A LINEAR ALGORITHM FOR BRICK WANG TILING

ALEXANDRE DEROUET-JOURDAN, SHIZUO KAJI, AND YOSHIHIRO MIZOGUCHI

Abstract. The *Wang tiling* is a classical problem in combinatorics. A major theoretical question is to find a (small) set of tiles which tiles the plane only aperiodically. In this case, resulting tilings are rather restrictive. On the other hand, Wang tiles are used as a tool to generate textures and patterns in computer graphics. In these applications, a set of tiles is normally chosen so that it tiles the plane or its sub-regions easily in many different ways. With computer graphics applications in mind, we introduce a class of such tileset, which we call *sequentially permissive* tilesets, and consider tiling problems with *constrained boundary*. We apply our methodology to a special set of Wang tiles, called Brick Wang tiles, introduced by Derouet-Jourdan et al. in 2016 to model wall patterns. We generalise their result by providing a linear algorithm to decide and solve the tiling problem for arbitrary planar regions with holes.

1. Introduction

Wang tiles are a class of formal systems introduced by Hao Wang in 1961 [15]. A Wang prototile can be thought of as a square tile with a colour on each side. Once a set of such prototiles is given, copies of prototiles are placed side by side so that the colours of common edges match. Given a finite set of Wang prototiles, the *domino problem* asks whether a whole plane can be tiled with copies of them, and was proved undecidable by R. Berger in 1966 [2]. There are also sets of Wang prototiles which can tile the plane but only aperiodically ([2, 9, 4, 8]).

In practical applications, bounded planar regions are usually considered, with which the corresponding domino problem is obviously decidable. Moreover, sets of Wang prototiles are normally restricted to a permissive class so that there exist many solutions in general. In this case, it is more reasonable to ask for efficient algorithms to decide tilability with constrained boundary and give solutions if they exist. We introduce a graphical formalism in §2 which incorporates various tiling problems. In general, the problem is known to be NP-complete [12], so we restrict ourselves to a certain sub-problem. Namely, we deal with a specific setting which arises in computer graphics applications such as procedural synthesis of textures, height fields, and sampling points [3, 10, 14]. In this case, sets of Wang prototiles are often *sequentially permissive* (Def. 7), and we have a general strategy described in Lemma 9. In §3, we specialise to the tiling problems with the *Brick Wang tiles* (Def. 10) introduced in [6] to generate wall pattern textures. It is proved in [6] that any rectangle region greater than the 2×2 square with any boundary colour condition can be tiled. This result was then formally verified and implemented in [13] using the Coq proof assistant [1]. We generalise the result by giving a complete characterisation of the set of regions which can be tiled with any given boundary constraints. In §4, we provide a linear time algorithm for the Brick Wang tiling problem, with an implementation with source codes at [5].

2. Problem setting

We begin with introducing a type of Wang tiling problem which can be regarded as an *extension problem* or a *tiling problem with boundary constraints*. Then, we consider a class of Wang prototiles having a special property (1) and discuss a strategy for a tiling algorithm (Lemma 9).

We first define the object to be tiled as an enriched graph.

2010 *Mathematics Subject Classification.* Primary 05B45; Secondary 68R10, 52C20.

Key words and phrases. Wang tile, texture generation, graphical model, satisfiability problem .

The second named author was partially supported by JST PRESTO Grant Number JPMJPR16E3, Japan.

Definition 1. A *board* is a finite undirected graph $G = (V \cup \{v_0\}, E)$ with a distinguished vertex v_0 called the *constrainer* which satisfies the following:

- at each $v \in V$, the set of edges $N(v)$ adjacent to v is ordered: $N(v) = \{e_1(v), \dots, e_{\deg(v)}(v)\}$, where $\deg(v)$ is the degree of v
- the full sub-graph on V is connected.

Vertices in V are called *cells*. We often identify a board with its underlying graph when there is no confusion.

Definition 2. A *tiling problem* (G, C, W) consists of

- a board $G = (V \cup \{v_0\}, E)$
- a set of colours C
- a set of *prototiles* $W = \bigcup_{k=1}^{\infty} W_k$, where $W_k \subset C^k$.

A *tiling* of a subset $F \subset E$ is a map $\tau : F \to C$ satisfying $(\tau(e_1(v)), \dots, \tau(e_{\deg(v)}(v))) \in W$ for any $v \in V$ with $N(v) \subset F$. It is said to be *full* if $F = E$. A tiling $\tau' : F' \to C$ is called an *extension* of another tiling $\tau : F \to C$ with $F \subset F'$ if $\tau(e) = \tau'(e)$ for any $e \in F$.

We are particularly interested in finding a full extension $\tau : E \to C$ of a given tiling $\beta : N(v_0) \to C$, which we regard as *boundary constraints*. Edges in $N(v_0)$ are called *constrained legs* and those in $E \backslash N(v_0)$ are called *free legs*. Given boundary constraints $\beta : N(v_0) \to C$, the tiling problem with β is said to be *solved* if we find a full extension of β. A tiling problem is said to be *always solvable* when there exists a full extension $\tau : E \to C$ of any tiling $\beta : N(v_0) \to C$. Given a tiling $\tau : F \to C$, we informally say to *tile* $v \in V$ with $w \in W$ when we can extend τ by assigning $\tau(e_i(v)) = w_i$ $(1 \le i \le \deg(v))$.

Figure 1 shows an intuitive correspondence between our graphical formalisation and the usual Wang tile.

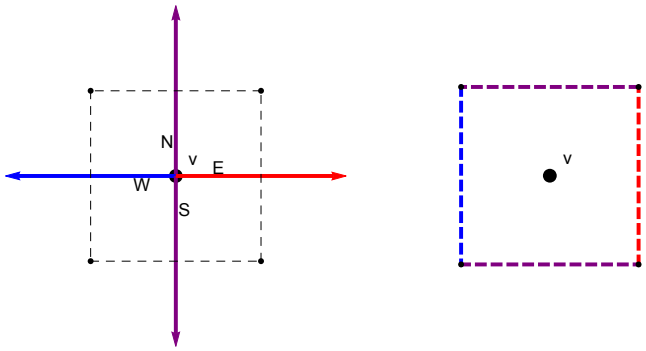

FIGURE 1. Graph vs Wang tile

Remark 3. This formulation encompasses tiling of non-planar graphs such as a surface in the three dimensional space ([7]). For example, a periodic tiling of a planar grid can be regarded as a tiling of a grid graph on the torus.

Example 4. For natural numbers n and m, consider a rectangular grid

$$\begin{aligned} V &= \{v_{i,j} \,|\, 1 \le j \le n, 1 \le i \le m\}, \text{ and} \\ E &= \{(v_{i,j}, v_{i+1,j}) \,|\, 0 \le i \le m, 1 \le j \le n\} \cup \{(v_{i,j}, v_{i,j+1}) \,|\, 1 \le i \le m, 0 \le j \le n\}, \end{aligned}$$

where we set $v_{i,j} = v_0$ for $i = 0$, $i = m + 1$, $j = 0$, or $j = n + 1$. For each $v_{i,j} \in V$, the set of edges $N(v_{i,j})$ adjacent to $v_{i,j}$ consists of four edges $e_E(v_{i,j}) = (v_{i,j}, v_{i+1,j})$, $e_N(v_{i,j}) = (v_{i,j}, v_{i,j+1})$, $e_W(v_{i,j}) = (v_{i,j}, v_{i-1,j})$, and $e_S(v_{i,j}) = (v_{i,j}, v_{i,j-1})$. The order of $N(v_{i,j})$ is given by e_E, e_N, e_W, and e_S. These data define a board $G_{m,n} = (V \cup \{v_0\}, E)$.

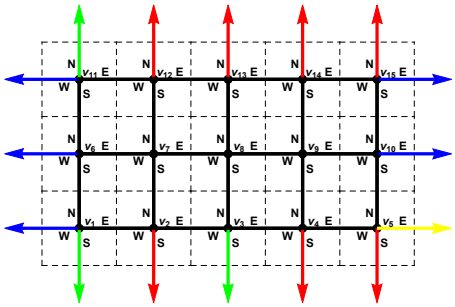

FIGURE 2. Boundary constraints

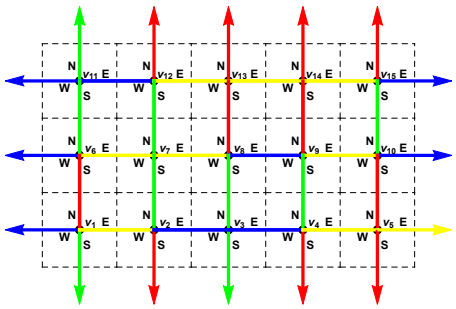

FIGURE 3. A full extension

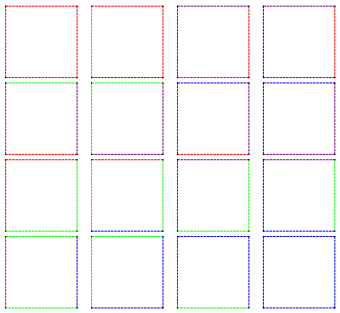

FIGURE 4. Wang prototiles in W_E

Let $C = \{Red, Blue, Purple, Green, Yellow\}$, W_E be as in Figure 4, and consider the tiling problem $(G_{5,3}, C, W_E)$. We set boundary constraints $\beta : N(v_0) \to C$ as in Figure 2. We note that edges connected to v_0 are denoted by arrows in the figure. Also we consider double edges between v_0 and $v_{1,1}$, $v_{5,1}$, $v_{1,3}$ and $v_{5,3}$ which may have different colours. An example of full extension of β is given by $\tau : E \to C$ defined as in Figure 3.

Example 5. Here we give an example of a board which is not a simple graph. For natural numbers n and m, we consider again a rectangular region

$$V = \{v_{i,j} \mid 1 \leq j \leq n, 1 \leq i \leq m\}.$$

This time however the edge set is given as follows: for each $v_{i,j} \in V$, the set of adjacent edges is given by

$$N(v_{i,j}) = \{e_{ENE}, e_{NE}, e_{NNE}, e_{NNW}, e_{SE}, e_{WNW}, e_{WSW}, e_{SW}, e_{SSW}, e_{SSE}, e_{NW}, e_{ESE}\},$$

where the target of each edge is respectively

$$v_{i+1,j}, v_{i+1,j+1}, v_{i,j+1}, v_{i,j+1}, v_{i-1,j+1}, v_{i-1,j}, v_{i-1,j}, v_{i-1,j-1}, v_{i,j-1}, v_{i,j-1}, v_{i+1,j-1}, v_{i+1,j}.$$

(See Figure 5.) Here our convention is $v_{i,j} = v_0$ for $i = 0$, $i = m + 1$, $j = 0$ or $j = n + 1$. These data define a board $G'_{m,n} = (V \cup \{v_0\}, E = \bigcup_{v \in V} N(v))$. The **corner Wang tiling** introduced in [11] is a special instance of tiling problem $(G'_{m,n}, C, W_C)$, where $W_C = \{(c_i) \in C^{12} \mid c_{3k+1} = c_{3k+2} = c_{3k+3}, 0 \leq k \leq 3\}$. In Figure 5, we show an example of a correspondence between our graphical formalisation and the corner Wang tile in [11]. Figure 6 illustrates a solution to the tiling problem $(G'_{2,3}, C, W_C)$ with $C = \{Red, Blue\}$.

As mentioned in [11], a corner Wang tiling is equivalent to an ordinary Wang tiling with more colours. Let $\tilde{C} = \{Red, Blue, Purple, Green\}$. We think of a purple edge having blue and red ends, and a green edge having red and blue ends. The tiling problem $(G_{2,3}, \tilde{C}, W_E)$ defined in Example 4 is equivalent to $(G'_{2,3}, C, W_C)$. The solution of $(G_{2,3}, \tilde{C}, W_E)$ in Figure 6 corresponds to a solution of $(G'_{2,3}, C, W_C)$ in Figure 7.

Definition 6. For a board $G = (V \cup \{v_0\}, E)$ and its sub-graph (U, F), where $U \subset V$, the *restriction* $G|_{(U,F)}$ of G is a board obtained by contracting the complement of (U, F) to the constrainer v_0: The vertex set is the union $U \cup \{v_0\}$ and the edge set is

$$E|_{(U,F)} = F \cup \{(u, v_0) \mid u \in U, (u, v) \in E \setminus F\}.$$

That is, an edge not belonging to F is cut into two pieces and each of them is reconnected to v_0. For a full sub-graph of G on $U \subset V$, we abbreviate the corresponding restriction as $G|_U$.

In many applications of Wang tiling in computer graphics, the set of Wang prototiles often comes equipped with a nice property, which is rather permissive in terms of tilability.

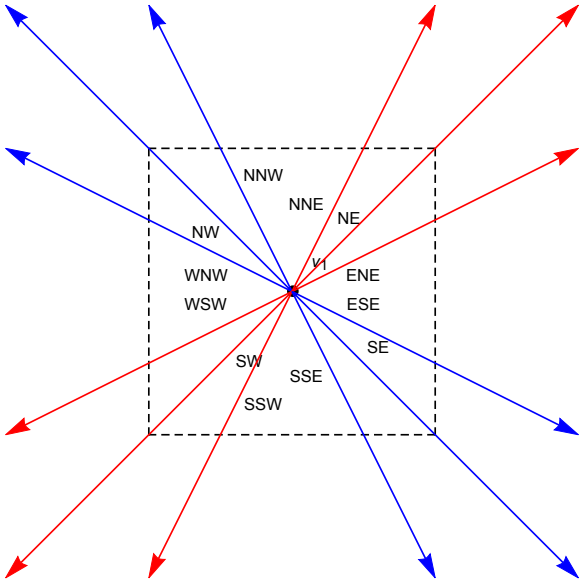

FIGURE 5. Corner Wang tile

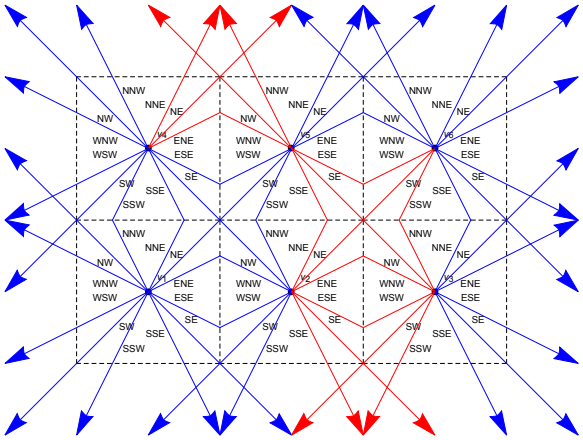

FIGURE 6. Our graphical model for the corner Wang tiling

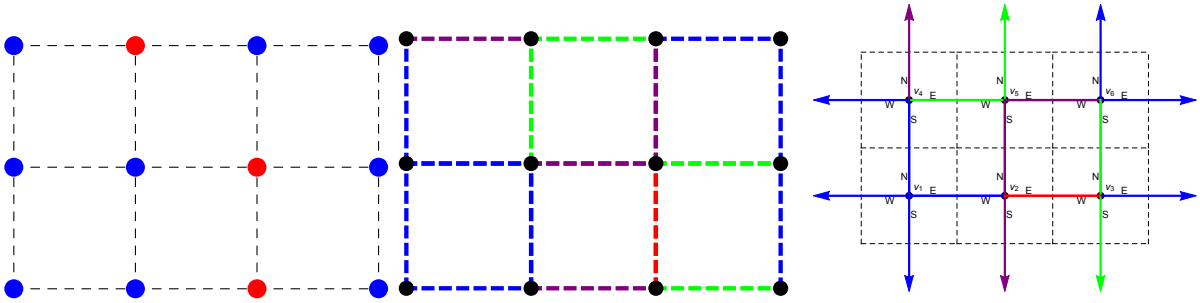

FIGURE 7. Equivalent tiling with different models. Left: corner Wang tiling, Middle: usual Wang tiling, Right: our graphical model for the usual Wang tiling

Definition 7. For a natural number k, a set W of prototiles is said to be *k-sequentially permissive* if the following holds:

$$\pi_{\bar{i}} : W_k \to C^{k-1} \text{ is surjective for all } 1 \le i \le k, \tag{1}$$

where $\pi_{\bar{i}}$ is the projection dropping the i-th factor. In other words, a cell with k-legs can always be tiled when it has at least one free leg.

Example 8. Figure 8 shows an example, taken from [3], of a set of prototiles in computer graphics which is 4-sequentially permissive. In their applications, the colours are replaced by parts of an image to generate textures. Essentially, each side of an edge is one half of an image cut on the diagonal. That way, when tiles are placed adjacently, the transition from one tile to another in the picture is smooth. It is used to produce arbitrarily large textures from a small picture. In [3], this technique is applied to create large Poisson distribution of points by replacing the image parts with carefully designed point distributions.

The following lemma provides a general reduction strategy to solve tiling problems:

Lemma 9. Let G be a board with the set of cells V. Let W be a set of prototiles which is k-sequentially permissive (1) for all $k \in \{\deg(v) \mid v \in V\}$. A tiling problem (G, C, W) is always solvable when there exists a restriction $G|_U$ such that $(G|_U, C, W)$ is always solvable.

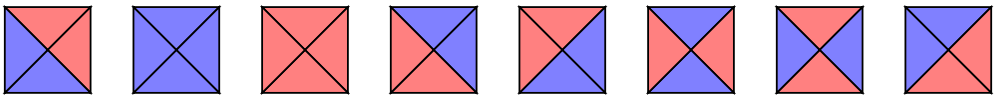

FIGURE 8. Example of a set of Wang tiles used in computer graphics [3].

Proof. Pick a cell $v \in V$ which is adjacent to U, and consider $(G|_{U\cup\{v\}}, C, W)$. The vertex v has at least one free leg, which is connected to U. So we can tile v by (1) for any boundary constraints. Now we are left with $(G|_U, C, W)$ with some boundary constraints, which can be fully extended by assumption. Since V is finite, the assertion follows by induction. □

In practice, we proceed in the "opposite" way; we first construct a spanning tree of $G|_{V\setminus U}$ and tile it from leaves by (1), and then finally cells in U are tiled by assumption. Now the problem is reduced to finding a small sub-region which is always solvable. This is a simple but important observation for the algorithm discussed in §4.

3. Brick Wang tiles

From now on, we focus on a special set of prototiles which we call the *brick Wang tiles*. We give a necessary and sufficient condition for a tiling problem to be always solvable.

The brick Wang tiles were introduced in [6] to model wall patterns. Each tile represents how corners of four bricks meet. It is assumed that the edges of the bricks are axis aligned and that each tile is traversed with a straight line, either vertically or horizontally. For aesthetic reasons, crosses are forbidden, where all four bricks are aligned and the corresponding tile is traversed by two straight lines. In this model, the colour in the Wang tiles indicates the position of the edge of the brick on the side of the tile. The formal definition is given as follows:

Definition 10. The set of *brick Wang prototiles* W_B is defined by

$$W_B = \left\{(c_1, c_2, c_3, c_4) \in C^4 \mid ((c_1 = c_3) \wedge (c_2 \neq c_4)) \text{ or } ((c_1 \neq c_3) \wedge (c_2 = c_4))\right\},$$

where we always assume $\#C \geq 3$.

It is easy to see that W_B is 4-sequentially permissive (1).

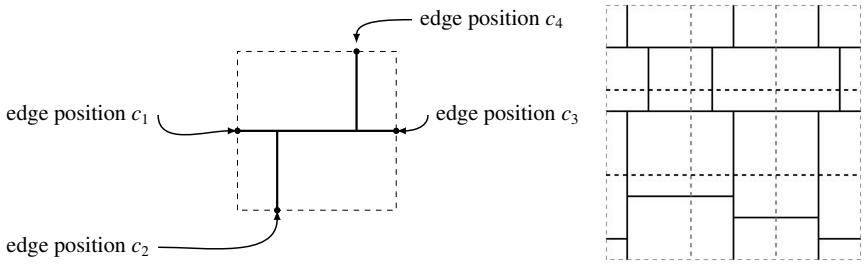

FIGURE 9. A brick Wang tile and a tiling of 3×3-board

In the previous work [6, 13], only rectangular regions are considered. Here, we deal with any planar region possibly with holes. In practice, this allows the user to manually tile a part of the region, and leave the rest to be automatically completed by our algorithm in §4. See Example 16.

Definition 11. Consider the graph $\bar{G} = (V \cup \{v_0\}, E)$, where

- $V = \{(i, j) \in \mathbb{Z} \times \mathbb{Z}\}$
- $E = \{((i, j), (i', j')) \in V \times V \mid |i - i'| + |j - j'| = 1\}$.

The set $N((i,j))$ of edges adjacent to a vertex (i,j) are given ordering by

$$(((i,j),(i+1,j)),((i,j),(i,j+1)),((i,j),(i-1,j)),((i,j),(i,j-1)).$$

A board is said to be *square grid* if it is a restriction of $\bar{G}$ to a finite sub-graph.

We give a simple necessary and sufficient condition for a square grid board G to be always solvable with W_B. A *path* is an ordered sequence of distinct cells $(v_1, v_2, \dots, v_l)$ where consecutive cells are connected by an edge. A *cycle* is an ordered sequence of cells $(v_1, v_2, \dots, v_{l-1}, v_l = v_1)$ where all v_i $(1 \le i \le l-1)$ are distinct, and consecutive cells are connected by an edge. A cell v_i in a path or a cycle is said to be *straight* if edges $e_j, e_k \in N(v_i)$ connected to consecutive cells satisfy $|j-k| = 2$ with the ordering of $N(v_i)$. *Corner* cells in a path or a cycle are those that are not straight.

Lemma 12. A tiling problem (G, C, W_B) for a square grid board G is always solvable when the full sub-graph of G on the cells is a cycle.

Proof. Let $\beta : N(v_0) \to C$ be any colouring of the constrained legs, and $(v_1, \dots, v_l = v_1)$ be the full sub-graph of G on the cells. For a cell v_k, we call the edge connected to v_{k-1} the incoming leg, denoted by $in(v_k)$, and the edge connected to v_{k+1} the outgoing leg, denoted by $out(v_k)$.

First, we can assume there is no straight cells whose constrained legs are given two different colours since they have no effect. Denote by $C_{a,b}$ a corner cell whose constrained leg opposite to the incoming leg is coloured with $a \in C$ and the outgoing leg with $b \in C$. Also, denote by S a straight cell whose constrained legs are coloured with the same colour. These serve as a kind of logical gate:

- $\begin{cases} \tau(in(C_{a,b})) = a \Rightarrow \tau(out(C_{a,b})) \neq b \\ \tau(in(C_{a,b})) \neq a \Rightarrow \tau(out(C_{a,b})) = b \end{cases}$
- $in(S) \neq out(S)$.

We prove the assertion by case by case analysis. Note that we can change the ordering of $(v_1, \dots, v_l = v_1)$ cyclically or in the reverse order.

- Assume that $v_1 = S, v_2 = C_{a,b}$. We set $\tau(out(C_{a,b})) = b$ and tile $(v_3, v_4, \dots, v_{l-1})$ sequentially by (1). Say $out(v_{l-1}) = in(S)$ is coloured with c. Since $\#C \ge 3$, we can choose a colour for $out(S)$ which is different from both c and a to obtain a solution.
- From the previous case, we can now assume that there is no straight cell. Assume that $v_1 = C_{a,b}, v_2 = C_{c,d}$ with $b \neq c$. We set $\tau(out(C_{c,d})) = d$ and tile $(v_3, v_4, \dots, v_{l-1})$ sequentially by (1). If $\tau(out(v_{l-1})) = a$, we choose a colour for $out(C_{a,b})$ which is different from both b and c. If $\tau(out(v_{l-1})) \neq a$, we set the colour of $out(C_{a,b}))$ to be b. In either situation, we obtain a solution.
- Now, the only remaining case is when $(v_1, v_2, \dots, v_{l-1}) = (C_{a_1,a_2}, C_{a_2,a_3}, \cdots, C_{a_{l-1},a_1})$. Note that $l-1$ should be even since the board is a square grid. We set $\tau(in(C_{a_i,a_{i+1}})) = a_i$ and $\tau(out(C_{a_i,a_{i+1}})) \neq a_{i+1}$ for odd i to obtain a solution.

□

Combining Lemma 9 with Lemma 12, we obtain

Theorem 13. A tiling problem (G, C, W_B) for a square grid board G is always solvable if and only if the full sub-graph of G on the cells contains a cycle.

Proof. We only have to show the "only if" part. Given any board G whose full sub-graph on V is a tree, we construct $\beta : N(v_0) \to C$ which cannot be fully extended. We proceed by induction on the number of cells $\#V$. Pick a leaf $v \in V$ and set $G' = G_{V \setminus \{v\}}$. Then, by induction, there exists $\beta' : N'(v_0) \to C$ which cannot be fully extended in G', where $N'(v_0)$ is the edges of G' adjacent to v_0. Let u be the cell connected to v with the edge e. We can choose the value of β for $N(v) \setminus \{e\}$ so that the colour of e is forced to coincide with $\beta'(e)$ in order to tile v. Then, the resulting β cannot be fully extended. □

Remark 14. The main theorem in [6] states that a rectangular board (with no holes) can always be tiled if it contains a 2×2 region. A 2×2 region is a cycle of length four in our language. So the above theorem is a generalisation of the result in [6]. Our proof based on the very simple and general Lemma 9 looks trivial with hindsight. Recall that the proofs in [6, 13] are rather complicated and the theorem is limited. This shows that our graphical framework provides a right way to look at the tiling problems.

4. The tiling algorithm for brick Wang tiles

In this section, we will discuss an algorithm to decide what β can be extended to a full tiling $\tau : E \to C$ and how we can obtain one. An implementation is given at [5].

Fix a tiling problem with a finite planar square grid board G with the Brick Wang tiles W_B and a number of colours $\#C \geq 3$. Fix also boundary constraints $\beta : N(v_0) \to C$. Based on Theorem 13, we design an algorithm to decide solubility and to construct a valid tiling of the problem. The outline goes as follows: let G' be the full sub-graph on the cells

(1) Detect a cycle in G' by performing a depth first traversal of G'. If there is no cycle, use the tree solver given in Lemma 15.
(2) Pick a cell v which is adjacent to but not in the cycle. Construct a tiling in a depth-first manner; that is, tile sequentially from the leaves of a spanning tree rooted at v of the connected component of the complement of the cycle.
(3) Repeat Step 2 until there is no cell other than those in the cycle.
(4) Construct a tiling of the cycle using the procedure of Lemma 12.

Since the degree of a cell is always four, then the complexity of the cycle extraction is linear in the number of cells in G. Similarly, we will see that all other operations are performed in a linear time with regards to the number of cells in the board.

The remaining task is now to decide and solve a tree-structured board. To this extent, we introduce the *condition propagation* that will tell how the border colouring put constraints on the inner legs.

A map $f : C \to \{true, false\}$ of one of the following forms is called a *condition*:

- (c) It must match a given colour c; $f(x) = true \Leftrightarrow x = c$.
- $(\neg c)$ It must not match a given colour c; $f(x) = true \Leftrightarrow x \neq c$.
- $(*)$ It can match any colour; $f(x) = true$ for any x.

To each free leg e, we assign a condition f_e in such a way that a map $\tau : E \to C$ satisfying $\tau(x) = \beta(x)$ for $x \in N(v_0)$ and $f_e(\tau(e)) = true$ for all free legs e gives a full extension of β.

Fix a cell v. If for every free leg of v except for one is assigned a condition, a condition for the last free leg is canonically determined by choosing the weakest one (lower in the above list means weaker). We say that the condition is inferred on the last leg. The precise rule of this is described in Figure 10. For a tree rooted at v, we can propagate the condition recursively by inferring from the leafs to the root v.

Lemma 15. Let G be a board whose full sub-graph on the cells is a tree rooted at v. For a tiling problem (G, C, W_B), boundary constraints $\beta : N(v_0) \to C$ admit a full extension if and only if v can be tiled in such a way that it matches propagated conditions on its legs. Furthermore, any such tiling of v can be extended to a full extension.

Proof. Only if part is trivial by definition.

Observe that the inferred conditions are so that for any choice of a matching colour for a free leg of a cell, there exists at least one colouring of all other free legs which gives a valid tiling of the cell.

Tile v in such a way that it matches propagated conditions on its legs. Consider the connected components obtained by removing v from G, with boundary constraints induced by β and the tiling of v. Now the proof goes by induction on the number of cells. □

The algorithm is illustrated in Figure 11. Since each cell is visited at most four times, it has a linear complexity with respect to the number of cells.

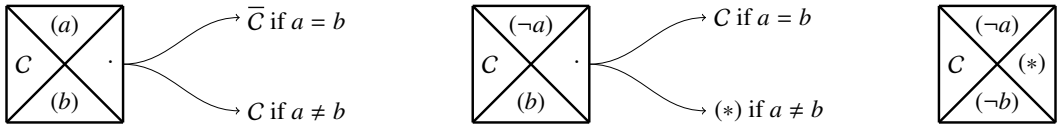

FIGURE 10. Propagation of conditions for cells in a tree. When some of the legs of the cell are on the boundary, we consider them input legs with a condition (c) where c is the colour of the leg. C denotes a given condition and $\overline{C}$ its negation, with $\overline{(a)} = (\neg a)$, $\overline{(\neg a)} = (*)$ and $\overline{(*)} = (*)$. When any of input conditions is $(*)$, then the output is $(*)$.

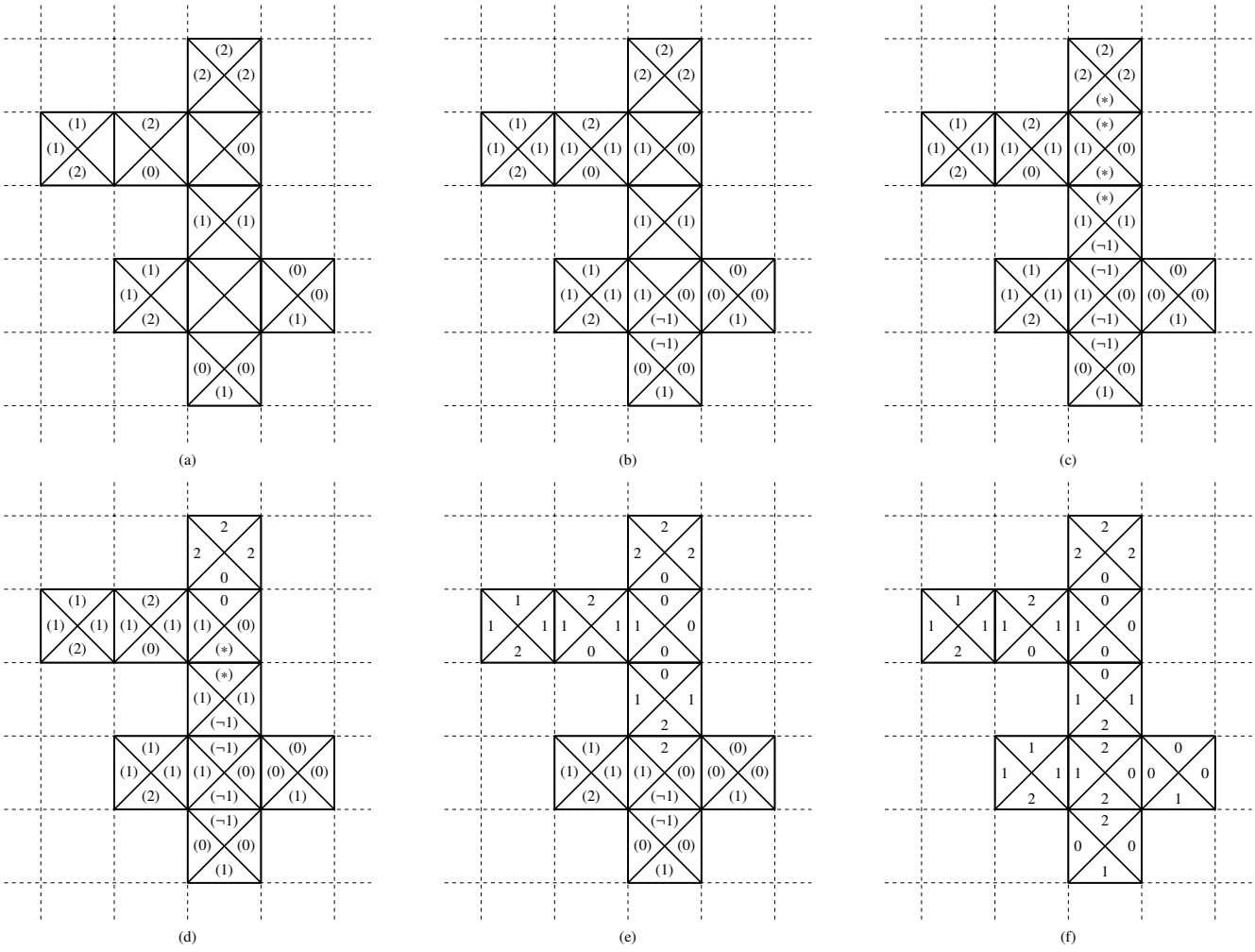

FIGURE 11. Illustration of the tree solver. First, (a) we pick input legs on the leaves of the tree and an output leg on the root of the tree (here it is the top cell). Then, we propagate the conditions to the nodes in the tree (b) and to the root (c). If the conditions on the root are compatible, we propagate the solution back from the root (d) to the nodes (e) and the leaves (f).

Example 16. Figure 12 shows an example of hand-filled regions, completed with an automatically tiled region.

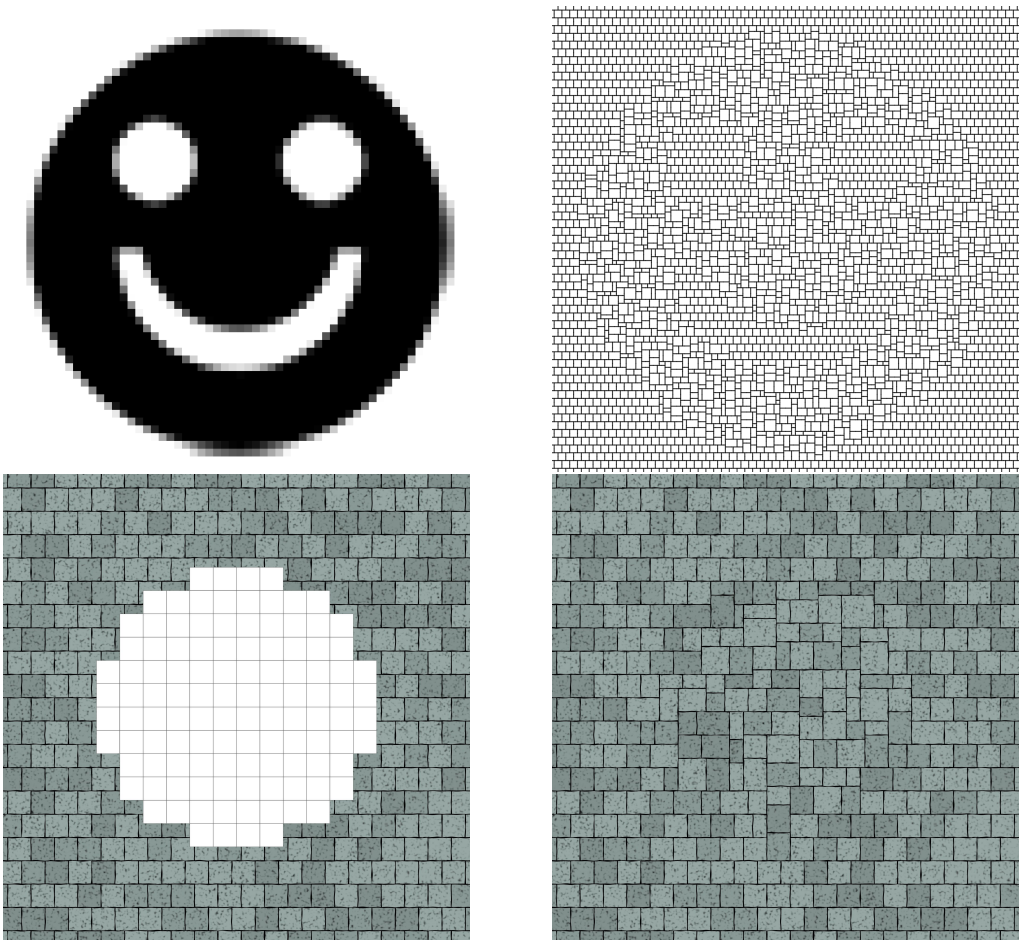

FIGURE 12. Example of a tiling of a given pattern. For the smiley, the tiling of the background, the eyes, and the mouth are specified to be regular (top left indicates the mask) and the rest is tiled automatically (top right). For the wall pattern, a tiling extending the boundary constraints (lower left) is shown (lower right).

5. Conclusion and Future work

In this paper, we have proposed a graph theoretical approach to the Wang tiling problem. We have introduced the notion of sequentially permissive prototiles, which are often observed in computer graphics applications. For the class of Wang prototile sets with this property, we have devised a general strategy to reduce a tiling problem to its sub-problem. Then, we used the general result to give a linear algorithm for the tiling of planar regions with the brick Wang tiles. In future work, we would like to consider tiling problems on hexagonal/triangular and irregular polygonal surface meshes, and also volumetric meshes, which would be useful for texturing and point sampling on meshes. Also, we would like to discuss enumeration of all valid tilings for a given problem.

References

[1] The Coq Proof Assistant, `https://coq.inria.fr/`.
[2] R. Berger. The undecidability of the domino problem. *Memoirs of the American Mathematical Society*, 66:72, 1966.
[3] M. F. Cohen, J. Shade, S. Hiller, and O. Deussen. Wang Tiles for Image and Texture Generation. *ACM Transaction on Graphics*, 22(3):287–294, 2003.
[4] K. Culik. An aperiodic set of 13 Wang tiles. *Discrete Mathematics*, 160(1-3):245–251, 1996.
[5] A. Derouet-Jourdan, Implementation of a linear algorithm for Brick Wang tiling, `https://github.com/KyushuUniversityMathematics/LinearWang`

[6] A. Derouet-Jourdan, Y. Mizoguchi, and M. Salvati. Wang Tiles Modeling of Wall Patterns. In *Mathematical Progress in Expressive Image Synthesis III*, 71–81, Springer Singapore, 2016.
[7] Chi-Wing Fu and Man-Kang Leung. Texture tiling on arbitrary topological surfaces using wang tiles. In *Proceedings of the Sixteenth Eurographics conference on Rendering Techniques (EGSR '05)*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 99–104, 2005.
[8] E. Jeandel and M. Rao. An aperiodic set of 11 Wang tiles. http://arxiv.org/abs/1506.06492, 2015.
[9] J. Kari. A small aperiodic set of Wang tiles. *Discrete Mathematics*, 160(1-3):259–264, 1996.
[10] J. Kopf, D. Cohen, O. Deussen, and D. Lischinski. Recursive Wang Tiles for Real-Time Blue Noise. *ACM Transaction on Graphics*, 25(3):509–518, 2006.
[11] A. Lagae, P. Dutré. An Alternative for Wang Tiles: Colored Edges versus Colored Corners. *ACM Tarnsactions on Graphics*, 25(4):1442–1459, 2006.
[12] H. Lewis. Complexity of solvable cases of the decision problem for predicate calculus. *Proc. 19th Symposium on Foundations of Computer Science*, 35–47, 1978.
[13] T. Matsushima, Y. Mizoguchi, and A. Derouet-Jourdan. Verification of a brick Wang tiling algorithm. in the proceedings of SCSS2016, 2016.
[14] J. Stam. Aperiodic texture mapping. Technical Report R046, European Research Consortium for Informatics and Mathematics (ERCIM), 1997.
[15] H. Wang. Proving theorems by pattern recognition–II. *Bell System Technical Journal*, 40(1):1–41, 1961.

OLM Digital Inc.
E-mail address: alexandre.derouet-jourdan@olm.co.jp

Kyushu University / JST PRESTO
E-mail address, Corresponding author: skaji@imi.kyushu-u.ac.jp

Kyushu University
E-mail address: ym@imi.kyushu-u.ac.jp