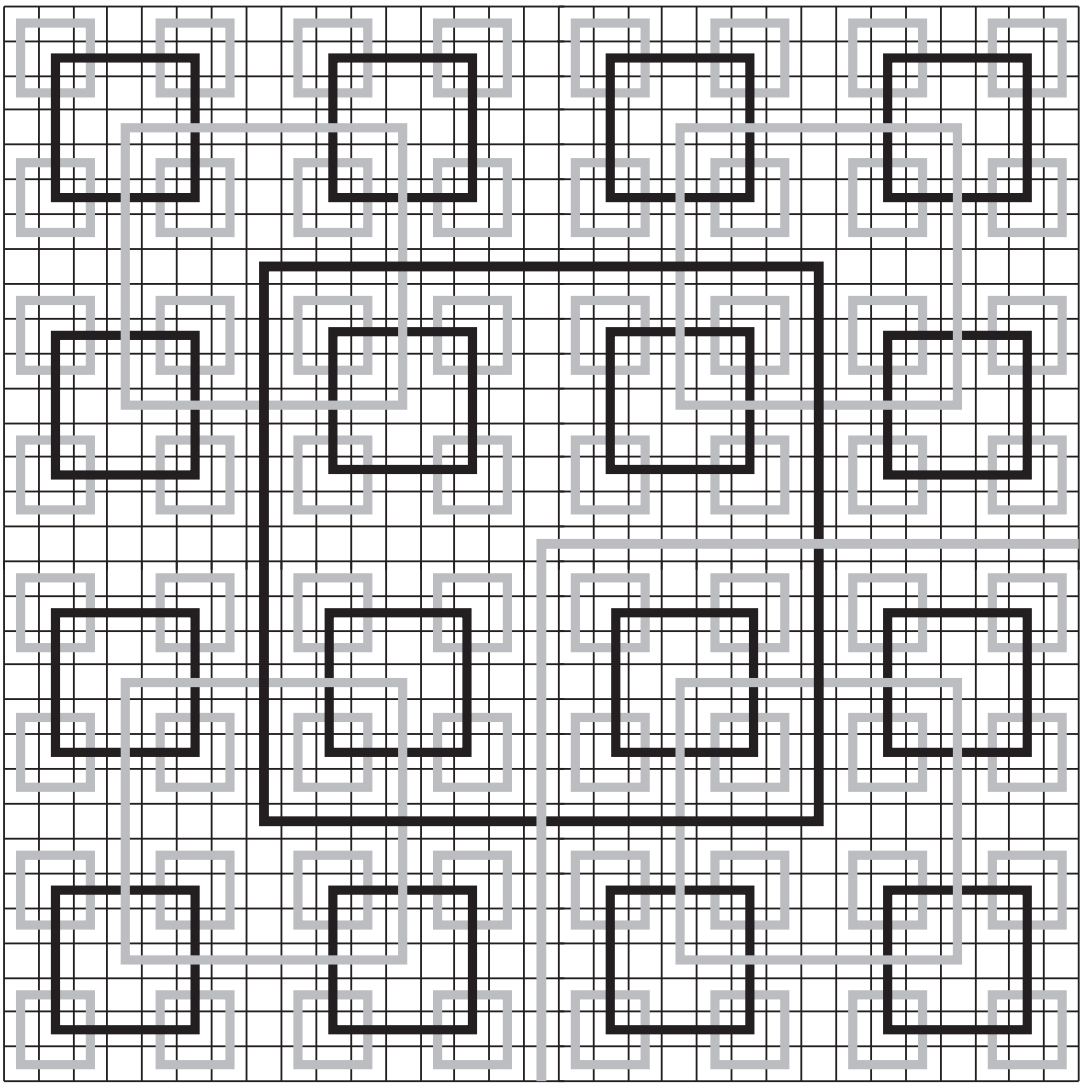
Notice that the side arrows form overlapping squares: The side arrows emitted from the crosses form squares whose centers contain crosses, which in turn are corners of bigger squares. The smallest squares have the corners at the odd-odd -positions. They are of size $2 \times 2$, and they only intersect one $4 \times 4$ square whose corner is at the center. Any other square $S$ is of size $2^n \times 2^n$, for $n \geq 2$, and it intersects

one bigger square of size $2^{n+1} \times 2^{n+1}$ whose corner is at the center of $S$, and four smaller squares of sizes $2^{n-1} \times 2^{n-1}$ whose centers are the corners of of $S$.

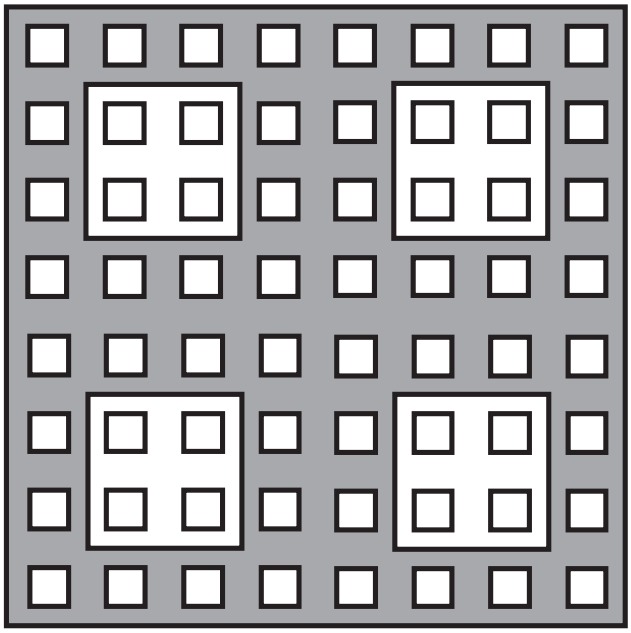
In order to pick non-overlapping squares we color the side arrows red or green according to the following rules:

- The side arrows of each cross are both red or both green. The crosses at odd-odd positions have green side arrows.
- In each arm the horizontal side arrows have the same color and the vertical side arrows have the same color. In this way the color is transmitted unchanged through the arm. If the arm contains both horizontal and vertical side arrows then these side arrows have different colors.
- In neighboring tiles the matching rule is that the meeting arrow heads and tails must have the same color.

Following these rules, each square will be colored completely red or green, and intersecting squares have opposite colors. The smallest squares are green, so the coloring of the squares is completely determined. Notice that red squares do not intersect each other and green squares do not intersect each other. The red squares are of sizes $4^n \times 4^n$ for $n = 1, 2, \ldots$. In the following figure green and red borders are indicated light and dark, respectively.



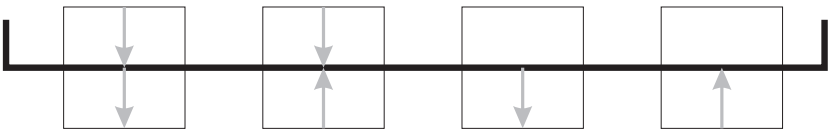
A piece of a Wang tiling with a seed tile will be enforced inside each red square. Small red squares nested within a larger red square contain their own copies. We call a region within a red border but outside all nested red borders within it a *board*. Here is the board with side $4^3 = 64$:



Next we need to identify those rows and columns of a board that run completely across the board without intersecting a smaller board inside. Let us call these *free* rows and columns. Let $F_n$ be the number of free rows (and columns) in a board of side $4^n$. In the middle the board of side $4^{n+1}$ we have the same smaller boards as in the middle of the $4^n$ board, and at the sides we have halves of those same boards. The boundary of the $4^{n+1}$ square occupies one row, so the total number of free rows in the $4^{n+1}$ board is $F_{n+1} = 2F_n - 1$. Since $F_1 = 3$, we easily obtain $F_n = 2^n + 1$. Hence a valid tiling necessarily contains boards with arbitrarily large numbers of free rows and columns.

To identify tiles of the board that are on a free row and/or a free column, we use a new set of arrows, called *obstruction signals*. There are vertical and horizontal obstruction signals, used to identify free columns and rows, respectively. An outer edge of a red border must emit or absorb an obstruction signal, whereas the inner edges of the red borders may absorb but not emit such a signal. Inner edges can also be without an obstruction signal.

Here are the four possible combinations of obstruction signals on the lower boundary of a red square:

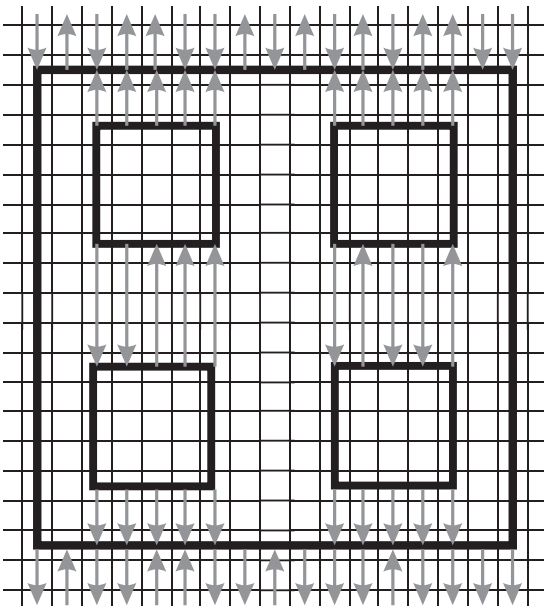


The other boundaries (top, left, right) are analogous. Note that the position of a red side arrow identifies whether it belongs to the top, bottom, left or right boundary of a red square.

Obstruction signals are transmitted unchanged through tiles that are not on the boundary of a board. Then no free column can contain a vertical obstruction signal and no free row can contain a horizontal obstruction signal, as such a signal would have to be emitted from the inside edge of the boundary. In contrast, any interior tile of a board that is not on a free column is either between the inner edge of the board and the outer edge of a smaller board, or between outer edges of two smaller boards. In either

case, there is a vertical obstruction signal at the tile. Analogously, any tile that is not on a free row must contain a horizontal obstruction signal. We conclude that the horizontal and vertical obstruction signals correctly identify the free rows and columns of the boards.

The following figure shows a possible arrangement of vertical obstruction signals on a $16 \times 16$ board:



Based on the presence or absence of horizontal/vertical obstruction signals, tiles inside a board can be classified into four classes:

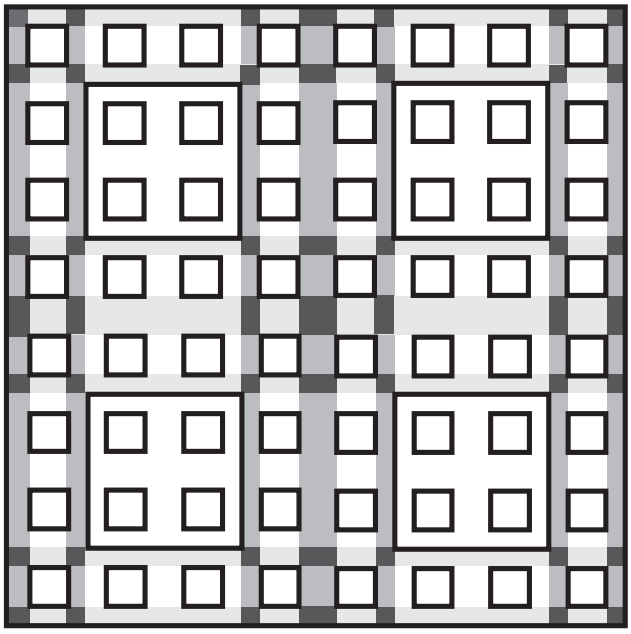
(00) with horizontal and with vertical obstruction signal,

(01) with horizontal and without vertical obstruction signal,

(10) without horizontal and with vertical obstruction signal,

(11) without any kind of obstruction signal.

Tiles of type (11) are free, and they form a scattered $(2^n + 1) \times (2^n + 1)$ -square, whose disjoint parts are connected by tiles of types (01) and (10):

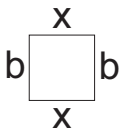


Let $\mathcal{R}$ be the tile set constructed above. Now we are ready to reduce the tiling problem with the seed tile into the tiling problem without a seed tile: Let $\mathcal{P}$ be a given set of Wang tiles, and let $s \in \mathcal{P}$ be the given seed tile. Let $C$ be the set of colors used in $\mathcal{P}$. To determine if $\mathcal{P}$ admits a tiling that contains a copy of $s$ we construct a set $\mathcal{X}$ of "sandwich tiles" $(r, p)$, whose first component $r \in \mathcal{R}$, and the second component $p$ is a Wang tile over the color set $C \cup \{b\}$, where $b \notin C$ is a new "blank color".

The first components tile according to the local matching constraints of $\mathcal{R}$ described above. The second components tile under the color constraints, as in Wang tiles. The set $\mathcal{X}$ contains all the pairs $(r, p)$ that satisfy the following:

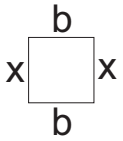
(a) If $r$ is a free tile (no obstruction signal present in $r$) then $p \in \mathcal{P}$.

(b) If $r$ is on a free column but not on a free row then $p$ is a tile




where $x \in C$ and $b$ is the blank color.

(c) If $r$ is on a free row but not on a free column then $p$ is a tile




where $x \in C$ and $b$ is the blank color.

(d) If $r$ is not on a free row or column then $p$ is arbitrary: any element of $(C \cup \{b\})^4$ is acceptable.

(e) If $r$ is a corner of a green square of size $> 2 \times 2$ (that is, a cross with green side arrows in an even-even position), then $p = s$, the seed tile.

Notice that the corners of green squares that are in even-even positions are exactly at the centers of red squares. These center positions are always free.

Condition (e) guarantees that the center of each board is paired with the seed tile $s$. Properties (b) and (c) mean that color information is transmitted along free rows and columns between disjoint parts of the board, while (a) guarantees that on free areas a tiling by $\mathcal{P}$ is formed. These conditions make the board behave as if the free rows and columns were contiguous and the board then is like a square board of side $2^n + 1$. Condition (d) simply allows different boards to be joined arbitrarily.

Let us prove that our sandwich tiles admit a tiling if and only if $\mathcal{P}$ admits a tiling that contains a copy of the seed tile $s$:

$\Longleftarrow$ Suppose first that $\mathcal{P}$ admits a tiling that contains $s$. Then we can properly tile any board by placing $s$ at the center and scatter the proper tiling containing $s$ in the free areas. Smaller nested boards can be tiled in the same way. Different boards are not immediate neighbors of each other since there are at least the red boundary tiles between them. So different boards can be tiled independently of each other. As we can tile arbitrarily large squares in this way, the whole plane can be tiles as well.

$\Longrightarrow$ For the converse direction, suppose that the sandwich tiles admit a tiling. The underlying Robinson's tiling necessarily contains special $(2^n - 1)$ -squares for arbitrarily large number $n$. Hence there are red squares of size $4^n \times 4^n$, for every $n$, and consequently there are arbitrarily large boards. The center of each board is paired with $s$, and the free areas of the board necessarily contain a piece of a valid tiling by $\mathcal{P}$. As the free area is arbitrarily large, and its center contains $s$, we conclude that $\mathcal{P}$ admits a tiling that contains a copy of tile $s$.

We have proved the following theorem by Berger. The proof presented here is from 1971 by Robinson.

**Theorem 5.8 (Berger 1966)** *The tiling problem "Does a given Wang protoset admit a valid tiling ?" is undecidable.* □

Each finite protoset is exactly one of the following types:

1. protosets that do not admit any tilings,
2. protosets that admit some periodic tilings,
3. aperiodic protoset

Membership in the first two classes are known to be semi-decidable (Lemmas 5.1 and 5.6). The undecidability of the tiling problem implies that membership in the third class cannot be semi-decidable, so there is no semi-algorithm to determine if a given protoset is aperiodic. Membership in the union of classes 2 and 3 is not semi-decidable (Theorem 5.8), and the membership in the union of classes 1 and 3 is not semi-decidable (Theorem 5.7). Hence we have been able to determine the semi-decidability status for all combinations of the three classes above.

## 5.6 The completion problem

Consider next the following decision problem: Is a given finite pattern part of some valid tiling ? We call it the completion problem. More precisely, a finite pattern over the tile set $T$ is a pair $(D, p)$ where $D \subseteq \mathbb{Z}^2$ is a finite domain, and $p : D \longrightarrow T$ assigns tiles to the cells in the domain. We say that $(D, p)$ is a subpattern of configuration $c \in T^{\mathbb{Z}^2}$ if $c(\vec{n}) = p(\vec{n})$ for all $\vec{n} \in D$. We want to determine for a given $(D, p)$ whether there exists a valid tiling that contains it as a subpattern.

The completion problem is clearly undecidable if the input contains both the tile set and the pattern (as the tiling problem with a seed tile is a particular case of this). But it turns out that the completion problem is undecidable already for some fixed tile sets. In this case only the pattern $(D, p)$ is the input. A tile set with undecidable completion problem can be constructed from any Turing machine with

undecidable halting problem: here we consider computations of Turing machines from initial tapes that are not necessarily blank. Instead, we assume that there may be finitely many tape locations that have a non-blank symbol. So let us call *b-finite* a tape content $f : \mathbb{Z} \longrightarrow \Gamma$ such that the set $\{i \in \mathbb{Z} \mid f(i) \neq b\}$ is finite.

**Theorem 5.9** *There exists a Turing machine $U = (S, \Gamma, \delta, s, h, b)$ such that the following decision problem is undecidable: "Given b-finite $f : \mathbb{Z} \longrightarrow \Gamma$, does $U$ reach the halting state $h$ from the initial configuration $(s, 0, f)$ ?"*

*Proof.* (sketch) The machine $U$ we construct is a universal Turing machine: it is able to simulate any other Turing machine if a description of that machine is initially written on the tape. Then if we could determine whether $U$ halts from a given initial tape then we could also determine for any given Turing machine whether it halts from the empty input tape.

First we observe that the halting problem from the empty tape is undecidable even among Turing machines with the binary tape alphabet $\Gamma = \{a, b\}$, where $b$ is the blank symbol:

**Lemma 5.10** *It is undecidable if a given Turing machine with binary tape alphabet eventually halts from the empty input tape.*

*Proof.* (sketch) For any given Turing machine $M$ with $k$ tape letters $1, 2, \ldots, k$, we effectively construct machine $M'$ that has tape alphabet $\{0, 1\}$ and that halts from the empty input tape if and only if $M$ halts from the empty input tape. On the tape of machine $M'$ a binary encoding of the $k$ letter alphabet is done using blocks of $n = \lceil \log_2(k) \rceil$ bits. The code for the blank tape letter of $M$ is $00 \ldots 0$.

At all times, $M'$ memorizes the current state $q$ of machine $M$. To simulate one instruction of $M$, the machine $M'$ does the following:

- It reads and memorizes the next block of $n$ bits from the tape: this gives the current tape letter $x$ of $M$.
- Let $\delta(q, x) = (p, y, d)$.
- Machine $M'$ memorizes the new state $p$, writes the $n$ bits representing $y$ over the block that it just read, and moves $n$ positions left or right on the tape, depending on the value of $d$.

This is the simulation loop for one step of $M$. The initial and halting states of $M'$ are the first states of the simulation loop, with the initial and halting state of $M$ being memorized, respectively.

It is clear that when started on the empty tape (0 is the blank), machine $M'$ simulates $M$ until, if ever, it halts. □

Based on the lemma, it is enough for our universal Turing machine $U$ to simulate those Turing machines that have the tape alphabet $\{a, b\}$. The tape of $U$ will contain five tracks:

1. On the first track, the tape of the simulated machine is written. This track has alphabet $\{a, b\}$.
2. The second track stores the state of the simulated machine. We may assume, without loss of generality, that the states of $M$ are numbers $1, 2, \ldots, k$, where 1 is the initial state and $k$ is the halting state. Current state $i$ is expressed on track two as a segment of $i$ symbols \$, starting at position zero of the tape. Outside the segment the track contains the blank $b$, so the track alphabet is $\{\$, b\}$.
3. The third track stores the position of $M$ on its tape during the simulation. The track simply contains $\uparrow$ at cell $i$ if machine $M$ currently reads that cell. Other cells are blank. This track alphabet is $\{\uparrow, b\}$.

4. The fourth track stores the "program", i.e. the transition function $\delta$ of the simulated machine $M$. Each transition $\delta(i,x) = (j,y,d)$ is expressed as the word $\$^j\ y\ d$ where $y$ is either $a$ or $b$, and $d$ is either $L$ or $R$. For example, transition $\delta(i,x) = (4,a,L)$ is represented as $\$\$\$\$aL$. We represent the transition table $\delta$ as the word

$$\#\ E(1,a)\ \&\ E(1,b)\ \#\ E(2,a)\ \&\ E(2,b)\ \#\ \ldots\ \#\ E(k-1,a)\ \&\ E(k-1,b)\ \#$$

where $E(i,x)$ is the word representing $\delta(i,x)$. Note that the transitions from the halting state $k$ are not written. The alphabet of track four is $\{\#, \&, \$, L, R, a, b, B\}$. (We use $B$ for the blank.)

5. The fifth track is a "scratch pad" where temporary information is stored during the simulation.

As an example of the encoding of the program, consider the three-state Turing machine of Example 9, where we number the states as follows: $s = 1, t = 2, h = 3$. The corresponding program in our universal machine is

$$\#\ \$\$aL\ \&\ \$\$aR\ \#\ \$\$\$aL\ \&\ \$aL\ \#$$

The tape at the beginning of the simulation will be

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | $ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | ↑ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | # | $ | $ | a | L | & | $ | $ | a | R | # | $ | $ | $ | a | L | & | $ | a | L | # | | | | | | | | | | | | | |

To simulate one step of $M$, the universal machine memorizes the current tape symbol scanned by $M$, as well as whether the position being scanned is positive or negative (so that ↑ can be easily found). Then it performs the following operations:

- Using the scratch pad (track 5) to store check markers, the machine matches the symbols $ of track 2 and the symbols # of track 4 starting from the left. This way the machine finds the $i$'th marker # from the program where $i$ is the state of the simulated machine.
- If the symbol on track four that follows the $i$'th delimiter # is a blank then the machine halts. The simulated machine has namely entered the halting state $k$.
- If the tape letter being scanned by $M$ is $b$ then the machine finds the next & symbol on track 4. Now the machine is at the beginning of $E(i,x)$ for the next instruction to be executed.
- Using the scratch pad (track 5) the machine copies the signs $ from the word $E(i,x)$ on track 4 to track 2.
- The machine memorizes the the next two symbols $y$ and $d$ of the instruction. The machine finds the symbol ↑ on track 3, writes the symbol $y$ in that location on track 1 and moves ↑ on track 3 one position left or right, depending on the value of $d$. The tape letter in the new location is memorized.
- The machine clears the scratch pad, and moves the position 0, ready to simulate the next step.

For example, after the first simulation round the tape in our sample machine will be

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | $ | $ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | ↑ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | # | $ | $ | a | L | & | $ | $ | a | R | # | $ | $ | $ | a | L | & | $ | a | L | # | | | | | | | | | | | | | |

It is clear that each simulation loop of $U$ sketched above properly simulates one step of $M$. If the simulated machine $M$ enters the halting state, then also $U$ halts.

If we had an algorithm to determine whether $U$ halts when started on a given finite initial tape, then we could use this algorithm to determine if any given Turing machine $M$ halts form the empty tape. Indeed, we construct the initial configuration where the program track four contains the description of $M$ and ask whether $U$ halts from this configuration. □

The universal Turing machine sketched in the proof above contains many states and and has a large tape alphabet. Smaller universal machines have been discovered.

**Example 11.** The following universal Turing machine (due to Y.Rogozhin) with $4 + 1$ states and 6 tape symbols has an undecidable halting problem, i.e. it satisfies the condition of the previous theorem: $M = (\{q_1, q_2, q_3, q_4, h\}, \{1, b, >, <, 0, c\}, \delta, q_1, h, b)$, and $\delta$ is given by the following table

| | $q_1$ | $q_2$ | $q_3$ | $q_4$ |
|---|---|---|---|---|
| 1 | $(q_1, <, L)$ | $(q_2, 0, R)$ | $(q_3, 1, R)$ | $(q_4, 0, R)$ |
| $b$ | $(q_1, >, R)$ | $(q_3, >, L)$ | $(q_4, <, R)$ | $(q_2, c, L)$ |
| $>$ | $(q_1, b, L)$ | $(q_2, <, R)$ | $(q_3, b, R)$ | $(q_4, <, R)$ |
| $<$ | $(q_1, 0, R)$ | $(q_2, >, L)$ | $h$ | $h$ |
| 0 | $(q_1, <, L)$ | $(q_2, , 1, L)$ | $(q_1, c, R)$ | $(q_2, c, L)$ |
| $c$ | $(q_4, 0, R)$ | $(q_2, b, R)$ | $(q_1, 1, R)$ | $(q_4, b, R)$ |

where the item on column $q$, row $x$ is $\delta(q, x)$. □

Modifying slightly the Turing machine tile construction in Section 5.2 we can now easily obtain the following:

**Theorem 5.11** *There exists a finite set $\mathcal{P}$ of Wang prototiles such that the following problem is undecidable: "Is a given finite pattern a subpattern of some valid tiling ?"*

*Proof.* In the homework assignments.

□

## 5.7 Beyond aperiodicity: arecursive tile sets

Robinson's tile set is aperiodic: no periodic tilings exist. Still, valid tilings that are "simple" exist. Using the special $2^n - 1$ -squares one can effectively construct bigger and bigger portions of a fixed valid tiling.

But there exist tile sets that only admit very complicated tilings: namely tilings that cannot be algorithmically constructed. We call a tiling $c : \mathbb{Z}^2 \longrightarrow T$ *recursive* if there exists an algorithm that outputs $c(i, j)$, when given arbitrary integers $i, j$ as input. If no such algorithm exists then $c$ is called *non-recursive*. Analogously, a tape content $f : \mathbb{Z} \longrightarrow \Gamma$ of a Turing machine is recursive if there exists an algorithm that returns $f(i)$ for any given input $i \in \mathbb{Z}$. Otherwise $f$ is non-recursive.

Clearly any two-way periodic tiling is recursive. Some non-periodic tilings are recursive, too: for example Robinson's tile set admits a recursive, non-periodic tiling. Analogously to aperiodicity, we define the concept of *arecursivity* as follows: Wang tile set $\mathcal{P}$ is arecursive if and only if

(i) it admits valid tilings, and

(ii) it does not admit any recursive valid tilings.

Clearly any arecursive tile set is aperiodic, but the converse is not true since Robinson's tile set is not arecursive. So arecursivity is a stronger property than aperiodicity.

Arecursive tile sets exist. First, let us sketch a proof that there exist Turing machines $M_R = (S, \Gamma, \delta, s_0, s_h, b)$ with the following behavior:

(i) For every recursive $f : \mathbb{Z} \longrightarrow \Gamma$, machine $M_R$ halts from the initial configuration $(s_0, 0, f)$, but

(ii) there exists a (non-recursive) tape content $f : \mathbb{Z} \longrightarrow \Gamma$ such that $M_R$ does not halt from the initial configuration $(s_0, 0, f)$.

To construct such $M_R$ we consider the problem of determining which of two halting states does a given Turing machine halt. In this problem we are given a TM $M$ with two halting states $h_1$ and $h_2$. We are looking for an algorithm that returns answer 1 or 2 if $M$ halts in state $h_1$ or $h_2$, respectively, when started on the blank tape. If $M$ does not halt then the algorithm can (and must) return either answer 1 or answer 2. That such an algorithm can not exist can be proved using a diagonal argument (which we skip), similar to Turing's proof for the undecidability of the halting problem:

**Lemma 5.12** *There is no algorithm that, for any given TM $M$ with two halting states $h_1, h_2 \in S$,*

- *always returns an answer "1" or "2", and*
- *returns answer "1" ("2",respectively) if $M$ halts in state $h_1$ (or $h_2$, respectively) when started on the blank tape.*

We say that the set $A$ of Turing machines that halt in state $h_1$ is *recursively inseparable* from the set $B$ of Turing machines that halt in state $h_2$. (More generally, disjoint sets $A$ and $B$ are recursively inseparable if there is no algorithm that returns an answer on every input $x$, and if $x \in A$ then the answer must be 1 and if $x \in B$ then the answer must be 2. In other terms: there is no decidable set $R$ such that $A \subseteq R$ and $B \cap R = \emptyset$.)

Now we can construct TM $M_R$ with several tracks on the tape:

- On the first track we have symbols of the alphabet $\{1, 2\}$. These symbols are not modified by $M_R$.
- On the second track the machine enumerates integers $n = 1, 2, 3, \ldots$ one-by-one.
- For each value $n$ on the second track, the machine then enumerates on the third track numbers $m = 1, 2, \ldots n$. Values of $m$ are interpreted as TM descriptions using, say, the encoding given in the proof of Theorem 5.9: Number $m$ is written in the number system with 7 digits, using the symbols of the alphabet $\Sigma = \{\#, \&, \$, L, R, a, b\}$ as the digits.
- If m is not a proper encoding of a TM transition rule (which can be easily checked) then nothing is done, but the machine simply moves on to the next value of $m$.
- But if $m$ represents some TM $M$ then a simulation of $M$ is started from the blank tape, using the universal machine from the proof of Theorem 5.9. The simulation is done only for $n$ steps, where $n$ is the number on the second track.
- If the simulated machine reaches state $h_1$ or $h_2$ (which we can fix to be the second and the third state of the machine, respectively) before $n$ simulation steps are executed then $M_R$ checks the symbol on the first track in position $m$: if the symbol is 1 but $M$ reached state $h_2$, or the symbol is 2 but $M$ reached state $h_1$, then $M_R$ halts. Otherwise it moves on to the next value of $m$.

The idea of $M_R$ is to simulate all Turing machines for arbitrarily long times, and to verify for each machine that the first track correctly identifies which state $h_1$ or $h_2$ is first reached in each machine. Machine $M_R$ halts from an initial configuration if and only if for some Turing machine $m$ the first track identifies in position $m$ incorrectly the state in which $m$ halts.

Suppose the initial tape content $f : \mathbb{Z} \longrightarrow \Gamma$ of machine $M_R$ is recursive. Let us prove that $M_R$ must halt. Suppose the contrary: $M_R$ does not halt. Then for any Turing machine $m$ we can effectively calculate the symbol on the first track in position $m$ of $f$. Because $M_R$ does not halt, this symbol must be 1 if $m$ halts in state $h_1$ and 2 if $m$ halts in state $h_2$. So we have described an algorithm that contradicts lemma 5.12. We conclude that $M_R$ halts when started on any recursive initial tape content.

On the other hand, if the first track is such that it correctly identifies which machines halt in state $h_1$ and $h_2$ then $M_R$ does not halt. So there are (non-recursive) initial tapes from which $M_R$ does not halt.

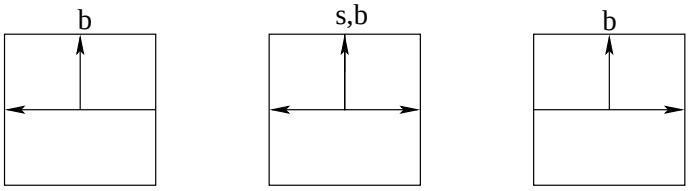
We have sketched a proof of the following lemma:

**Lemma 5.13** *There exists a Turing machine $M_R$ that halts when started on any recursive initial tape, but for some non-recursive initial tape $M_R$ does not halt.* □

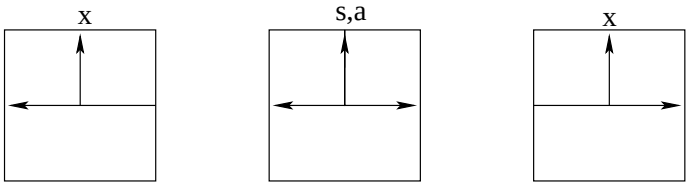
Based on machine $M_R$ we can now easily construct a Wang tile set with a seed tile such that there are valid tilings that contain the seed tile, but none of these tilings is recursive. This is a weaker property than arecursivity defined above, because of the presence of a seed tile.

**Theorem 5.14** *There exists a finite set $\mathcal{P}$ of Wang prototiles such that for some $t \in \mathcal{P}$ every valid tiling that contains $t$ is non-recursive, and there are valid non-recursive tilings that contain $t$.*

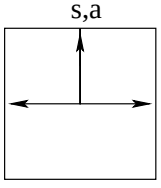
*Proof.* Let us construct for $M_R$ of lemma 5.13 the machine tiles of Section 5.2, except that the start tiles




that represent the initial empty tape will be replaced by the following tiles for all tape letters $x \in \Gamma$, the initial state $s \in S$, and a single tape letter $a \in \Gamma$. Letter $a$ is chosen so that there is an initial tape content $f : \mathbb{Z} \longrightarrow \Gamma$ with $f(0) = a$ such that $M_R$ does not halt when started on tape $f$.




These start tiles allow non-blank initial tapes. As the seed tile $t$ we select the new start tile



The initialization tiles allow the horizontal row with $t$ to contain any initial tape content $f$ with $f(0) = a$. The machine tiles then force the rows above to simulate machine $M_R$. If $M_R$ halts then the tiling becomes impossible.

These tiles admit a valid tiling: We simply choose an initial row that represents a tape content $f$ from which $M_R$ does not halt. But the tiles do not admit any recursive tiling containing $t$: If the tiling is recursive then the horizontal row containing $t$ is recursive, so we would have a recursive initial tape from which $M_R$ does not halt, a contradiction with Lemma 5.13. □

It is possible (but we skip the proof) to modify Robinson's tile set so that the seed tile restriction in Theorem 5.14 can be removed. As in the proof of Theorem 5.8 we form nested boards, and a simulation of the Turing machine $M_R$ is done on all boards. The main problem to be addressed in the construction is the fact the simulations on all boards should be from the same initial tape content. Otherwise, if different boards are allowed to run $M_R$ on different initial tape contents then a recursive tiling could be easily built. Hence new signals need to be introduced that carry the information about the initial tape content between boards of different sizes, so that different boards are forced to be consistent with each other.

**Theorem 5.15** *There exist arecursive sets of Wang tiles.*

*Proof.* For the original proof, if interested, see:

Dale Myers. Nonrecursive Tilings of the Plane, II. *The Journal of Symbolic Logic*, **39(2)**, pp. 286-294, 1974. □

# 6 Compact topology on Wang tilings

In this section we assign a metric to the space $T^{\mathbb{Z}^2}$ of configurations over the finite tile set $T$. The space under this metric is compact and complete. Convergence of a sequence $c_1, c_2, \ldots$ of elements under this metric is exactly equivalent to the convergence introduced in Section 4.2. The compactness principle of that section then simply reflect the compactness of the metric space.

We define the distance $d(e, c)$ between configurations $e, c \in T^{\mathbb{Z}^2}$ as follows:

$$d(e,c) = \begin{cases} 0, & \text{if } e = c, \\ 2^{-\min\{|x|+|y| \;\mid\; e(x,y) \neq c(x,y)\}}, & \text{if } e \neq c. \end{cases}$$

In other words, two configurations that differ in a cell that is close to $\vec{0}$ are far away from each other under this metric, while configurations that agree with each other on a large area around the origin are close to each other. Under this metric, two configurations have distance $< 2^{-r}$ if and only if they agree with each other at all positions $(x, y)$ where $|x| + |y| \leq r$.

Note that other vector norms $\|(x, y)\|$ could be used instead $|x|+|y|$, and any other decreasing function could be used instead of $x \mapsto 2^{-x}$. A different metric, but the same topology would result.

**Lemma 6.1** *Function $d : T^{\mathbb{Z}^2} \times T^{\mathbb{Z}^2} \longrightarrow \mathbb{R}$ is a metric.*

*Proof.* We have to check the three defining properties of metric:

(a) $d(c, e) \geq 0$, and $d(c, e) = 0$ if and only if $c = e$,

(b) $d(c, e) = d(e, c)$, and

(c) $d(c,e) \leq d(c,c') + d(c',e)$.

The first two conditions (a) and (b) are immediate. The third condition (c), called the triangle inequality, follows from the fact that for every $\vec{x} \in \mathbb{Z}^2$, if $c(\vec{x}) \neq e(\vec{x})$ then either $c(\vec{x}) \neq c'(\vec{x})$ or $c'(\vec{x}) \neq e(\vec{x})$, or both. This means that either $d(c,c') \geq d(c,e)$ or $d(c',e) \geq d(c,e)$, so even the strong form

$$d(c,e) \leq \max\{d(c,c'), d(c',e)\}$$

of the triangle inequality holds. □

From now on we consider $T^{\mathbb{Z}^2}$ as a metric topological space under this metric. The following subsection contains a brief review of some basic facts about metric spaces.

## 6.1 Review of topology and metric spaces

Let $X$ be a set. A family $\mathcal{T}$ of subsets of $X$ is called a *topology* if it satisfies the following three conditions:

(i) $\emptyset \in \mathcal{T}$ and $X \in \mathcal{T}$,

(ii) the union of the sets in any subfamily of $\mathcal{T}$ is in $\mathcal{T}$,

(iii) the intersection of finitely many elements of $\mathcal{T}$ is always in $\mathcal{T}$.

Elements of $\mathcal{T}$ are called *open* sets, and their complements (with respect to $X$) are *closed* sets. A set that is both open and closed is called *clopen*.

**Example 12.** For any $X$, let $\mathcal{T}$ contain all subsets of $X$. Then $\mathcal{T}$ is a topology, the *discrete* topology of $X$. Also $\{X, \emptyset\}$ is a topology, the *trivial* topology of $X$. □

**Example 13.** Let us call $S \subseteq \mathbb{R}$ open if for every $x \in S$ there is a positive real $\varepsilon > 0$ such that $|y - x| < \varepsilon \Longrightarrow y \in S$. These open sets form a topology of $X = \mathbb{R}$. It is called the usual topology of $\mathbb{R}$. For example, all open intervals $(a,b)$ for $a < b$ are open sets. Closed intervals $[a,b]$ are not open but they are closed. Set $\mathbb{Q}$ of rational numbers is not open or closed. The only clopen sets are $\emptyset$ and $\mathbb{R}$. □

Generalizing the previous example, let $X$ be a set and let $d : X \times X \longrightarrow \mathbb{R}$ be a metric. For every $\varepsilon > 0$ and $x \in X$ we denote

$$B_\varepsilon(x) = \{y \in X \mid d(x,y) < \varepsilon\}$$

and call $B_\varepsilon(x)$ the (open) $\varepsilon$-ball with center $x$. Let us call $U \subseteq X$ open if

$$\forall x \in U \ : \ \exists \varepsilon > 0 \ : \ B_\varepsilon(x) \subseteq U.$$

These open sets form a topology of $X$, the *metric topology* induced by $d$.

**Example 14.** The discrete topology is induced by the discrete metric

$$d(x,y) = \begin{cases} 0, & \text{if } x = y, \\ 1, & \text{if } x \neq y. \end{cases}$$

In contrast, if $|X| \geq 2$ then the trivial topology $\{X, \emptyset\}$ is not metric. □

Let $A \subseteq X$. Point $x \in X$ is an *accumulation point* of $A$ if every open set $U$ that contains $x$ also contains some element $y \neq x$ of $A$. The following simple properties hold for closed sets:

**Proposition 6.2** *A subset $A \subseteq X$ is closed if and only if its accumulation points belong to $A$. Closed sets satisfy the following properties (that are dual statements of the defining properties of open sets):*

*(i) The empty set $\emptyset$ is closed, and $X$ is closed,*

*(ii) the intersection of any number of closed sets is closed, and*

*(iii) the union of a finite number of closed sets is closed.*

□

Let $A \subseteq X$. The *closure* of $A$ is the intersection of all closed sets that contain $A$. It is then the smallest closed set that contains $A$. We denote the closure of $A$ by $\overline{A}$. Notice that $A$ itself is closed if and only if $\overline{A} = A$. Notice also that the closure of $A$ is the union of $A$ and its set of accumulation points.

Set $A$ is called *dense* if $\overline{A} = X$.

**Example 15.** Consider the usual topology of $\mathbb{R}$. All real numbers are accumulation points of the set $\mathbb{Q}$ of rational numbers. This means that the closure of $\mathbb{Q}$ is $\mathbb{R}$, so $\mathbb{Q}$ is dense in $\mathbb{R}$. Accumulation points of the open interval $(0, 1)$ are the elements of the closed interval $[0, 1]$, while the set $\mathbb{Z}$ of integers has no accumulation points. □

Let $A \subseteq X$. Point $x \in A$ is an *interior point* of $A$ if there is an open set $U$ such that $x \in U$ and $U \subseteq A$. The set of all interior points of $A$ is the *interior* of $A$. It is easily seen to be the union of all open subsets of $A$, or equivalently, the largest open subset of $A$. Then set $A$ is open if and only if its interior is $A$ itself.

The *exterior* of set $A \subseteq X$ is the interior of the complement of $A$, and the *boundary* of $A$ consists of all points that are not in the interior or the exterior of $A$. Note that the interior, exterior and boundary of $A$ is a partitioning of $X$. A set $A \subseteq X$ is called a *neighborhood* of $x \in X$ if $x$ is an interior point of $A$, that is, if there is an open set $U$ such that $x \in U \subseteq A$.

**Example 16.** In the usual topology of $\mathbb{R}$, the interior, exterior and the boundary of an open interval $(a, b)$ are $(a, b)$, $(-\infty, a) \cup (b, \infty)$ and $\{a, b\}$, respectively. The closed interval $[a, b]$ has these same interior, exterior and boundary. Set $\mathbb{Q}$ has empty interior and exterior. All real numbers are in its boundary. □

A topology is called *Hausdorff* if for every $x \neq y$ there are open $U_x$ and $U_y$ such that $x \in U_x$, $y \in U_y$ and $U_x \cap U_y = \emptyset$. In other words, any two distinct points have non-intersecting neighborhoods.

**Example 17.** Every metric topology is Hausdorff. Indeed, if $x \neq y$ then $d(x, y) > 0$. If we choose $\varepsilon = \frac{1}{2}d(x, y)$ then $B_\varepsilon(x)$ and $B_\varepsilon(y)$ are non-intersecting neighborhoods of $x$ and $y$ □

A sequence $x_1, x_2, \ldots$ of points of $X$ *converges* to point $x \in X$ if for every open $U \subseteq X$ that contains $x$ there is positive integer $n$ such that $x_i \in U$ for all $i \geq n$. If the topology is metric this is equivalent to saying that for every $\varepsilon > 0$ there is $n$ such that $d(x_i, x) < \varepsilon$ for all $i \geq n$.

Note that generally a converging sequence may converge to several different points, but if the topology is Hausdorff (e.g. metric) the limit is unique.

**Proposition 6.3** *In Hausdorff topology every converging sequence converges to a unique point.* □

*Proof.* Suppose $x_1, x_2, \ldots$ converges to $x$ and $y$ where $x \neq y$. Since $X$ is Hausdorff, there are open sets $U$ and $V$ such that $x \in U$, $y \in V$ and $U \cap V = \emptyset$. By the definition of convergence, $x_i \in U$ and $x_i \in V$ for all sufficiently large $i$, a contradiction. □

Note: the proposition does not hold in all topological spaces. For example, in the trivial topology $\mathcal{T} = \{\emptyset, X\}$ every sequence converges to every point.

In Hausdorff topology we denote by $\lim_{i\to\infty} x_i$ the unique point into which the sequence $x_1, x_2, \ldots$ converges, if it exists. This point is the limit of the sequence.

The following proposition states that if the topology is metric then the closure $\overline{A}$ of any set $A$ consists exactly of the limits of converging sequences of elements of $A$:

**Proposition 6.4** *Let $X$ be a metric space and $A \subseteq X$. Then $x \in \overline{A}$ if and only if $x = \lim_{i\to\infty} a_i$ for some converging sequence $a_1, a_2, \ldots$ where all $a_i \in A$.*

*Proof.* "$\Longleftarrow$": Let $a_1, a_2, \ldots$ be a converging sequence where all $a_i \in A$ and let $x = \lim_{i\to\infty} a_i$. Let $U$ be an arbitrary open set that contains $x$. By the definition of convergence there are some $a_i \in U$, so $U \cap A \neq \emptyset$. This means that $x \in \overline{A}$. (This direction of the proof holds for any topological space.)

"$\Longrightarrow$": Conversely, suppose $x \in \overline{A}$. For every positive integer $i$, let $a_i$ be an element of $A \cap B_{\frac{1}{i}}(x)$. Then $d(x, a_i) < \frac{1}{i}$, so $x = \lim_{i\longrightarrow\infty} a_i$. □

**Corollary 6.5** *In metric space $X$, set $A$ is closed if and only if it contains the limit of every converging sequence of its elements.* □

A family $\mathcal{B}$ of open sets is called a *base* of the topology iff every open set is the union of some members of $\mathcal{B}$. Equivalently: $\mathcal{B} \subseteq \mathcal{T}$ is a base if for every open set $U$ and $x \in U$ there exists some $B \in \mathcal{B}$ with the property that $x \in B \subseteq U$.

**Example 18.** The open intervals $(a, b)$ with $a < b$ form a base of the usual topology of $\mathbb{R}$. More generally, in any metric topology the open balls $B_\varepsilon(x)$ over all $\varepsilon > 0$ and $x \in X$ form a base. □

If $\mathcal{B}$ is a base of a topology then this topology is uniquely determined by $\mathcal{B}$:open sets are exactly the unions of members of $\mathcal{B}$.

Next we define compactness. Let $A \subseteq X$ where $X$ is a topological space. A family of open sets $U_i$ is called an *open cover* of $A$ if every element of $A$ belongs to some $U_i$. A subfamily of an open cover of $A$ is called a *subcover* if it is also a cover of $A$.

Set $A \subseteq X$ is called *compact* if every open cover of $A$ has a finite subcover of $A$. The topology is called compact if the whole space $X$ is compact. In other words, a topology is compact if every family of open sets whose union is $X$ has a finite subfamily whose union is $X$.

**Example 19.** In the usual topology of $\mathbb{R}$ the set

$$A = \{0\} \cup \{\frac{1}{n} \mid n \in \mathbb{Z}_+\}$$

is compact. Namely, an open set that contains 0 covers all but finitely many elements of $A$. So any open cover of $A$ contains a finite subcover: Open set $U$ that covers 0 together with a finite number of open sets that cover the finitely many elements of $A$ that are outside of $U$.

On the other hand, set $B = \{\frac{1}{n} \mid n \in \mathbb{Z}_+\}$ is not compact. It has an open cover in which every open set covers exactly one element of $B$. Such cover has no finite subcover. □

The following proposition states the finite intersection property. It is dual to the open cover property we used as the definition, and in fact the finite intersection property could have been taken equally well as the definition of compactness. We state the property for the whole space $X$:

**Proposition 6.6** *Topology of $X$ is compact if and only if every family of closed sets whose intersection is empty has a finite subfamily whose intersection is empty.*

*Proof.* This follows directly from the definition of compactness and de Morgan's laws: A family of open sets is a cover of $X$ if and only if the family of their complements have empty intersection. □

We typically apply the previous proposition in the following set-up:

**Corollary 6.7** *Let $F_1 \supseteq F_2 \supseteq F_3 \supseteq \ldots$ be an infinite chain of closed sets in a compact space $X$. If*

$$\bigcap_{i=1}^{\infty} F_i = \emptyset,$$

*then $F_i = \emptyset$ for some $i$.* □

The next proposition gives a characterization of compact subsets in metric spaces. The proposition gives a condition that looks very similar to Proposition 4.2 for configurations. In fact, we use the proposition later to show the compactness of the configuration space. The proposition is valid (and is stated) for arbitrary metric spaces, but we only prove it now for metric spaces that have a countable base. Our configuration space satisfies this restriction, so the proof is sufficient for our set-up. The proof for general metric spaces is not very difficult either.

**Proposition 6.8** *Suppose $X$ is a metric space. Set $A \subseteq X$ is compact if and only if every sequence $a_1, a_2, \ldots$ of elements of $A$ has a subsequence that converges to an element of $A$.*

*Proof.* "$\Longrightarrow$" Suppose $A$ is compact, and let $a_1, a_2, \ldots$ be arbitrary sequence where each $a_i \in A$.

Suppose first that there is some $a \in A$ such that for every $\varepsilon > 0$ the ball $B_\varepsilon(a)$ contains infinitely many different elements of the sequence $a_1, a_2, \ldots$. Then the sequence has a subsequence that converges to $a$: There namely is a subsequence whose $n$'th element belongs to $B_{\frac{1}{n}}(a)$.

Suppose then that for every $a \in A$ there is some $\varepsilon_a > 0$ such that $B_{\varepsilon_a}(a)$ only contains finitely many different elements of the sequence $a_1, a_2, \ldots$. Clearly the family of $B_{\varepsilon_a}(a)$ over all $a \in A$ is an open cover of $A$, so by compactness of $A$ it has a finite subcover

$$U_i = B_{\varepsilon_{a_i}}(a_i) \text{ for } i = 1, 2, \ldots m.$$

But each $U_i$ only covers finitely many different elements of sequence $a_1, a_2, \ldots$, while each element of the sequence is covered by some $U_i$. This means that the sequence has only finitely many different elements. Then some element $a \in A$ repeats infinitely many times in the sequence so the sequence has a constant subsequence $a, a, \ldots$ which trivially converges to $a \in A$.

"$\Longleftarrow$" Suppose every sequence of elements of $A$ has a converging subsequence whose limit is in $A$. Here we simplify the set-up by making the additional assumption that the topology has a countable base. Then it is enough to show that any countable open cover of $A$ has a finite sub-cover. (Indeed, for an arbitrary open cover by $U_i$ we can consider instead the countable cover that consists of all base sets $B_j$ that are completely included in some $U_i$. If every countable cover has a finite subcover, then the original cover also has a finite subcover where we take for each selected $B_j$ one $U_i$ from the original cover that satisfies $B_j \subseteq U_i$.)

So consider a countable open cover $\{U_1, U_2, \ldots\}$ of $A$. If it has no finite subcover then for every $i$ there is some $a_i \in A$ such that $a_i \notin U_j$ for all $j < i$. By the hypothesis, sequence $a_1, a_2, \ldots$ has a converging subsequence with limit $a \in A$. But $a \in U_j$ for some $j$, and then by the definition of convergence $a_i \in U_j$ for infinitely many indices $i$. In particular, there is $i > j$ such that $a_i \in U_j$, which contradicts the choice of $a_i$'s. We conclude that a finite subcover must exist. □

Next two propositions show that in our forthcoming situation compact sets of the space are exactly the closed sets.

**Proposition 6.9** *If $X$ is a compact topological space then every closed $A \subseteq X$ is compact.*

*Proof.* Let $A \subseteq X$ be closed. Consider an open cover of $A$. Together with the complement of $A$ it forms an open cover of $X$. By compactness of $X$ this has a finite subcover of $X$, from which we obtain a finite subcover of $A$ by removing the complement of $A$ (if present). Hence $A$ is compact. □

**Proposition 6.10** *If $X$ is Hausdorff then every compact $A \subseteq X$ is closed.*

*Proof.* Let $A \subseteq X$ be compact. Let $x \in X \setminus A$. By the Hausdorff property, for every $a \in A$ there are open sets $U_a$ and $V_a$ such that $a \in U_a$, $x \in V_a$ and $U_a \cap V_a = \emptyset$. Sets $U_a$ form an open cover of $A$ so by compactness of $A$ there is a finite subcover $U_{a_1}, \ldots, U_{a_m}$ of $A$. But then the intersection

$$V_x = V_{a_1} \cap \ldots \cap V_{a_m}$$

of the corresponding sets $V_{a_i}$ is an open set satisfying $x \in V_x$ and $V_x \cap A = \emptyset$. The union of sets $V_x$ over all $x \in X \setminus A$ is the complement of $A$. Since the union is open, we see that $A$ is closed. □

A topological space is *separable* if it has a countable dense subset,and it is *second countable* if it has a countable base. Our space of interest is both separable and second countable. In fact, every compact metric space has these properties.

**Proposition 6.11** *A compact metric space is separable.*

*Proof.* For every $n$ the cover of $X$ by the open balls $B_{1/n}(x)$ has a finite subcover. The centers of all the balls in these finite subcovers for $n = 1, 2, 3, \ldots$ form a countable set $A$. It is dense: For every $y \in X$ and $n \geq 1$ there is a ball $B_{1/n}(x)$ with center $x \in A$ that contains $y$. Then $x \in B_{1/n}(y)$. □

**Proposition 6.12** *A metric space is separable if and only if it has a countable base.*

*Proof.* Let $\{x_1, x_2, \ldots\}$ be a dense countable subset of $X$. Then the open balls $B_{1/n}(x_i)$ over all positive integers $i, n$ form a countable base. Indeed: For every open $U$ and $x \in U$ there exists $\varepsilon > 0$ such that $B_\varepsilon(x) \subseteq U$. Choose an integer $n > 2/\varepsilon$. Some $x_i \in B_{1/n}(x)$. Because $1/n < \varepsilon/2$ we have

$$x \in B_{1/n}(x_i) \subseteq B_\varepsilon(x) \subseteq U.$$

Conversely, if $U_1, U_2, \ldots$ is a countable base, then $\{x_1, x_2, \ldots\}$ is dense where each $x_i \in U_i$. □

Let $A \subseteq X$. We say that point $x \in A$ is *isolated* in $A$ if there is an open set $U$ such that $A \cap U = \{x\}$. In other words, some open neighborhood of $x$ does not contain any other elements of $A$. A non-empty set $S$ is called *perfect* if it is closed and has no isolated points.

**Proposition 6.13** *In a compact metric space, a perfect set is uncountable.*

*Proof.* Clearly, in a Hausdorff space, all points of a finite set are isolated, so a perfect set is infinite. Suppose there is a countable perfect set

$$S = \{x_1, x_2, \ldots\}.$$

In the following we define a decreasing sequence $F_1 \supseteq F_2 \supseteq F_3 \supseteq \ldots$ of closed sets such that, for all $i$, set $F_i$ contains an open neighborhood $U_i$ of some element of $S$, but $x_i \notin F_i$.

First, $F_1 = \overline{B}_\varepsilon(x_2)$ where $\varepsilon < d(x_1, x_2)$. Then the open neighborhood $U_1 = B_\varepsilon(x_2)$ of $x_2$ is contained if $F_1$, but $x_1 \notin F_1$. Suppose then $F_{i-1}$ and $U_{i-1}$ have been defined. Because $S$ has no isolated points, the open neighborhood of any point contains also other elements of $S$. Hence $U_{i-1}$ contains at least two elements of $S$, and consequently some $a \in S$, $a \neq x_i$, is in $U_{i-1}$. We choose

$$\begin{aligned} F_i &= F_{i-1} \cap \overline{B}_\varepsilon(a) \text{ and} \\ U_i &= U_{i-1} \cap B_\varepsilon(a), \end{aligned}$$

where $\varepsilon < d(a, x_i)$. Then $F_i$ is closed, $F_{i-1} \supseteq F_i$ and $x_i \notin F_i$. Moreover, $U_i$ is open and $a \in U_i \subseteq F_i$.

Because $F_i \cap S$ are closed and non-empty, by Corollary 6.7 the intersection

$$A = \bigcap_{i=1}^{\infty} F_i \cap S$$

is not empty. But $x_i \notin F_i$, so $A = \emptyset$, a contradiction. □

Finally, a few words about continuous functions. Let $X$ and $Y$ be two topological spaces. A function $f : X \longrightarrow Y$ is *continuous* at point $x \in X$ if for every open $V \subseteq Y$ that contains $f(x)$ there exists an open $U \subseteq X$ such that $x \in U$ and $f(U) \subseteq V$.

If $X$ and $Y$ are metric spaces with metrics $d$ and $e$, respectively, then continuity at $x$ is equivalent to the following: For every $\varepsilon > 0$ there exists $\delta > 0$ such that $f(B_\delta(x)) \subseteq B_\varepsilon(f(x))$.

We call function $f : X \longrightarrow Y$ is *continuous* if it is continuous at every $x \in X$.

**Example 20.** If $X$ has the discrete topology then every function $f : X \longrightarrow Y$ is continuous. Also, if $Y$ has the trivial topology $\{\emptyset, Y\}$ then every $f : X \longrightarrow Y$ is continuous. In all topological spaces $X$ and $Y$ all constant functions $f : X \longrightarrow Y$ are continuous. If $X$ has the trivial topology and $Y$ has the discrete topology then the constant functions are the only continuous functions. □

**Proposition 6.14** *Let $f : X \longrightarrow Y$ be a function between two topological spaces. The following conditions are equivalent:*

*(i) Function $f : X \longrightarrow Y$ is continuous,*

*(ii) pre-image $f^{-1}(V)$ is open in $X$ for every open $V \subseteq Y$,*

*(iii) pre-image $f^{-1}(C)$ is closed in $X$ for every closed $C \subseteq Y$.*

*Proof.* (i) $\Longrightarrow$ (ii): Suppose $f$ is continuous and let $V \subseteq Y$ be open. Let $x \in f^{-1}(V)$ be arbitrary, so $f(x) \in V$. From continuity it follows that there is an open $U \subseteq X$ such that $f(U) \subseteq V$ and $x \in U$. This means that $x \in U \subseteq f^{-1}(V)$, which implies that $f^{-1}(V)$ is open.

(ii) $\Longrightarrow$ (i): Suppose $f^{-1}(V)$ is open for every open $V \subseteq Y$. Let $x \in X$ be arbitrary. Let us show that $f$ is continuous at point $x$. Let $f(x) \in V$ for open $V \subseteq Y$. Then $U = f^{-1}(V)$ is an open set that satisfies $x \in U$ and $f(U) \subseteq V$. So $f$ is continuous at $x$.

(ii) $\Longleftrightarrow$ (iii): Follows directly from the fact that for every $A \subseteq Y$ holds

$$X \setminus f^{-1}(A) = f^{-1}(Y \setminus A).$$

□

Next propositions give some properties of continuous functions and compact sets.

**Proposition 6.15** *Suppose function $f : X \longrightarrow Y$ is continuous. For every compact $A$ the set $f(A)$ is compact.*

*Proof.* Consider an open cover of $f(A)$ by open sets $V_i$. Then, by Proposition 6.14 the sets $f^{-1}(V_i)$ form an open cover of $A$. By compactness of $A$ there is a finite subcover of $A$ by $f^{-1}(V_i)$ where $i \in F$ for some finite set $F$. But then the corresponding sets $V_i$ for $i \in F$ form a finite subcover of $f(A)$. Hence $f(A)$ is compact. □

**Proposition 6.16** *If $f : X \longrightarrow Y$ is a continuous bijection where $X$ is compact and $Y$ is Hausdorff then the inverse function $f^{-1} : Y \longrightarrow X$ is also continuous.*

*Proof.* By Proposition 6.14 it is enough to show that for every closed $A \subseteq X$ also $f(A)$ is closed. But if $A \subseteq X$ is closed then by Proposition 6.9 it is also compact. By Proposition 6.15 set $f(A)$ is also compact, and then by Proposition 6.10 set $f(A)$ is closed. □

## 6.2 Basic facts about the configuration space

Let us return to the space of interest to us: The space $T^{\mathbb{Z}^2}$ of configurations, with the metric

$$d(e,c) = \begin{cases} 0, & \text{if } e = c, \\ 2^{-\min\{|x|+|y| \mid e(x,y)\neq c(x,y)\}}, & \text{if } e \neq c. \end{cases}$$

The open ball of radius $\varepsilon = 2^{-r}$ centered at $c \in T^{\mathbb{Z}^2}$ is

$$B_\varepsilon(c) = \{e \in T^{\mathbb{Z}^2} \mid e(\vec{x}) = c(\vec{x}) \text{ for all } \|\vec{x}\| \leq r\},$$

where (and from now on) we denote $\|(x,y)\| = |x| + |y|$. These balls form a base of the topology.

More generally, for any finite domain $D \subseteq \mathbb{Z}^2$ and configuration $c \in T^{\mathbb{Z}^2}$ we define the *cylinder set*

$$\mathrm{Cyl}(c,D) = \{e \in T^{\mathbb{Z}^2} \mid e(\vec{x}) = c(\vec{x}) \text{ for all } \vec{x} \in D\}$$

that contains all those configurations that agree with $c$ in domain $D$.

Note that for sufficiently large $r$ we have $D \subseteq E$ where

$$E = \{\vec{x} \in \mathbb{Z}^d \mid \|\vec{x}\| \leq r\}.$$

Then

$$\mathrm{Cyl}(c,D) = \bigcup_{e \in \mathrm{Cyl}(c,D)} \mathrm{Cyl}(e,E),$$

so all cylinders are (finite) unions of open balls, and hence they are open in the topology. Balls form a base of the topology, so also cylinders form a base.

The complement of cylinder $\mathrm{Cyl}(c,D)$ is

$$\bigcup_{e \notin \mathrm{Cyl}(c,D)} \mathrm{Cyl}(e,D),$$

so all cylinders are also closed, hence clopen. Our space has a clopen base. Clopen sets are exactly finite unions of cylinders (homework).

Let us next show that a sequence of configurations $c_1, c_2, \ldots$ converges to $c \in T^{\mathbb{Z}^2}$ in this topology, if and only if it converges to $c$ according to the definition of convergence in Section 4.2. First, suppose convergence to $c$ in the topology, and let $\vec{n} \in \mathbb{Z}^2$ be arbitrary. Denote

$$U = \mathrm{Cyl}(c, \{\vec{n}\}).$$

Convergence to $c$ implies that for all sufficiently large $i$ holds $c_i \in U$, that is, $c_i(\vec{n}) = c(\vec{n})$. So the sequence converges to $c$ according to the definition of Section 4.2.

Conversely, suppose converge to $c$ as defined in Section 4.2. Let $U$ be an open set that contains $c$. Because cylinders form a base, there is a finite $D \subseteq \mathbb{Z}^d$ such that $\mathrm{Cyl}(c, D) \subseteq U$. By the definition of convergence of $c_1, c_2, \ldots$ there is $k \in \mathbb{Z}$ such that $c_i \in \mathrm{Cyl}(c, D)$ for all $i > k$. This means that the sequence converges to $c$ in the topology.

Now we immediately obtain the following corollaries of our earlier propositions:

**Corollary 6.17** *The metric space $T^{\mathbb{Z}^2}$ is compact.*

*Proof.* Follows directly from Propositions 4.2 and 6.8. □

Based on the propositions in the previous section, every compact metric space is a Hausdorff, separable and second countable, so the space $T^{\mathbb{Z}^2}$ has all these properties.

Next we look into translations and show that they are continuous functions. A *translation* by $\vec{n} \in \mathbb{Z}^2$ is the transformation $\tau_{\vec{n}} : T^{\mathbb{Z}^2} \longrightarrow T^{\mathbb{Z}^2}$ that maps $c \mapsto e$ where $e(\vec{m}) = c(\vec{m} - \vec{n})$ for all $\vec{m} \in \mathbb{Z}^2$. Translations are bijective, and $\tau_{\vec{n}}$ and $\tau_{-\vec{n}}$ are inverses of each other. The *east shift* $\sigma_e$ and the *north shift* $\sigma_n$ are translations by vectors $(1, 0)$ and $(0, 1)$ respectively, and the *west* and the *south shifts* are their inverses $\sigma_w = \sigma_e^{-1}$ and $\sigma_s = \sigma_n^{-1}$. All translations are compositions of the four shifts. Let us denote by $\mathbb{T}$ the set of all translations.

For every $\vec{n} \in \mathbb{Z}^2$ and $D \subseteq \mathbb{Z}^2$ we denote the translation of $D$ by $\vec{n}$ as

$$D + \vec{n} = \{\vec{d} + \vec{n} \mid \vec{d} \in D\}.$$

Let $\mathrm{Cyl}(c, D)$ be an arbitrary cylinder. Because

$$\tau_{\vec{n}}\left(\mathrm{Cyl}(c, D)\right) = \mathrm{Cyl}(\tau_{\vec{n}}(c), D + \vec{n})$$

we have that translations $\tau_{\vec{n}}$ are continuous.

So we have a compact, metric space $T^{\mathbb{Z}^2}$, equipped with continuous transformations generated by $\sigma_s, \sigma_e, \sigma_n, \sigma_w$. This is a set-up studied by topological dynamics.

## 6.3 Subshifts

A set $A \subseteq T^{\mathbb{Z}^2}$ is *translation invariant* if $\tau(A) = A$ for every $\tau \in \mathbb{T}$. For translation invariance it is enough to verify that $\sigma_e(A) = A$ and $\sigma_n(A) = A$. A topologically closed, translation invariant set is a (two-dimensional) *subshift*, while the entire configuration space $T^{\mathbb{Z}^2}$ is also called the (two-dimensional) *full shift* over the alphabet $T$.

Recall from the beginning of Section 5.6 that a finite pattern over $T$ is a pair $(D, p)$ where $D \subseteq \mathbb{Z}^2$ is finite, the domain of the pattern, and $p : D \longrightarrow T$. Let us denote by $\mathcal{P}(T)$ the set of all finite patterns over $T$. Clearly $\mathcal{P}(T)$ is countable as the number of finite subsets of $\mathbb{Z}^2$ is countable.

Pattern $(D, p)$ is a *subpattern* of $c \in T^{\mathbb{Z}^2}$ if $c(\vec{n}) = p(\vec{n})$ for all $\vec{n} \in D$. Configurations that have $(D, p)$ as a subpattern form a cylinder which we denote by

$$\mathrm{Cyl}(p, D) = \{c \in T^{\mathbb{Z}^2} \mid c(\vec{n}) = p(\vec{n}) \text{ for all } \vec{n} \in D\ \}.$$

This is of course the same cylinder as $\mathrm{Cyl}(c, D)$ for any configuration $c$ in the cylinder.

We say that the pattern $(D, p)$ *appears* in $c$ if $(D, p)$ is a subpattern of $\tau(c)$ for some translation $\tau \in \mathbb{T}$. For any configuration $c$ let $\mathrm{Patt}(c)$ be the set of all finite patterns that appear in $c$, and for any $A \subseteq T^{\mathbb{Z}^2}$ we denote by

$$\mathrm{Patt}(A) = \bigcup_{c \in A} \mathrm{Patt}(c)$$

the set of finite patterns that appear in some element of $A$.

For any set $P$ of finite patterns we define the set

$$\Sigma(P) = \{c \in T^{\mathbb{Z}^2} \mid \mathrm{Patt}(c) \cap P = \emptyset\}$$

of configurations in which no element of $P$ appears. Next we prove that sets $\Sigma(P)$ are precisely the subshifts over $T$.

**Theorem 6.18** *$\Sigma \subseteq T^{\mathbb{Z}^2}$ is a subshift if and only if $\Sigma = \Sigma(P)$ for some set $P$ of finite patterns over $T$.*

*Proof.* First we observe that $\Sigma(P)$ is a subshift, for every $P \subseteq \mathcal{P}(T)$. Clearly $\Sigma(P)$ is translation invariant because $\mathrm{Patt}(c) = \mathrm{Patt}(\tau(c))$ for all configurations $c$ and translations $\tau$. And $\Sigma(P)$ is closed because for every $c \notin \Sigma(P)$ there exists $(D, p) \in P$ that appears in $c$, i.e., is a subpattern of $\tau(c)$ for some translation $\tau$. Then $\tau(c) \in \mathrm{Cyl}(p, D)$ and $\mathrm{Cyl}(p, D) \cap \Sigma(P) = \emptyset$. This means that $\tau^{-1}(\mathrm{Cyl}(p, D))$ is an open neighborhood of $c$ whose intersection with $\Sigma(P)$ is empty.

For the converse direction, let $\Sigma$ be an arbitrary subshift over $T$ and define

$$P = \mathcal{P}(T) \setminus \bigcup_{x \in \Sigma} \mathrm{Patt}(x),$$

that is, $P$ contains all the patterns that do not appear in any configuration belonging to $\Sigma$. Let us prove that $\Sigma = \Sigma(P)$. If $c \in \Sigma$ then by the definition of $P$ we have $\mathrm{Patt}(c) \cap P = \emptyset$. Hence $c \in \Sigma(P)$. And if $c \in \Sigma(P)$ then $\mathrm{Patt}(c) \subseteq \cup_{x \in \Sigma} \mathrm{Patt}(x)$, so for every finite $D \subseteq \mathbb{Z}^2$ we have $\Sigma \cap \mathrm{Cyl}(c, D) \neq \emptyset$. Because $\Sigma$ is closed we have $c \in \Sigma$. □

Subshifts $\Sigma(P)$ for finite $P$ are called *subshifts of finite type (SFT)*. So a SFT can be specified by giving a finite collection $P$ of forbidden patterns. But this is exactly how we defined valid tilings in Section 5.3. So the valid tilings form a SFT, and conversely, every SFT is the set of valid tilings when the forbidden patterns are defined as in the SFT. Valid tilings by Wang tiles are particular types of subshifts of finite type, with small forbidden patterns. The construction in Lemma 5.5 in Section 5.3 in fact shows the following: Every subshift of finite type is conjugate to the set of valid tilings by some Wang tile set. (Two subshifts $X$ and $Y$ are called *conjugate* if there exists a translation commuting homeomorphism between them, i.e., a continuous bijection $h : X \longrightarrow Y$ such that $h \circ \tau = \tau \circ h$ for all $\tau \in \mathbb{T}$. Conjugate subshifts are in many respects equivalent with each other.)

## 6.4 Orbits, transitivity and minimality

For any $c \in T^{\mathbb{Z}^2}$ the set

$$\mathcal{O}(c) = \{\tau(c) \mid \tau \in \mathbb{T}\}$$

is the *orbit* of $c$. The set $\mathcal{O}(c)$ is trivially translation invariant. The orbit is not necessarily closed, so we frequently consider the *orbit closure* $\overline{\mathcal{O}(c)}$. The orbit closure is translation invariant: Let $e \in \overline{\mathcal{O}(c)}$ and let $\tau$ be any translation. Since there are $e_1, e_2, \ldots \in \mathcal{O}(c)$ such that $\lim_{i \longrightarrow \infty} e_i = e$, we have $\lim_{i \longrightarrow \infty} \tau(e_i) = \tau(e)$ and each $\tau(e_i) \in \mathcal{O}(c)$. This means that $\tau(e) \in \overline{\mathcal{O}(c)}$. We have proved the following:

**Lemma 6.19** *The orbit closure $\overline{\mathcal{O}(c)}$ is a subshift, for every $c \in T^{\mathbb{Z}^2}$.* □

The orbit closure $\overline{\mathcal{O}(c)}$ is the subshift generated by $c$, that is, the intersection of all subshifts that contain $c$.

**Example 21.** Let $T = \{0, 1\}$ and let $c \in T^{\mathbb{Z}^2}$ be the infinite cross: $c(i, 0) = c(0, i) = 1$ for all $i \in \mathbb{Z}$ and $c(i, j) = 0$ if $i, j \neq 0$. Then $\mathcal{O}(c)$ is not closed. The orbit closure also contains the zero configuration $c_0$ with $c_0(i, j) = 0$ for all $i, j \in \mathbb{Z}$, as well as horizontal and vertical rows of 1's, i.e. elements of $\mathcal{O}(c_v)$ and $\mathcal{O}(c_h)$ where $c_h(i, 0) = c_v(0, i) = 1$ for all $i \in \mathbb{Z}$ and $c_h(i, j) = c_v(j, i) = 0$ if $j \neq 0$. □

**Lemma 6.20** *$e \in \overline{\mathcal{O}(c)}$ if and only if Patt$(e) \subseteq$ Patt$(c)$.*

*Proof.* We have $e \in \overline{\mathcal{O}(c)}$ if and only if for every finite $D \subseteq \mathbb{Z}^2$ there exists $\tau \in \mathbb{T}$ such that $\tau(c) \in \mathrm{Cyl}(e, D)$. But this is equivalent to $\mathrm{Patt}(e) \subseteq \mathrm{Patt}(c)$. □

A non-empty subshift $\Sigma$ is called *transitive* if for every $(D_1, p_1), (D_2, p_2) \in \mathrm{Patt}(\Sigma)$ there exists $c \in \Sigma$ such that $(D_1, p_1), (D_2, p_2) \in \mathrm{Patt}(c)$. In other words, any two patterns that appear in some elements of $\Sigma$, appear in the same element of $\Sigma$. Next we show that transitive subshifts are exactly the orbit closures of configurations:

**Theorem 6.21** *Subshift $\Sigma$ is transitive if and only if $\Sigma = \overline{\mathcal{O}(c)}$ for some $c \in \Sigma$.*

*Proof.* For every configuration $c$ the subshift $\overline{\mathcal{O}(c)}$ is transitive: By Lemma 6.20 we have the inclusion $\mathrm{Patt}(\overline{\mathcal{O}(c)}) \subseteq \mathrm{Patt}(c)$, so all patterns that appear in some elements of $\overline{\mathcal{O}(c)}$ appear in $c$, and hence $\Sigma$ is transitive.

Conversely, suppose that $\Sigma$ is transitive. By transitivity and translation invariance of $\Sigma$, if $U$ and $V$ are cylinders such that $U \cap \Sigma \neq \emptyset$ and $V \cap \Sigma \neq \emptyset$ then there is a translation $\tau$ such that $U \cap \tau(V) \cap \Sigma \neq \emptyset$. And since non-empty intersections of cylinders are cylinders, the set $U \cap \tau(V)$ is a cylinder.

Let $U_1, U_2, \ldots$ be all the cylinders such that $U_i \cap \Sigma \neq \emptyset$. By the observation above, there are translations $\tau_1, \tau_2, \ldots$ such that

$$V_n = \tau_1(U_1) \cap \tau_2(U_2) \cap \ldots \cap \tau_n(U_n) \cap \Sigma$$

is non-empty for every $n = 1, 2, \ldots$. Every $V_n$ is closed and $V_1 \supseteq V_2 \supseteq V_3 \supseteq \ldots$. By compactness there exists $c$ in their intersection. This $c$ is in $\Sigma$ and it contains the patterns in $\mathrm{Patt}(\Sigma)$. This means that $\Sigma = \overline{\mathcal{O}(c)}$. □

So we can observe that if $\Sigma$ is transitive then some $c \in \Sigma$ contains all the finite patterns that appear in any elements of $\Sigma$.

Motivated by the previous theorem, we call an element $c \in \Sigma$ *transitive* in $\Sigma$ if $\Sigma = \overline{\mathcal{O}(c)}$. By the theorem, a transitive element $c$ exists exactly in those subshifts that are transitive, and in that case $\mathcal{O}(c)$ is a dense subset of $\Sigma$ consisting of transitive elements.

**Example 22.** In Example 21 the infinite cross is transitive in its orbit closure. The orbit closure contains a non-transitive subset $\Sigma = \overline{\mathcal{O}(c_v)} \cup \overline{\mathcal{O}(c_h)}$ generated by the horizontal and vertical rows of 1's. □

A non-empty subshift $\Sigma$ is called *minimal* if the only subshifts contained in $\Sigma$ are $\emptyset$ and $\Sigma$.

**Theorem 6.22** *Let $\Sigma$ be a subshift. The following are equivalent:*

*(i) $\Sigma$ is minimal.*

*(ii) All elements of $\Sigma$ are transitive in $\Sigma$.*

*(iii) Patt$(e)$ = Patt$(c)$ for all $e, c \in \Sigma$.*

*Proof.* (i) $\implies$ (ii): For every $c \in \Sigma$ the orbit closure $\overline{\mathcal{O}(c)}$ is a subshift inside $\Sigma$, so by minimality $\overline{\mathcal{O}(c)} = \Sigma$.

(ii) $\implies$ (i): If $\Sigma$ is not minimal then it properly contains a non-empty subshift $\Sigma' \subsetneq \Sigma$. If $c \in \Sigma'$ then $\overline{\mathcal{O}(c)} \subseteq \Sigma'$, so $c$ is not transitive in $\Sigma$.

(ii) $\implies$ (iii): By the definition of transitivity, $\overline{\mathcal{O}(e)} = \Sigma = \overline{\mathcal{O}(c)}$ for all $e, c \in \Sigma$. By Lemma 6.20 this means that $\mathrm{Patt}(e) = \mathrm{Patt}(c)$.

(iii) $\implies$ (ii): If $\mathrm{Patt}(e) = \mathrm{Patt}(c)$ for all $e, c \in \Sigma$ then by Lemma 6.20 we have $e \in \overline{\mathcal{O}(c)}$. As $e \in \Sigma$ is arbitrary, we have $\Sigma \subseteq \overline{\mathcal{O}(c)}$, i.e., $c$ is transitive in $\Sigma$. □

So the theorem states that $\Sigma$ is minimal if and only if the orbits of all its elements are dense in $\Sigma$. Next we show that minimal subshifts are found inside all non-empty subshifts. This could be proved directly using Zorn's lemma. We present an elementary topological proof.

**Theorem 6.23** *Every non-empty subshift $\Sigma$ contains a minimal subshift.*

*Proof.* Cylinders form a countable base $U_1, U_2, \ldots$ of the topology. Let us denote by

$$\mathcal{O}(U_i) = \{\tau(c) \mid \tau \in \mathbb{T}, c \in U_i\} = \bigcup_{\tau \in \mathbb{T}} \tau(U_i)$$

the orbit $U_i$. It is clearly translation invariant, and also open as a union of open sets $\tau(U_i)$.

Inductively we construct a sequence $F_0 \supseteq F_1 \supseteq F_2 \supseteq \ldots$ of non-empty, closed, translation invariant sets as follows. $F_0 = \Sigma$. Then suppose that $F_{m-1}$ has been defined. If $F_{m-1} \subseteq \mathcal{O}(U_m)$ then $F_m = F_{m-1}$, else $F_m = F_{m-1} \setminus \mathcal{O}(U_m)$. Then $F_m \neq \emptyset$, $F_m$ is closed as $F_{m-1}$ is closed and $\mathcal{O}(U_m)$ is open, and $F_m$ is translation invariant because $F_{m-1}$ and $\mathcal{O}(U_m)$ are translation invariant. Let

$$F = \bigcap_{i=1}^{\infty} F_i.$$

Because all $F_i$ are closed and translation invariant, so is $F$, and it follows from the compactness that $F \neq \emptyset$. So $F$ is a non-empty subshift.

Let us show that $F$ minimal. Suppose on the contrary that there exist $e, c \in F$ such that $\mathrm{Patt}(e) \setminus \mathrm{Patt}(c) \neq \emptyset$. This means that there is a cylinder $U_i$ such that $e \in \mathcal{O}(U_i)$ and $c \notin \mathcal{O}(U_i)$. As $F \subseteq F_{i-1}$ is not a subset of $\mathcal{O}(U_i)$, we see that $F_i = F_{i-1} \setminus \mathcal{O}(U_i)$. But this contradicts the assumption $e \in \mathcal{O}(U_i)$. □

## 6.5 Periodicity and recurrence properties

A configuration $c \in T^{\mathbb{Z}^2}$ is one-way periodic if there exists $\vec{n} \in \mathbb{Z}^2 \setminus \vec{0}$ such that $c = \tau_{\vec{n}}(c)$. Vector $\vec{n}$ is a period of $c$. Configuration $c$ is two-way periodic if it is periodic with two linearly independent periods $\vec{n}_1$ and $\vec{n}_2$. A two-way periodic configuration is always periodic with horizontal and vertical periods $(0, n)$ and $(n, 0)$ for some $n > 0$, as shown in the beginning of Section 4.1. If a subshift of finite type contains a one-way periodic element then it contains a two-way periodic element as well: this was show for valid Wang tilings in Theorem 4.1, and by Lemma 5.5 subshifts of finite type are conjugate to sets of valid Wang tilings. Note that a conjugacy preserves vectors of periodicity.

The orbit $\mathcal{O}(c)$ of $c$ is finite if and only if $c$ is two-way periodic. The orbit is also closed if and only if $c$ is two-way periodic (homework).

Two-way periodicity is a very strong form of recurrence. Some weaker recurrence properties are defined in the following. A configuration $c \in T^{\mathbb{Z}^2}$ is *uniformly recurrent* if for every open $U$ with $c \in U$ there exists a finite $D \subseteq \mathbb{Z}^2$ such that for every $\vec{n} \in \mathbb{Z}^2$ we have $\tau_{\vec{n}+\vec{d}}(c) \in U$ for some $\vec{d} \in D$. In other words, if a finite pattern appears somewhere in a uniformly recurrent $c$ then it appears inside every $n \times n$ square, for some $n$.

A configuration is called *recurrent* if for every open $U$ with $c \in U$ there exists $\vec{n} \neq \vec{0}$ such that $\tau_{\vec{n}}(c) \in U$. In other words, every pattern that appears in $c$ appears more than once. It is easy to see that then the pattern has to appear infinitely many times in $c$:

**Lemma 6.24** *Configuration $c$ is recurrent if and only if for every open neighborhood $U$ of $c$ there are infinitely many translations $\tau \in \mathbb{T}$ such that $\tau(c) \in U$.*

*Proof.* Let $c$ be recurrent, and let $U$ be open, $c \in U$. Suppose, contrary to the claim, that the only translations $\tau$ such that $\tau(c) \in U$ are by vectors $\vec{n}_1, \vec{n}_2, \ldots \vec{n}_k$. Let

$$V = \bigcap_{i=1}^{k} \tau_{-\vec{n}_i}(U).$$

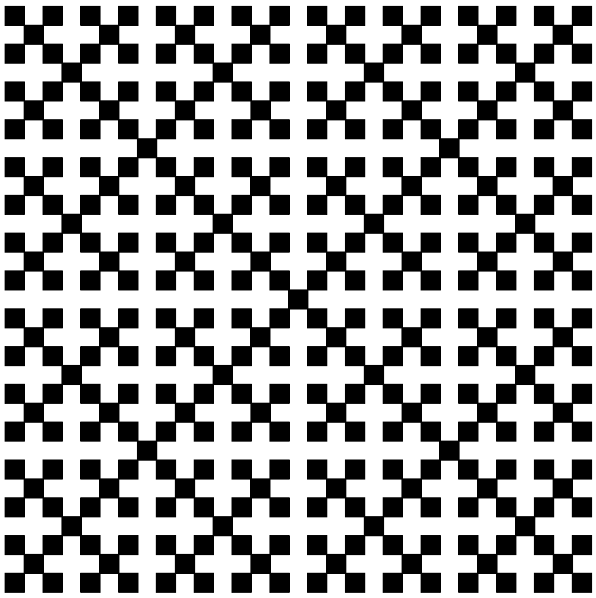
As $V$ is open and $c \in V$, by the recurrence of $c$ there exists $\vec{n} \neq \vec{0}$ such that $\tau_{\vec{n}}(c) \in V$. Then $\tau_{\vec{n}_i+\vec{n}}(c) \in U$ for all $i = 1, 2, \ldots, k$. This means that $\{\vec{n}_1 + \vec{n}, \vec{n}_2 + \vec{n}, \ldots \vec{n}_k + \vec{n}\} = \{\vec{n}_1, \vec{n}_2, \ldots \vec{n}_k\}$. This is possible only if $\vec{n} = \vec{0}$, a contradiction. (Namely, suppose w.l.o.g. that $\vec{n} = (x, y)$ where $x > 0$. If $\vec{n}_i$ has the largest $x$-coordinate among $\vec{n}_1, \vec{n}_2, \ldots \vec{n}_k$, then $\vec{n}_i + \vec{n}$ would have a larger $x$-coordinate than any of the vectors in the set.) □

Configuration $c$ is *quasi-periodic* if for every open neighborhood $U$ there exist linearly independent translation vectors $\vec{a}, \vec{b} \in \mathbb{Z}^2$ such that $\tau_{i\vec{a}+j\vec{b}}(c) \in U$ for all $i, j \in \mathbb{Z}$. In other words, in quasi-periodic configurations all finite patterns are part of a two-way periodic repetition of the pattern, but the period may be different for different patterns. Configuration $c$ is called *isochronous* if for every open neighborhood $U$ there exists an offset vector $\vec{c} \in \mathbb{Z}^2$ and two linearly independent $\vec{a}, \vec{b} \in \mathbb{Z}^2$ such that $\tau_{i\vec{a}+j\vec{b}+\vec{c}}(c) \in U$ for all $i, j \in \mathbb{Z}$.

The following implications are obvious from the definitions:

$$c \text{ two-way periodic} \Longrightarrow c \text{ quasi-periodic} \Longrightarrow c \text{ isochronous} \Longrightarrow c \text{ uniformly recurrent} \Longrightarrow c \text{ recurrent}$$

**Example 23.** For any integer $n$ let us define $\deg_2(n) = k$ if $n = a2^k$ for some odd $a$, and $\deg_2(0) = \infty$. Let $T = \{0, 1\}$. Define configuration $c \in T^{\mathbb{Z}^2}$ as follows: $c(i, j) = 1$ if and only if $\deg_2(i) = \deg_2(j)$. The following figure illustrates $c$ where black square indicates value 1 and white square value 0:



This $c$ is isochronous but not quasi-periodic, as defined above. The configuration is not quasi-periodic, because symbol 1 in position $(0,0)$ is not repeated two-way periodically: all other cells at $(i,0)$ and $(0,i)$ carry symbol 0. But the configuration is isochronous (and hence uniformly recurrent and recurrent) because $\deg_2(n) = \deg_2(n+2^k)$ for every $k > \deg_2(n)$. This means that for any odd integers $a$ and $b$ the translated configuration $\tau_{(a2^k,b2^k)}(c)$ agrees with $c$ inside the square $\{(i,j) \in \mathbb{Z}^2 \mid -2^k < i,j < 2^k\}$. □

Uniformly recurrent configurations are important because they exactly generate all minimal subshifts.

**Theorem 6.25** *Subshift $\overline{\mathcal{O}(c)}$ is minimal if and only if $c$ is uniformly recurrent.*

*Proof.* By Theorem 6.22 subshift $\overline{\mathcal{O}(c)}$ is minimal if and only if $\mathrm{Patt}(e) = \mathrm{Patt}(c)$ for all $e \in \overline{\mathcal{O}(c)}$.

"$\Longleftarrow$": Let $c$ be uniformly recurrent and let $e \in \overline{\mathcal{O}(c)}$ arbitrary. By Lemma 6.20 we know that $\mathrm{Patt}(e) \subseteq \mathrm{Patt}(c)$ so it is enough to show that $\mathrm{Patt}(c) \subseteq \mathrm{Patt}(e)$. Let $(D,p) \in \mathrm{Patt}(c)$. By uniform recurrence of $c$, there exists $n$ such that every $n \times n$ square in $c$ contains a copy of $(D,p)$. Because every $n \times n$ square pattern in $e$ also appears in $c$, every $n \times n$ square pattern of $e$ contains $(D,p)$. So $(D,p) \in \mathrm{Patt}(e)$.

"$\Longrightarrow$": Conversely, assume that $c$ is not uniformly recurrent. Let us show that $\Sigma = \overline{\mathcal{O}(c)}$ is not minimal. By uniform recurrence there exists a finite domain $D \subseteq \mathbb{Z}^2$ such that for any finite $\mathbb{F} \subseteq \mathbb{T}$ a translation $\alpha \in \mathbb{T}$ exists such that for all $\tau \in \mathbb{F}$ holds $\alpha\tau(c) \notin \mathrm{cyl(c,D)}$. In particular, if $\mathbb{T} = \{\tau_1, \tau_2, \ldots\}$ then for every $j = 1, 2, \ldots$ there exists $\alpha_j \in \mathbb{T}$ such that $\alpha_j\tau_i(c) \notin \mathrm{cyl(c,D)}$ for all $i = 1, 2 \ldots, j$. Let $e$ be the limit of a converging subsequence of $\alpha_1(c), \alpha_2(c), \alpha_3(c), \ldots$. Then $e \in \Sigma$ but for all $\tau_i \in \mathbb{T}$ holds $\tau_i(e) \notin \mathrm{cyl(c,D)}$. This is due to the fact that for all $j \geq i$ we have $\tau_i\alpha_j(c) = \alpha_j\tau_i(c) \notin \mathrm{cyl(c,D)}$. We conclude that $\mathrm{Patt}(c) \setminus \mathrm{Patt}(e) \neq \emptyset$, so $\Sigma$ is not minimal. □

We have shown that a subshift $\Sigma$ is minimal if and only if $\Sigma = \overline{\mathcal{O}(c)}$ for some uniformly recurrent $c$, and that in this case all elements of $\Sigma$ are uniformly recurrent. There are two possibilities for the elements of a minimal $\Sigma$: Either all of them are two-way periodic, in which case the subshift is their finite orbit, or all elements are non-periodic (but uniformly recurrent) configurations that contain exactly the same finite patterns. In the second case the subshift turns out to contain an uncountable number of elements:

**Theorem 6.26** *A minimal subshift is either finite or uncountably infinite.*

*Proof.* Let $\Sigma$ be a minimal subshift of countable cardinality. By Proposition 6.13, there is an isolated point $c \in \Sigma$, so that some open set $U$ satisfies $\Sigma \cap U = \{c\}$. By Theorem 6.25 configuration $c$ is uniformly recurrent. This means that there are many translations $\tau$ such that $\tau(c) \in U$. In fact, for some $n$, every $n \times n$ square must contain a point $\vec{n}$ such that $\tau = \tau_{\vec{n}}$ has this property. Whenever $\tau_{\vec{n}}(c) \in U$ we have $\tau_{\vec{n}}(c) = c$, so $c$ is periodic. This implies that $\Sigma = \overline{\mathcal{O}(c)}$ is finite. □

The following corollary states some implications of the above theorems in the case the subshift considered is the set of valid tilings:

**Corollary 6.27** *If a tile set admits a valid tiling then it admits a uniformly recurrent tiling. If it admits a uniformly recurrent tiling that is not two-way periodic, then it admits uncountably many different tilings. In particular, every aperiodic tile set admits uncountably many valid, uniformly recurrent tilings.* □

## 6.6 Equicontinuity and isolated points

Recall that point $c \in \Sigma$ is *isolated* in $\Sigma$ if there exists an open set $U$ such that $U \cap \Sigma = \{c\}$.

**Lemma 6.28** *Let $\Sigma$ be a subshift. All $c \in \Sigma$ are isolated in $\Sigma$ if and only if $\Sigma$ is finite.*

*Proof.* Let $\Sigma$ be a finite subshift, and let $c \in \Sigma$. Set $F = \Sigma \setminus \{c\}$ is closed as a finite union of singleton sets. Hence the complement of $F$ is an open neighborhood of $c$ that does not contain any other elements of $\Sigma$.

Conversely, suppose that $\Sigma$ is an infinite subshift. By compactness there exists an infinite converging sequence $c_1, c_2, \ldots$ where each $c_i \in \Sigma$ and and $c_i \neq c_j$ whenever $i \neq j$. The limit $c = \lim_{i\to\infty} c_i$ is in $\Sigma$, but it is not isolated in $\Sigma$. □

Finite subshifts are exactly the ones whose elements are all two-way periodic:

**Theorem 6.29** *A subshift $\Sigma$ is finite if and only if every $c \in \Sigma$ is two-way periodic.*

*Proof.* If $c$ is not two-way periodic then its orbit is infinite, so one direction is trivial. We only need to show that if all $c \in \Sigma$ are two-way periodic then $\Sigma$ is finite.

Suppose the contrary: $\Sigma$ is infinite and all $c \in \Sigma$ are two-way periodic. Due to infinity, there exists a converging sequence $c_1, c_2, \ldots$ such that all $c_i \in \Sigma$ and $c_i \neq c_j$ for all $i \neq j$. Let $c = \lim_{i\to\infty} c_i$. Because all elements of $\Sigma$ are two-way periodic, $c \in \Sigma$ is two-way periodic.

For each $c_i \neq c$ let us identify a vector $(x_i, y_i) \in \mathbb{Z}^2$ of minimum $n_i = \max\{|x_i|, |y_i|\}$ such that $c_i(x_i, y_i) \neq c(x_i, y_i)$. In other words, $c_i(x, y) = c(x, y)$ for all $-n_i < x, y < n_i$, but $c_i(x_i, y_i) \neq c(x_i, y_i)$ for some $x_i, y_i$ satisfying $|x_i| = n_i$ or $|y_i| = n_i$. Note that $\lim_{i\to\infty} n_i = \infty$.

Infinitely many of the vectors $(x_i, y_i)$ are in the same quadrant of the plane. Without loss of generality we assume now that all $x_i, y_i \geq 0$. By setting $\tau_i = \tau_{(-x_i,-y_i)}$ we see that for all $i = 1, 2, \ldots$ $\tau_i(c_i)(\vec{0}) \neq \tau_i(c)(\vec{0})$ but $\tau_i(c_i)(x, y) = \tau_i(c)(x, y)$ for all $-n_i < x < 0$ and $-n_i < y < 0$. Because $c$ is two-way periodic, there are only finitely many different configurations among $\tau_i(c)$. By choosing a subsequence, we can hence assume now that $\tau_i(c) = \tau(c)$ for some translation $\tau$ and all $i = 1, 2, \ldots$.

Sequence $\tau_1(c_1), \tau_2(c_2), \ldots$ has a converging subsequence. The limit $e \in \Sigma$ of the subsequence coincides with a two-way periodic configuration $\tau(c)$ at $(x, y)$ for all $x, y < 0$, but it does not coincide with $\tau(c)$ at $(0, 0)$. All elements of $\Sigma$ are two-way periodic, so $e$ is two-way periodic. Configurations $e$ and $\tau(c)$ have a common period $(a, b)$ with $a, b < 0$, which implies that

$$\tau(c)(0, 0) = \tau(c)(a, b) = e(a, b) = e(0, 0),$$

a contradiction. □

The following term comes from topological dynamics: Configuration $c \in \Sigma$ is an *equicontinuity point* if

$$(\forall \varepsilon > 0)(\exists \delta > 0)(\forall e \in \Sigma)(\forall \tau \in \mathbb{T})\ d(c,e) < \delta \Longrightarrow d(\tau(c), \tau(e)) < \varepsilon.$$

In other words, if $e$ is chosen sufficiently close to $c$ then all translates $\tau(e)$ and $\tau(c)$ are close to each other.

Because for any $c \neq e$ there exists $\tau \in \mathbb{T}$ such that $d(\tau(c), \tau(e)) = 1$, we see that $c$ is an equicontinuity point in $\Sigma$ if and only if it is isolated in $\Sigma$. Indeed, if $c$ is isolated then for some $\delta > 0$ the only configuration $e$ satisfying $d(c,e) < \delta$ is $c$ itself. And conversely, if $c$ is not isolated then all neighborhoods of $c$ contain $e \neq c$, so the choice $\varepsilon = 1/2$ contradicts the equicontinuity condition at $c$.

The subshift $\Sigma$ is called *equicontinuous* if all $c \in \Sigma$ are equicontinuity points. By the observation above, $\Sigma$ is equicontinuous if and only if all its elements are isolated. By Lemma 6.28 equicontinuous subshifts are exactly the finite subshifts, which by Lemma 6.29 are exactly the subshifts all of whose elements are two-way periodic.

A subshift $\Sigma$ is called *sensitive* if there exists $\varepsilon > 0$, called the sensitivity constant, such that

$$(\forall c \in \Sigma)(\forall \delta > 0)(\exists e \in \Sigma)(\exists \tau \in \mathbb{T})\ 0 < d(c,e) < \delta \text{ and } d(\tau(c), \tau(e)) > \varepsilon.$$

In other words, arbitrarily close to each $c$ there is another configuration $e$ such that for a suitable translation $\tau$ the configurations $\tau(c)$ and $\tau(e)$ are not close to each other. Note that if $c$ is isolated in $\Sigma$ then there are no elements within distance $\delta$ of $c$ for small $\delta$, and hence the system is not sensitive. In contrast, if there are no isolated points then the system is sensitive with any sensitivity constant $0 < \varepsilon < 1$ because, as pointed out above, for any $c \neq e$ we have $d(\tau(c), \tau(e)) = 1$ for a suitable $\tau \in \mathbb{T}$.

Finally, the fact that

$$(\exists \varepsilon > 0)(\forall c, e \in \Sigma)\ c \neq e \Longrightarrow\ (\exists \tau \in \mathbb{T})\ d(\tau(c), \tau(e)) > \varepsilon$$

means, in terms of topological dynamics terminology that all subshifts are *expansive*.

**Theorem 6.30** *Let $\Sigma$ be a subshift.*

*(i) $\Sigma$ is expansive.*

*(ii) $\Sigma$ is sensitive if and only if it has no isolated points.*

*(ii) $\Sigma$ is equicontinuous if and only all its elements are isolated, i.e., the subshift is finite.*

□

# 7 A brief revisit to tilings by polygons

In the beginning of Section 4 we showed how any Wang protoset can be converted into an equivalent set of prototiles that are polygons, by replacing colors with suitable bumps and dents. By "equivalent" we mean that for every valid tiling by the polygons there is an isometry $\alpha$ that maps the tiling into another tiling where the tiles are aligned on integer lattice points so that the tiles – if replaced by the corresponding Wang tiles – provide a valid Wang tiling.

Using this construction of bumps and dents we can lift many results of the previous chapter to the case of polygonal prototiles. In particular, by Theorem 4.5 we know that there are aperiodic protosets of polygons, that is, finite sets of polygons that admit valid tilings but none of these tilings have translational symmetry. The bumps and dents prevent any reflectional or rotational symmetries so it is clear that the protosets we obtain only admit tilings without any non-trivial symmetries.