

非周期型 Wang タイル集合について (About aperiodic sets of Wang tiles)

水津樹春 (Suizu Kiharu)

February 9, 2022

Abstract

We consider a set consisting of several types of Wang tiles and the colors that make up the tiles, and place these tiles on a plane so that the colors of contiguous edges match. The tiles must not be rotated or mirrored.

Wang tiles was introduced by Wang to study decision procedures for a specific fragment of logic [2]. Wang raised the question of whether there exist an aperiodic tile set, and in 1964 Berger gave an example of an aperiodic tile set. This tileset consisted of 20426 Wang tiles. Knuth simplified Berger's tile set and reduced the number of tiles to 92. Since 6 of which are actually unneeded, it is reduced to 86. In 1971, Robinson published a set of 56 tiles [9], and in 1996, Kari invented a new method for constructing an aperiodic tile set and found an aperiodic tile set of 14 tiles [11]. In 2021, Jeandel and Rao found an aperiodic Wang tileset which has 11 tiles and proved that it is the minimal size of an aperiodic Wang tile set [13]. In this paper, we will briefly present the history of the discovery of this minimal aperiodic Wang tile set.

In addition, we present an implementation of Jeandel and Rao's algorithm [13] for proving the aperiodicity of Wang tilesets of 10 or less using Mathematica program.

概要

Wang タイリングとは、各辺に色付けされている正方形タイルでモデル化されているタイル (**Wang** タイル) を用いたタイリングで、数種の Wang タイルとタイルを構成する色からなる集合 (**Wang** 集合) を考え、それらのタイルを隣り合う辺の色が一致するように平面上に並べる。このときタイルを回転・鏡映させてはいけない。

Wang タイルは 1961 年に特定のロジックフラグメントの決定手順を研究するために Wang によって導入された [2]。Wang は非周期的なタイル集合の存在の有無の問題を提起したが、1964 年に Berger が非周期的なタイル集合の例を挙げたことで存在が確認された。そのタイル集合は 20426 個の Wang タイルで構成されていた。Knuth が Berger の集合を簡略化し、92 個までタイルの数を削減させた。そのうち 6 個は実際には不要で、86 個に削減された。1971 年に Robinson によって 56 個のタイル集合が公開された [9]。1996 年に Kari が非周期的なタイル集合を構築する新しい方法を発明し、14 個のタイルの非周期的なタイル集合を発見した [11]。2021 年に Jeandel と Rao が 11 個の非周期的な Wang タイル集合を発見し、それが最小であることを証明した [13]。本論文では、この最小の Wang タイル集合が見つかるまでの歴史を簡単に紹介する。

さらに、10 個以下の Wang タイル集合の非周期性を証明するための Jeandel ら [13] のアルゴリズムを Mathematica 言語で実装したので紹介する。

Contents

1	Wang タイル	4
1.1	Wang タイルとは	4
1.2	Wang タイルの記号的表現	5
1.3	Wang タイルの操作的表現	6
1.4	Wang 集合への操作	7
2	非周期な Wang タイル集合	8
2.1	非周期的なタイル集合の発見の歴史	8
2.2	非周期的なタイル集合の証明	8
3	Mathematica を使った非周期性の証明	11
3.1	Wang タイリングの基本的な関数	11
3.2	状態遷移図に関する基本的な関数	15
3.3	非周期性を判定する関数	19
4	まとめ	22

1 Wang タイル

1.1 Wang タイルとは

Wang タイルとは色付きの辺を持つ正方形のタイルで、平面のタイリングは、隣接する辺が同じ色を共有するように \mathbb{Z}^2 上の各セルに配置することで構成される。このときタイルを回転・鏡映させてはいけない。タイリングする際、周期的なタイリングと非周期的なタイリングが存在する。タイリングには、Figure1のタイル集合から成る Figure2のような周期的なタイリングと Figure3のタイル集合から成る Figure4のような非周期的なタイリングが存在する。

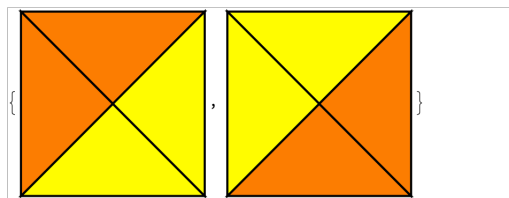


Figure 1: 周期的な Wang タイル集合

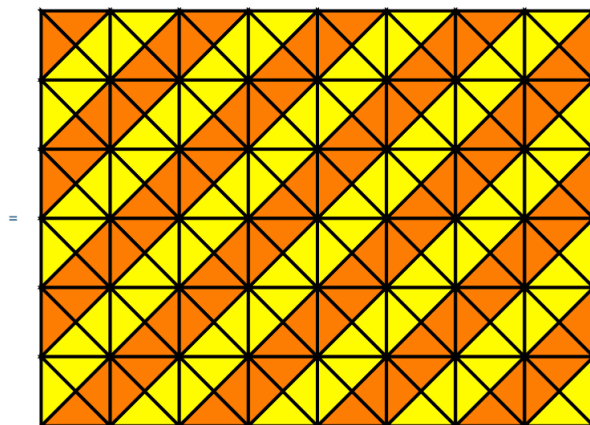


Figure 2: Figure1のタイリングの例

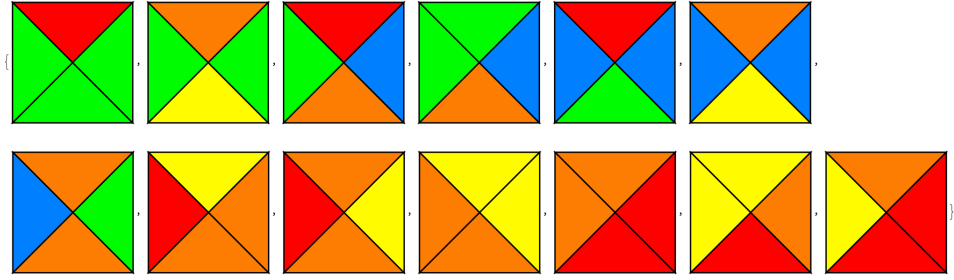


Figure 3: 非周期的な Wang タイル集合 (Culik の 13 個の Wang tiles([12]))

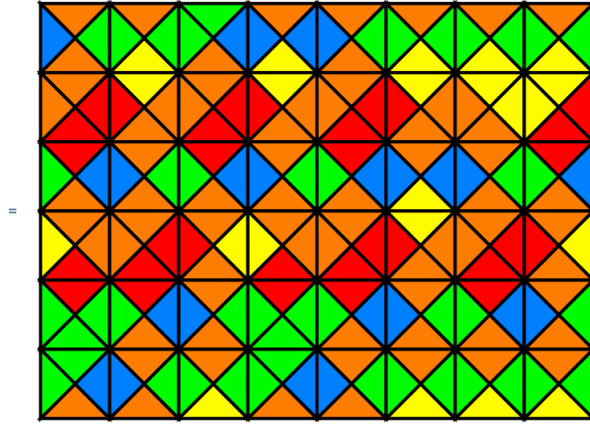


Figure 4: Figure3のタイリングの例

1.2 Wang タイルの記号的表現

- Wang タイルは色付きの辺を持つ単位正方形である。
- H, V を 2 つの有限集合 (それぞれ水平色と垂直色) としたとき、Wang タイル t は $t = (t_w, t_e, t_s, t_n)$ (それぞれ西, 東, 南, 北) と書け、 $H^2 \times V^2$ の元である。
- Wang 集合 $\mathcal{T} = (H, V, T)$ ($T \subseteq H^2 \times V^2$ はタイルの集合) は Wang タイルの集合である。
- T が空集合の時、Wang 集合を空という。

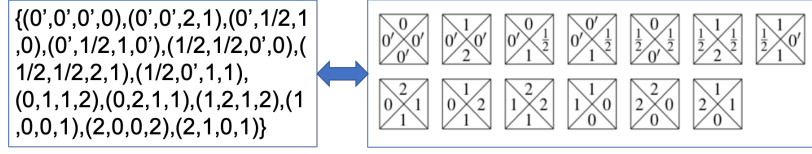


Figure 5: Culik の 13 個の Wang 集合 ([12])(右) と記号的表現 (左)

- $\mathcal{T} = (H, V, T)$ を Wang 集合、 $X \subseteq \mathbb{Z}^2$ とする。 \mathcal{T} による X のタイリングは、隣接する辺が同じ色を持つような X から \mathcal{T} への mapping である。すなわち、それは関数 $f: X \rightarrow T$ s.t $f(x, y)_e = f(x+1, y)_w$ かつ $f(x, y)_n = f(x, y+1)_s$ (任意の $(x, y) \in \mathbb{Z}^2$ に対して)
- タイリング f が周期的 \iff ある $(u, v) \in \mathbb{Z}^2 - (0, 0)$ が存在して s.t 任意の $(x, y) \in \mathbb{Z}^2$ に対して、 $f(x, y) = f(x+u, y+v)$
- タイリング f が非周期的 $\iff f$ が周期的でない。
- Wang 集合が X (または平面) をタイル張りするとは、 \mathcal{T} による X (または平面) のタイリングが存在することである。
- Wang 集合が周期的とは、 \mathcal{T} による周期的なタイリング f が存在する。
- Wang 集合が非周期的とは、平面にタイル張りする場合、 \mathcal{T} による全てのタイリングが周期的でない。

1.3 Wang タイルの操作的表現

- Wang 集合が有限状態トランスデューサー (入力と出力テープを備えた有限状態オートマトン) とみなせる。
- 各 $t = (w, e, s, n) \in T$ は遷移として遷移図を書ける。(Figure6)
- トランスデューサーは状態 w から状態 e への遷移と入力テープ上の文字 s を読むこと、出力状態に n を書き込むことを受理する。

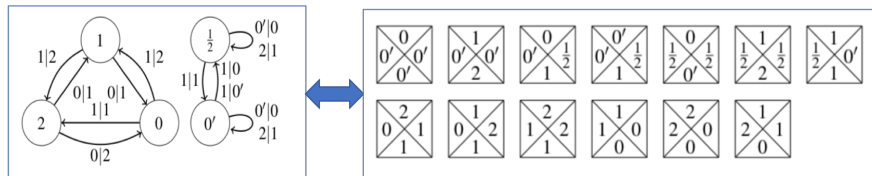


Figure 6: Culik の 13 個の Wang 集合 ([12])(右) と操作的表現 (左)

- アルファベット A の双方向無限列とはすべての $i \in \mathbb{Z}$ に対して $w_i \in A$ となる配列 $(w_i)_{i \in \mathbb{Z}}$ である。
- w, w' : アルファベット A の双方向無限列とする。 w' がトランスデューサによる w の像であるとき $w \mathcal{T} w'$ と書く。(Wang タイル形式では、 w が南の辺の色の双方向無限列、 w' が北の辺の色の双方向無限列となるような行をタイル張りできる場合 $w \mathcal{T} w'$ と書く。)
- トランスデューサ \mathcal{T} の実行 (run) は、全ての $\{i, i+1\} \subset I (I: \mathbb{Z} \text{ の区間})$ に対して $w_i \mathcal{T} w_{i+1}$ となるような双方向無限列 $(w_i)_{i \in I}$ の (おそらく無限または双方向無限の) 配列である。
- (H, V, T) が平面にタイル張りする $\iff (H, V, T)$ の双方向無限実行が存在する。
- コンパクト性により、 (H, V, T) の双方向無限実行が存在する場合に限り、 (H, V, T) の無限実行が存在する。

1.4 Wang 集合への操作

定義 1. Wang 集合の合成 $\mathcal{T} \circ \mathcal{T}'$

$\mathcal{T} = (H, V, T), \mathcal{T}' = (H', V, T')$ を 2 つの Wang 集合とする。

このとき $\mathcal{T} \circ \mathcal{T}'$ は Wang 集合 $(H \times H', V, T'')$ である。

$(T'' = \{((w, w'), (e, e'), s, n') : (w, e, s, n) \in T, (w', e', s', n') \in T', s' = n\})$

$$\mathcal{T}^k (k \in \mathbb{N} - \{0\}) = \begin{cases} \mathcal{T} & k = 1 \text{ のとき} \\ \mathcal{T}^{k-1} \circ \mathcal{T} & \text{その他} \end{cases} \quad (1)$$

これらの定義を用いると以下の変形ができる

- Wang 集合 \mathcal{T} が平面全体にタイリングできない \iff すべての k に対して $w_k \mathcal{T} w_{k+1}$ となるような、双無限方向列 $(w_k)_{k \in \mathbb{N}}$ が存在しない。
- Wang 集合 \mathcal{T} は周期的 \iff ある双方向無限列 w と正の整数 k が存在 s.t. $w \mathcal{T}^k w$

また以下の操作を定義する。

- 回転: \mathcal{T}^{tr} を (V, H, T') ($T' = \{(s, n, e, w) | (w, e, s, n) \in T\}$) とする。
この操作は、タイルセットの 90 度の回転に対応する。

- 簡略化: $s(\mathcal{T})$ を、 \mathcal{T} による（双無限）ライン行のタイリングで使用できないタイルを \mathcal{T} から削除する操作とする。
特に、 $s(\mathcal{T})$ が空である \iff 双無限語 w, w' が存在しない s.t. $w \mathcal{T} w'$

2 非周期的な Wang タイル集合

2.1 非周期的なタイル集合の発見の歴史

Wang タイルは 1961 年に Wang によって、一階述語論理の $\forall \exists \forall$ フラグメントの決定可能性を研究のために導入された。

Wang は非周期的なタイル集合は存在しないと推測していたが、Berger が非周期的なタイル集合の例を挙げたことで存在が確認された。

Berger の最初の非周期的なタイル集合は非常に多くのタイルで構成されていたが、Berger 自身が減らし、その後他の人が削減した結果、現在では 11 個の非周期的なタイル集合が見つかっており、それが最小であることも発見者の Jeandel と Rao によって証明されている。

その Wang タイリングの 60 年の歴史を表にすると以下のようなになる。

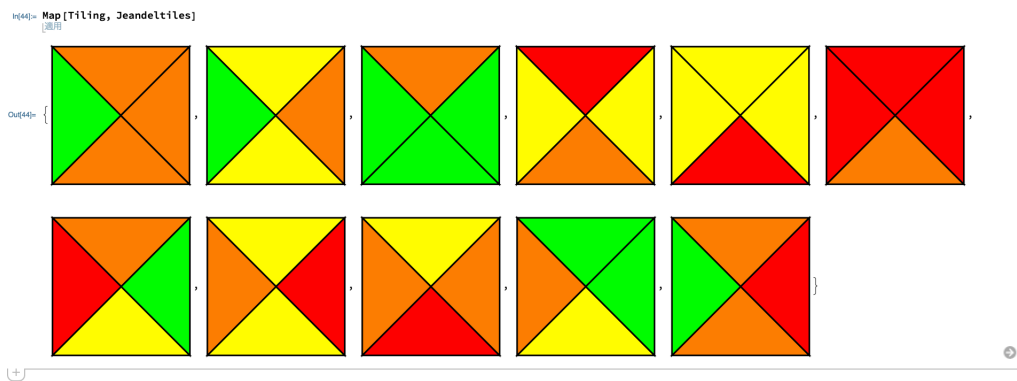


Figure 7: Jeandel, Rao の 11 個の非周期の Wang 集合 $\mathcal{T}'([13])$

2.2 非周期的なタイル集合の証明

現在次の 2 つが [13] で証明されている。

1. 11 個の Wang tiles で構成される新しい非周期的タイル集合を提示
2. このタイル集合が最小であることの証明（11 個未満の Wang タイルは非周期的でないことの証明）

Table 1: 非周期的なタイル集合発見の歴史

西暦	発見者	タイル数	備考
1961	Wang[2]	-	Wang タイルの提案
1964	Berger[4]	20426	非周期的なタイルの存在を証明 (B1)
1964	Berger[3]	104	B1 の簡略化 (論文に誤りがあり 103 タイルに)
1968	Kunth[5]	92	B1 の簡略化 (6 個は実際には不要で 86 タイルに)
1966	Lauchli[6]	40	1975 年に Wang の論文に掲載
1967	Robinson[7]	104	(R1)
1980	Poizat[8]	52	R1 の簡略化
1971	Robinson[9]	56	R1 の簡略化であり、その後 R.Penrose のアイデアで 32 に)
1987	Grunbaum,Shephard[10]	24	R.Penrose のアイデアにより削減
1977	Robinson	24	Ammann によるタイリング法
1978	Ammann[9]	16	
1996	Kari[11]	14	新しい方法
1996	Culik[12]	13	Kari と同じ方法
2021	Jeandel,Rao[13]	11	最小の非周期 Wang タイル集合 (Figure7)

特に 10 個以下のタイル集合が非周期的でないことはコンピューターを用いて証明されている。

アルゴリズムの考え方

10 タイル以下のすべての Wang 集合を生成し、それぞれをテストして非周期的かどうかを確認することで、10 個以下のタイル集合が非周期的でないことを証明する。

つまり、次の 2 つの工程に分けて考えられる。

- (1).10 タイル以下のすべての Wang 集合を生成する。
- (2).Wang 集合の非周期性のテストする。

(1):10 タイル以下のすべての Wang 集合を生成するアルゴリズム

Wang 集合が非周期的でないことを証明できる最小限の判断基準を定め、その判断基準を満たすような Wang 集合を全て生成する。

このとき、直接 Wang 集合を生成するのではなく、遷移のラベルを無視した状態遷移図の基礎となるグラフ G を考える。

このときグラフ G は Wang タイルの水平色もしくは垂直色の色の割り当てに対応する。最小限の判断基準を次で述べる補題 4 とする。

補題 4

補題 4. \mathcal{T} を Wang 集合、 G を対応するグラフとする。

- u から v へのエッジ（したがってタイル/遷移）があり、 v から u へのパスがないような 2 つの頂点/状態/色 $u, v \in G$ が存在する $\rightarrow \mathcal{T}$ は最小限の非周期的でない
- G にサイクルである強連結成分が含まれている $\rightarrow \mathcal{T}$ は最小限の非周期的でない
- G の頂点が 1 つしかない $\rightarrow \mathcal{T}$ は非周期的でない
- G のエッジの数と頂点の数の差が 2 未満 $\rightarrow \mathcal{T}$ は最小限の非周期的でない

与えられた $n \leq 10$ に対して、補題の判断基準を満たす n 個のエッジと最大で $n - 2$ 個の頂点を持つすべてのグラフの集合 \mathcal{G} を生成する。

こうして生成されたグラフの集合 \mathcal{G} のすべての G_1 と G_2 について、 G_1 を水平色、 G_2 を垂直色の割り当てることで Wang 集合を生成する。

(2):Wang 集合の非周期性のテストするアルゴリズム

次のどちらかを満たすとき、Wang 集合は非周期的でない。

1. $s(\mathcal{T}^k)$ が空となるある k が存在する。：これは、 $w\mathcal{T}^kw'$ であるような単語 w, w' が存在しないこと、つまり \mathcal{T} が平面上にタイリングできないことを意味する。

2. \mathcal{T}^k が周期的である k が存在する。： $w\mathcal{T}^kw$ であるような単語 w が存在すること、つまり \mathcal{T} が周期的であることを意味する。

よって非周期性をテストするアルゴリズムは次のように考えられる。

アルゴリズム

各 k について、 \mathcal{T}^k を生成し、2つのケースのいずれかが発生するかどうかをテストする。いずれかが発生した時点で非周期的でないことがわかる。

メモリは有限であるため、非周期的であるかは判断できない。
途中で終わらない Wang セットは別途深く調べる必要がある。

(1) で生成した Wang 集合を (2) でテストすることで 10 タイル以下の Wang 集合が非周期的でないことが証明されている。

3 Mathematica を使った非周期性の証明

Mathematica を用いて、補題 4 を満たす n 個のエッジと最大で $n - 2$ 個の頂点を持つすべてのグラフの集合 \mathcal{G} を生成するプログラムと、Wang 集合の非周期性を判定するプログラムを作成した。

Mathematica を用いることでグラフの可視化やタイリングの図面化が容易にした。

周期的であると判定された Wang 集合がどのような一周期を持つかを可視化できるようにした。

3.1 Wang タイリングの基本的な関数

色の定義: $\text{Color}[i, c]$: c 個の色を定めて、その中から i 番目の色を取り出す。

Color

```
Color[i_, c_] := Module[{x, r, g, b},
  x = i / (c + 1);
  r = Max[0, Min[1, 3 - 6x], 3x - 2];
  g = Max[0, Min[1, 3x, 2.5 - 3x]];
  b = Max[0, Min[1, 6x - 3, 6 - 6x]];
  RGBColor[r, g, b]]
```

タイルを表示する関数: $\text{Tiling}[\text{tile}]$: 4 辺の色の情報をもつリストからタイルを表示する。

Figure8 の左のタイルは、リスト $\{\{3\}, \{1\}, \{2\}, \{0\}\}$ から生成されており、リストの 4 つの成分はそれぞれタイルの西、東、南、北の色に対応している。
Figure8 の右のタイルのように 1×1 より大きいタイルも表示できる。この際、平面上に並べる際には必要のない情報なため、タイルの内側は白く塗られる。

Tiling

```

Tiling[tile_]:=Module[{tw,te,ts,tn,n,m,c,TriT,TriB,TriL,TriR},
tw= tile[[1]];
te= tile[[2]];
ts= tile[[3]];
tn= tile[[4]];
n=Length[tw];(*縦の長さ*)
m=Length[tn];(*横の長さ*)
c=5; (*c:色の数値の最大値*)
TriT[s_,t_]:=If[s==1,{Color[tn[[t]],c],Polygon[{2t-2,2-2s},{2t,2-2s},{2t-1,1-2s}]}],{RGBColor[1,1,1],Polygon[{2t-2,2-2s},{2t,2-2s},{2t-1,1-2s}]}];(*上辺の三角形*)
TriB[s_,t_]:=If[s==n,{Color[ts[[t]],c],Polygon[{2t-2,-2s},{2t,-2s},{2t-1,1-2s}]}],{RGBColor[1,1,1],Polygon[{2t-2,-2s},{2t,-2s},{2t-1,1-2s}]}];(*下辺の三角形*)
TriL[s_,t_]:=If[t==1,{Color[tw[[s]],c],Polygon[{2t-2,-2s},{2t-2,2-2s},{2t-1,1-2s}]}],{RGBColor[1,1,1],Polygon[{2t-2,-2s},{2t-2,2-2s},{2t-1,1-2s}]}];(*左辺の三角形*)
TriR[s_,t_]:=If[t==m,{Color[te[[s]],c],Polygon[{2t,-2s},{2t,2-2s},{2t-1,1-2s}]}],{RGBColor[1,1,1],Polygon[{2t,-2s},{2t,2-2s},{2t-1,1-2s}]}];(*右辺の三角形*)
Graphics[{EdgeForm[Thick],
Table[Table[{TriT[s,t],TriB[s,t],TriL[s,t],TriR[s,t]},{t,m}],{s,n}]]]

```

In[10]:= Map[Tiling, {{{3}, {1}, {2}, {0}}, {{3, 2}, {4, 1}, {3, 1, 1}, {1, 2, 3}}}]
適用

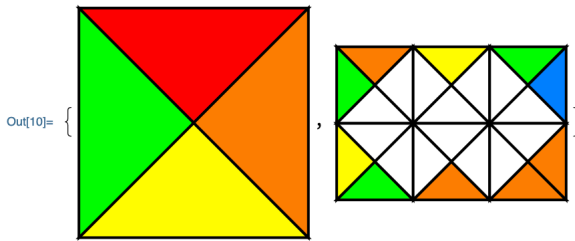


Figure 8: Tiling を用いたタイルの表示

タイルの合成 (垂直方向): $\text{Tilecompositionv}[\text{tiles1_}, \text{tiles2_}]$: 2 つの Wang 集合 $\text{tiles1}, \text{tiles2}$ に対し、 tiles1 の tile を南側、 tiles2 の tile を北側として Wang タイリングのルールにしたがって 2 つを南北に並べたものの集合で、Wang 集合の合成 $\text{tiles1} \circ \text{tiles2}$ に相当する。

```

Tilecompositionv —
Tilecompositionv[tiles1_,tiles2_] := Module[{tiles={}},

fcompositionv[t1_,t2_] :=
{Join[t2[[1]],t1[[1]]},Join[t2[[2]],t1[[2]]],t1[[3]],t2[[4]]};
For[i=1,i<=Length[tiles1],i++,
For[j=1,j<=Length[tiles2],j++,
If[tiles2[[j]][[3]]==tiles1[[i]][[4]],
tiles=Append[tiles,fcompositionv[tiles1[[i]],tiles2[[j]]]]
]
]
];
tiles
]

```

タイルの合成 (水平方向) : $\text{Tilecompositionh}[\text{tiles1_}, \text{tiles2_}]$: 2 つの Wang 集合 $\text{tiles1}, \text{tiles2}$ に対し、 tiles1 の tile を西側、 tiles2 の tile を東側として Wang タイリングのルールにしたがって 2 つを東西に並べたものの集合で、Wang 集合となる。

Tilecompositionh

```
Tilecompositionh[tiles1_,tiles2_]:= Module[{tiles={}},  
  
fcompositionh[t1_,t2_]:=  
{t1[[1]],t2[[2]],Join[t1[[3]],t2[[3]],Join[t1[[4]],t2[[4]]];  
For[i=1,i<=Length[tiles1],i++,  
For[j=1,j<=Length[tiles2],j++,  
If[tiles2[[j]][[1]]==tiles1[[i]][[2]],  
tiles=Append[tiles,fcompositionh[tiles1[[i]],tiles2[[j]]]  
]  
]  
];  
tiles  
]
```

タイル集合の簡略化:TileSimplify[tiles0_]:Wang 集合上で水平方向の合成に使用できないタイルを削除する。

この関数を Wang 集合が変化しなくなるまで適用することで、タイルの簡略化が行われる。

TileSimplify

```
TileSimplify[tiles0_]:=Module[{i=1,tiles10},  
tiles10 =DeleteDuplicates [tiles0];  
While[i<=Length[tiles10],  
If[ MemberQ[Transpose[tiles10][[2]],tiles10[[i]][[1]]]&&  
MemberQ[Transpose[tiles10][[1]],tiles10[[i]][[2]]],  
i++,tiles10=Delete[tiles10,i]  
]  
];  
tiles10  
]
```

3.2 状態遷移図に関する基本的な関数

補題 4

補題 4. \mathcal{T} を Wang 集合、 G を対応するグラフとする。

- u から v へのエッジ（したがってタイル/遷移）があり、 v から u へのパスがないような 2 つの頂点/状態/色 $u, v \in G$ が存在する $\rightarrow \mathcal{T}$ は最小限の非周期的でない
- G にサイクルである強連結成分が含まれている $\rightarrow \mathcal{T}$ は最小限の非周期的でない
- G の頂点が 1 つしかない $\rightarrow \mathcal{T}$ は非周期的でない
- G のエッジの数と頂点の数の差が 2 未満 $\rightarrow \mathcal{T}$ は最小限の非周期的でない

Generategraphs[n]:補題 4 の判断基準を満たすような n 個のエッジをもつグラフを全て生成するプログラム

- 補題 4 の (1): Mathematica 上の関数 ConnectedComponents[g] と WeaklyConnectedComponents[g] を用いて、グラフの強連結成分と弱連結成分を求め、その 2 つが一致するとき (1) の基準を満たす。
- 補題 4 の (2): グラフの任意の強連結成分 SCC に対して、SCC を構成するあるノードが出次数 2 以上のとき、(2) の基準を満たす。
- 補題 4 の (3),(4): グラフのエッジの数を n としたとき、頂点の数 v を $2 \leq v \leq n - 2$ の範囲で動かすことで、(3),(4) の基準を満たす。

Generategraphs

```

Generategraphs[n.]:=Module[{graphs,test=100},
graphs = {};(*補題 4 の仮説を満たすグラフをここに追加していく*)
For[v=2,v<=n-2,v++, (*頂点の数を指定：v, 補題 4 (3),(4)*)
part=Table[1,v-1];
part= Append[part,n-v+1];
For[t=0,t<=10000,t++,(*各辺の始点を動かすためのループ*)
(*edge の記述*)
from=Flatten[Table[Table[i,part[[i]]],{i,v}]];(*各辺の始点のリスト*)
to= Tuples[Range[1,v],n];(*各辺の終点のリストの全パターン*)
For[k=1,k<= Length[to],k++,(*各辺の終点を動かすループ*)
graph =Table[{from[[i]],to[[k]][[i]]},{i,n}];
(*対称性による削除*)
For[l=1,l<=n-1,l++,
ng=graph[[l,1]]==graph[[l+1,1]]&& graph[[l,2]]!=graph[[l+1,2]];
If[ng,Break[]];
];
If[ng,Continue[]];

(*弱連結成分 → 強連結成分の判定, 補題 4 (1)*)

ok=SubsetQ[Transpose[graph][[2]],Range[1,v]];

If[Not[ok],Continue[]];
array=ConstantArray[0,{v,v}];
For[i=1,i<=n,i++,
array[ [graph[[i,1]],graph[[i,2]] ] ]++
]; (*隣接行列の作成*)
adjacencygraph=AdjacencyGraph[array];
ok=Sort[ConnectedComponents[adjacencygraph]](*グラフの強連結成分*)
==Sort[WeaklyConnectedComponents[adjacencygraph]]
(*グラフの弱連結成分*);
If[Not[ok],Continue[]];

```


Generategraphs の続き

(*サイクルの有無の確認、補題 4 (2)*)

```
set={};
For[i=1,i<=v,i++,
If[Length[Select[graph,#[[1]]==i&]]>=2,AppendTo[set,i]];
(*set:出次数が 2 以上の頂点の集合*)
For[l=1,l<= Length[ConnectedComponents[adjacencygraph]],l++,
ng=Intersection[ConnectedComponents[adjacencygraph][[l]],set]=={};
(*各強連結成分と set のがない時はサイクルとなるため NG*)
If[ng,Break[]]
];
```

If[ng, Continue[]];

(*グラフを置換した後ソートして対称性による被りを取り除く*)

```
a=Permutations[Range[v]];
permutedfroms=Table[Table[a[ [i,graph[[j,1]] ] ],{j,n}},{i,Length[a]}];
permutedtos = Table[Table[a[ [i,graph[[j,2]] ] ],{j,n}},{i,Length[a]}];
permutedgraphs = Table[
Sort[Transpose[{permutedfroms[[i]],permutedtos[[i]]}]]
,{i,Length[a]}];
ok =Intersection[permutedgraphs,graphs]== {};
If[Not[ok],Continue[]];
permutedgraphs = Table[
Sort[Transpose[{permutedtos[[i]],permutedfroms[[i]]}]]
,{i,Length[a]}];
ok =Intersection[permutedgraphs,graphs]== {};
If[Not[ok],Continue[]];
```

Generategraphs の続き 2

```
(*グラフの追加*)
graphs=Append[graphs,graph]
];
(*part(from) の変更*)
If[part[[v]]-1<part[[v-1]],part[[v]]-;part[[v-1]]++,
For[j=v-1;s=part[[v]]+part[[v-1]]-1,part[[j]]>=part[[v]]-
1&&j<0,j--,s=s+part[[j]]];
If[j<=0,Break[]];
x=part[[j]]+1; part[[j]]=x; j++;
While[j<v,part[[j]]<=x;s=s-x;j++];
part[[v]]=s
];

If[t>=10000,Append[graphs,error]];
];
];
graphs
]
```

補題 4 の基準を満たすような 5 個のエッジを持つグラフは次のように表される。

```
In[17]:= qq = Generategraphs[5]
Out[17]= {{ {1, 2}, {2, 1}, {2, 1}, {2, 1}, {2, 1}}, { {1, 2}, {2, 1}, {2, 1}, {2, 1}, {2, 2}},
{ {1, 2}, {2, 1}, {2, 1}, {2, 2}, {2, 2}}, { {1, 2}, {2, 1}, {2, 2}, {2, 2}, {2, 2}},
{ {1, 1}, {1, 1}, {2, 2}, {2, 2}, {2, 2}}, { {1, 1}, {1, 2}, {2, 1}, {2, 1}, {2, 2}},
{ {1, 1}, {1, 2}, {2, 1}, {2, 2}, {2, 2}}, { {1, 2}, {1, 2}, {2, 1}, {2, 1}, {2, 1}},
{ {1, 2}, {1, 2}, {2, 1}, {2, 1}, {2, 2}}, { {1, 2}, {2, 3}, {3, 1}, {3, 1}, {3, 1}},
{ {1, 2}, {2, 3}, {3, 1}, {3, 1}, {3, 2}}, { {1, 2}, {2, 3}, {3, 1}, {3, 1}, {3, 3}},
{ {1, 2}, {2, 3}, {3, 1}, {3, 2}, {3, 2}}, { {1, 2}, {2, 3}, {3, 1}, {3, 2}, {3, 3}},
{ {1, 2}, {2, 3}, {3, 1}, {3, 3}, {3, 3}}, { {1, 3}, {2, 3}, {3, 1}, {3, 1}, {3, 2}},
{ {1, 3}, {2, 3}, {3, 1}, {3, 2}, {3, 3}}, { {1, 2}, {2, 1}, {2, 1}, {3, 3}, {3, 3}},
{ {1, 2}, {2, 1}, {2, 2}, {3, 3}, {3, 3}}, { {1, 2}, {2, 1}, {2, 3}, {3, 1}, {3, 2}},
{ {1, 2}, {2, 1}, {2, 3}, {3, 1}, {3, 3}}, { {1, 2}, {2, 1}, {2, 3}, {3, 2}, {3, 3}},
{ {1, 2}, {2, 2}, {2, 3}, {3, 1}, {3, 1}}, { {1, 2}, {2, 2}, {2, 3}, {3, 1}, {3, 3}},
{ {1, 2}, {2, 3}, {2, 3}, {3, 1}, {3, 1}}, { {1, 2}, {2, 3}, {2, 3}, {3, 1}, {3, 2}} }
```

Figure 9: Generategraphs[5] の実行結果

Figure9の $\{\{1, 2\}, \{2, 1\}, \{2, 1\}, \{2, 1\}, \{2, 1\}\}$ とは、ノード 1 からノード 2 へのエッジ 1 つとノード 2 からノード 1 へのエッジ 4 つで構成されるグラフという意味である。

これをグラフとして可視化させると Figure10のようになる。

```
In[19]:= Map[Graph, Apply[Rule, qq, {2}]]
適用 グラフ 頭... 規則
```

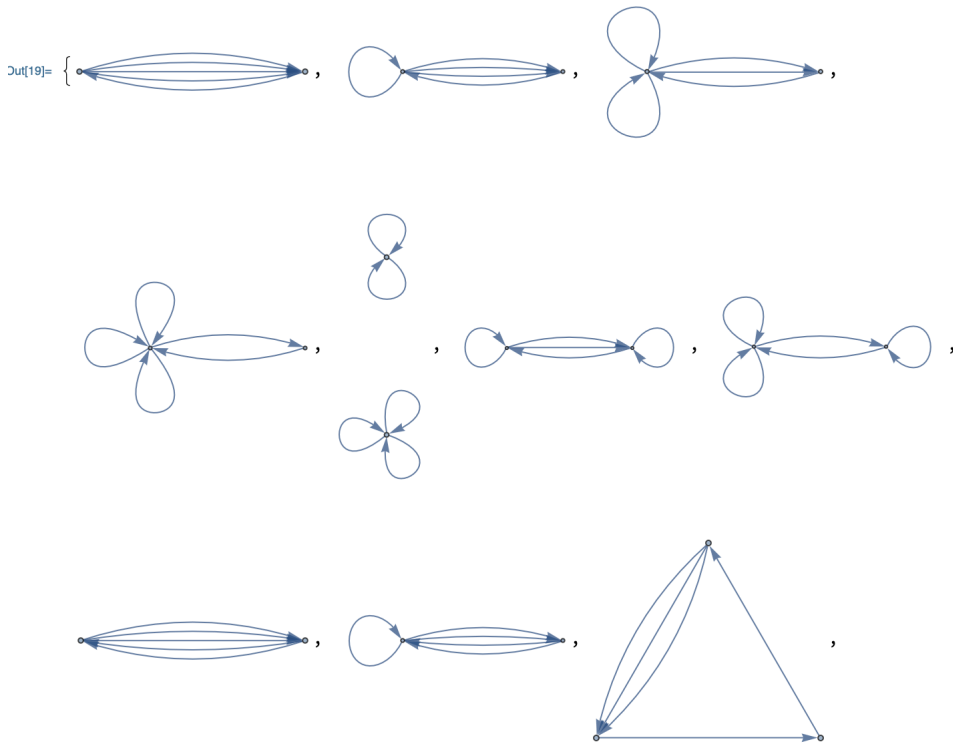


Figure 10: Generategraphs[5] の実行結果のグラフ化 (一部抜粋)

3.3 非周期性を判定する関数

1. $s(\mathcal{T}^k)$ が空となるある k が存在する： $w\mathcal{T}^kw'$ であるような単語 w, w' が存在しない
2. \mathcal{T}^k が周期的である k が存在する： $w\mathcal{T}^kw$ であるような単語 w が存在する

k を 1 から順に動かし、 \mathcal{T}^k を生成し、2つのケースのいずれかが発生するかどうかをテストする。いずれかのケースが発生した場合、非周期的でないことがわかりアルゴリズムを終了する。前者のケースが発生したときはタイリングできないことが、後者が起きたときは周期的であることがわかる。

Nowang[tiles_]:Wang 集合の周期性を判定するプログラム

- 入力 は Wang タイル集合である。

- 平面上に並べられない場合は第一成分に 1 を出力する。
- 周期的であると判定された場合には第一成分に 2 を出力し、第二成分を Periodictile[subtiles_] を適用することでその Wang 集合がどのような一周期を持つかを可視化できる。
- 周期的であると判定できなかった場合第一成分に 0 を出力する。

Nowang

```
Nowang[tiles_] :=
Module[{a=0,newtiles,oldtiles,subtiles,oldsubtiles,simpletiles},
newtiles=tiles;
oldtiles={};
subtiles={};
(*タイルの簡略化を Wang 集合が変化しなくなるまで行う*)
While[newtiles!=oldtiles,oldtiles=newtiles;
newtiles=TileSimplify[newtiles]];
simpletiles= newtiles;

For[k=1,k<100,k++,
(*簡略化の結果 Wang 集合が空になった場合 a=1 を出力し break*)
If[newtiles=={},a=1;Break[]];
(*南北が同色のタイルのみを取り出した集合 subtiles を考える*)
subtiles=Select[newtiles,#[[3]]==#[[4]]&];
oldsubtiles={};
(*subtiles を簡略化し続けても空にならない場合 a=2 を出力し break*)
While[subtiles!=oldsubtiles,oldsubtiles=subtiles;
subtiles=TileSimplify[subtiles]];
If[subtiles!={},a=2;Break[]];
newtiles=Tilecompositionv[newtiles,simpletiles];
(*タイルの簡略化を Wang 集合が変化しなくなるまで行う*)
While[newtiles!=oldtiles,oldtiles=newtiles;
newtiles=TileSimplify[newtiles]]
];
{a,subtiles}
]
```

Periodictile

(*周期的な Wang 集合の一周期の一つを可視化*)

```
Periodictile[subtiles_] := Module[{periodictiles, periodictile},
  periodictiles = {subtiles[[1]]};
  While[Select[periodictiles, #[[1]] == #[[2]] &] == {},
    periodictiles = Tilecompositionh[periodictiles, subtiles];
    periodictile = Map[Tiling, Select[periodictiles, #[[1]] == #[[2]] &]];
  periodictile
]
```

Figure11のようなタイル集合を Nowang[tiles_] を用いて周期性を判定すると Figure12のような結果となる。

Figure12を見ると、第一成分が2を出力しているため、周期的であると判定されている。

また、第二成分に Periodictile[subtiles_] を適用することでその Wang 集合がどのような一周期を持つかを可視化することができた。

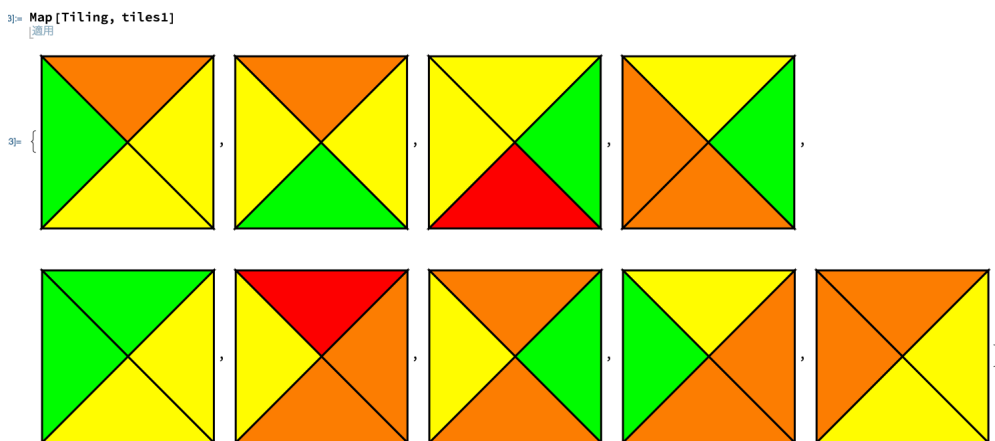
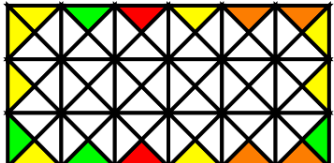


Figure 11: タイル集合 1

```

In[85]:= Nowang[tiles1][[1]]
Out[85]= 2

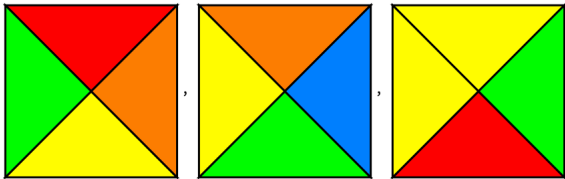
In[86]:= Periodictile[Nowang[tiles1][[2]]]
Out[86]= {

}

```

Figure 12: タイル集合 1 の周期性の判定

Figure13より、次の3つからなるタイル集合は、Nowang[tiles_]で第一成分が1を出力しているため、平面全体にタイリングできないことがわかる。

```

In[24]:= Map[Tiling, tiles2]
Out[24]= {

}

In[27]:= Nowang[tiles2][[1]]
Out[27]= 1

```

Figure 13: タイル集合 2

4 まとめ

1961年にWangが非周期的なタイル集合の存在に関する問題提起から、2021年にJeandelらが最小の非周期的なWangタイル集合を見つけるまでの歴史をまとめた。さらに、Wangタイル集合の可視化やJeandelらのWang集合の非周期性の判定アルゴリズムをMathematica言語で実装し、具体例とともに紹介した。

References

- [1] R.Wille, Restructing lattice theory: an approach based on hierarchies of concept, in: I.Rival, ed., Ordered sets, D.Reidel 445-470, 1982.
- [2] H.Wang, Proving theorems by pattern recognition- II, Best Syst Tech J, 40(1), 1-41, 1961.
- [3] R.Berger, The Undecidability of the Domino Problem. PhD thesis, Harvard University, 1964.
- [4] R.Berger, The Undecidability of the Domino Problem. Number 66 in Memoirs of the American Mathematical Society. The American Mathematical Society, 1966.
- [5] D.E.Knuth, Fundamental Algorithms, volume 1 of The Art of Computer Programming, section 2.3.4.3 The infinity lemma, pages 382 – 385. Addison-Wesley, first edition, 1968.
- [6] H.Wang, Notes on a class of tiling problems. Fundamenta Mathematicae, 82:295 – 305, 1975.
- [7] R.M.Robinson, Seven polygons which permit only nonperiodic tilings of the plane. Notices Amer. Math. Soc, 14:835, 1967.
- [8] B.Poizat, Une théorie finiment axiomatisable et superstable. Groupe d'études de théories stables, 3:1 – 9, 1980.
- [9] R.M.Robinson, Undecidability and Nonperiodicity for Tilings of the Plane. Inventiones Mathematicae, 12(3):177 – 209, 1971.
- [10] B.Grunbaum and G.C.Shephard, Tilings and Patterns. W.H. Freeman, 1987.
- [11] J.Kari. A small aperiodic set of Wang tiles. Discrete Mathematics, 160:259 – 264, 1996.
- [12] K.Culik, II, An aperiodic set of 13 Wang tiles, Discrete Math, 160:245-251, 1996.
- [13] E.Jeandel and M.Rao, An aperiodic set of 11 Wang tiles, Advances in combinatoric, pp.1-37, 2021.
- [14] A.Derouet-Jourdan, S.Kaji and Y.Mizoguchi, A linear algorithm for Brick Wang tiling, Japn J. Indust. Appl. Math. pp 1-13, 2019.