

逆探索に基づく p6 タイリングの生成

Generation of Polyiamonds for p6 Tiling by the Reverse Search

山根 祥悟*

堀山 貴史*

Shogo YAMANE

Takashi HORIYAMA

Polyiamonds are the two dimensional shapes made by connecting n equal-sized equilateral triangles, joined along their edges. In this paper, we propose algorithms to generate polyiamonds for p6 tiling, i.e., those covering the plane by only 6-fold rotations around two rotation centers. The conventional methods are basically trial and error, i.e., they repeat generating polyiamonds and checking whether the shapes have been already generated. Our approach is based on the reverse search, in which we design rules to generate the next. This technique has the following two characteristics: (1) No trial and error, which implies that we can reduce the computation time. (2) No need to store already generated polyiamonds. Thus, we can also reduce the space complexity. We also implement the algorithm and generate 26,875 polyiamonds with $n = 21$ equilateral triangles, while the conventional methods generated 29 polyiamonds with $n = 9$ equilateral triangles.

KeyWords: Algorithms, Tiling, Symmetry, Reverse Search, Polyiamonds

1 はじめに

タイリングは、基本図形に平行移動、回転、すべり鏡映などの単純な操作を繰り返し適用することで、隙間なく重なりなく平面を敷き詰めることをいう。タイリングは、アルハンブラ宮殿の壁面装飾やエッシャーの絵画、正倉院裂 [8] 等の織物のデザイン、その他にも着物やカーテン、壁紙などの工業デザインなど多岐に渡って利用されている。また、畳敷詰問題 [9, 10] では、敷き詰め対象が平面ではないが、与えられた矩形内に 1:2 の矩形の敷き詰め方を数え上げることが求められる。近年では、CG の分野において、3 次元空間内で床や

壁面などに様々な模様を描くためにタイリングを利用する手法 [4, 6, 7] が提案されている。このためにはタイリング可能な基本図形を自動的に数多く生成することが求められる。

タイリングの種類は、結晶物理学における結晶の原子配列の対称性および数学における対称性の議論により、2次元空間群は 17 種類、3次元空間群は 230 種類に分類されることが知られている。本稿では、その中でも p6 タイリングに着目し、p6 タイリング可能な polyiamond の生成を行う。ここで、polyiamond とは単位正三角形を辺同士が接続するように組み合わせてできる図形である。p6 タイリングでは、origin と terminus と呼ばれる 2つの回転中心のもとで、origin について 60 度回転、terminus について 120 度回転を繰り返すことで基本図形を平面に敷き詰める。

回転によるタイリングとして、p6 タイリング可能な polyiamond をはじめ、90 度回転による p4

* 埼玉大学 理工学研究科

Graduate School of Science and Engineering,
Saitama University, 255 Shimo-Okubo, Sakura-
ku, Saitama, Saitama 338-8570, Japan

(原稿受付日：平成 22 年 5 月 31 日)

タイリング可能な polyomino (単位正方形による図形) や 120 度回転による p3 タイリング可能な polyiamond, polyhex (単位正六角形による図形) の生成法が福田らにより提案されている [2, 3]. しかし, 試行錯誤による生成であるため計算時間が長くなり, また, 過去に生成した図形との同一性判定が必要なため, 計算時間と計算領域が大きくなる. これに対し, p4 タイリングについて, 逆探索 [1] による生成法が提案されている. 逆探索は木構造を利用した列挙手法であり, ルールに従って次に生成する図形を決定できるため, 計算時間の効率化が図れる. また, 過去に生成した図形との同一性判定が不要であるため, 計算時間と計算領域の削減が可能となる.

本稿では, 逆探索による手法を拡張し, p6 タイリング可能な polyiamond の生成を行う. 本稿では列挙ではなく生成問題を扱うため, 条件に合うものすべてではなくその部分集合を生成する. 提案手法では, 生成対象の図形の間親子関係を定義することで, 家系木と呼ばれる木構造を導入する. 具体的には, 任意の図形の親を一意に求めるルールを定め, その祖先へとたどることで根の図形と呼ばれる唯一の図形に到達できるように設計する. 子から親へのルールを逆向きにした親から子へのルールを与えることで, 根の図形から出発し, 家系木を順にたどることが可能となる. なお, p6 タイリングは p4 タイリングとは異なる性質を持つため, 生成ルールを拡張し, p6 タイリング特有の新たなルールを加える必要がある. さらに, 提案手法を実装して計算機実験を行う. 9 個の単位正三角形による 29 種類の図形までを生成した従来法に対し, 提案手法では 21 個の単位正三角形による 27,436 種類の図形を生成することに成功している.

2 準備

2.1 タイリング

タイリングとは, 与えられた基本図形に対し, 平行移動, 回転, すべり鏡映などの操作を加えるこ

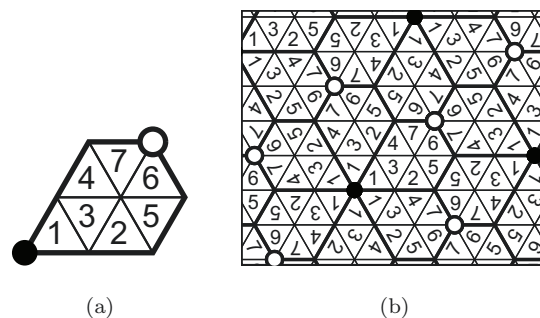


Figure 1: 7-iamond as a fundamental domain and its corresponding isohedral p6 tiling.

とにより, 隙間なく, また重なりもなく, 平面に敷き詰めることをいう. それらの操作の組み合わせにより 17 種類のタイリング方法があり, 本稿ではその中でも p6 タイリングについて取り扱う.

p6 タイリングでは, 基本図形は図 1 に示すように 2 つの回転中心 origin (黒丸), terminus (白丸) を持ち, origin について 60 度回転, terminus について 120 度回転を繰り返す. 例えば図 1(a) の基本図形は, この回転操作により図 1(b) のように平面に敷き詰められる.

ここで, p6 タイリング可能となる図形は以下の条件を満たす必要がある. (1) origin, terminus を中心に 60 度, 120 度回転を繰り返すことにより隙間なく, 重なりなく, 平面に敷き詰められる. (2) origin, terminus を含む単位正三角形を持つ. (3) 3 近傍で連結な図形である.

2.2 Polyiamond

Polyiamond とは単位正三角形を辺同士が接続するように組み合わせることができる図形であり, 単位正三角形の個数 n により n -iamond とも呼ばれる. 本稿で扱う p6 タイリング可能な polyiamond 生成問題では, 三角格子上に origin $(0, 0)$, terminus (x, y) が与えられ (単位ベクトル u, v は 60 度の角度をなす), この 2 つを回転中心としてもつ p6 タイリング可能な polyiamond を生成する. ここで, x, y は整数であり, 一般性を失うことなく $y \geq x \geq 0$ を仮定することができる. また, 与えられた x, y に対し, polyiamond は $n = x^2 + y^2 + xy$ 個の単位正三角形で構成される [2, 3].

2.3 同値類

p6 タイリングでは、60 度回転や 120 度回転のみが許される操作であり、単位正三角形の移動位置は固定的である。したがって、三角格子上的単位正三角形の位置に同値関係が定義でき、これにより規定される同値類に番号を付けて互いを区別できる。p6 タイリング可能な polyiamond は、同じ同値類に属する単位正三角形を同時に複数個持つことはなく、 n 種類の同値類からそれぞれ 1 つずつの単位正三角形を持つ [2, 3].

3 Polyiamond の生成

本章では、逆探索により p6 タイリング可能となる polyiamond を生成する方法について述べる。まず、生成対象の図形間の親子関係、すなわち子から親の図形を一意に求めるルールを定めることで、家系木を設計する。次に、子から親へのルールを逆向きにして親から子の図形を求めるルールを定める。根の図形から出発し、このルールを繰り返し適用することで家系木上の各図形を順に生成することができる。

3.1 子から親へのルール

Origin $(0,0)$, terminus (x,y) が与えられた時、対応するそれぞれの n -iamond に対し家系木を設計する。まずは、家系木の根となる図形を決定する。根の図形は、次の 3 つの領域からなる図形であり、任意の n -iamond に対してそれぞれ一意に定まる。

1. Origin から u の方向に一辺が x の正三角形。
2. $(x,0)$ から v の方向に一辺が y の正三角形。
3. 領域 1, 2 で、origin を中心に 60 度回転, terminus を中心に 120 度回転を繰り返すと、平行四辺形の隙間ができる。領域 3 は、この平行四辺形の半分の領域をとる。

例えば、図 2 は 7-iamond の根の図形を表している。3 の領域は x の偶奇により次のように場合分けを行う。 x が偶数の場合は、図 3(a) のように平行四辺形を高さが $x/2$ となるように 2 分割した

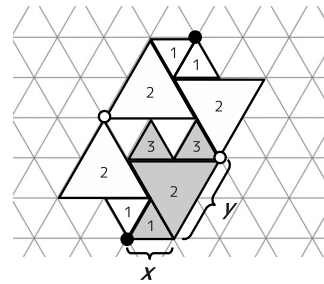


Figure 2: The root polyiamond for $n = 7$.

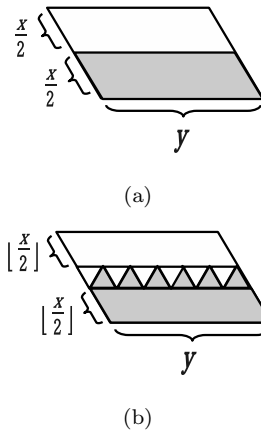


Figure 3: Region (3) of the root polyiamond.

領域をとり、 x が奇数の場合は、図 3(b) のように高さが $\lfloor x/2 \rfloor$ となる平行四辺形と、それに 3 近傍で隣接する単位正三角形の領域をとる。

以上の 3 つからなる領域を根の図形とすることで、任意の n -iamond に対して、隙間も重なりもなく、タイリング可能な図形を与えることができる。領域内に含まれる単位正三角形の個数は、領域 1 では x^2 個、領域 2 では y^2 個、領域 3 では、具体的には terminus から $-u$ の方向に y 個の単位正三角形を x 列とっている。以上 3 つの領域内に含まれる単位正三角形の個数の和は $n = x^2 + y^2 + xy$ を満たすことが確認できる。

次に、家系木を作るために、子から親の図形への基本的なルールを次のように定める。

(子から親への基本ルール) 根の図形に近づけるように単位正三角形を 1 つ移動する。

ここで、単位正三角形の根の図形からの「近さ」について定義する。「近さ」はレベルで表すとし、レベルは根の図形からの 3 近傍での距離とする。つまり、基本ルールの根の図形に近づけるとは、レ

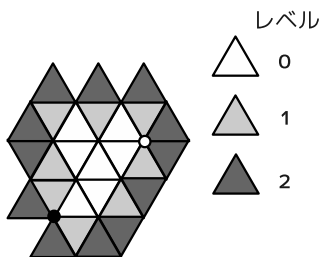


Figure 4: Levels of the equilateral triangles.

レベルを下げるように単位正三角形を移動することと等価となる。例えば、図4は7-diamondのレベルを示している。根の図形内部のレベルは0であり、根の図形に隣接する単位正三角形はレベル1、そこからさらに隣接する単位正三角形のレベルは2となる。

また、単位正三角形の移動には移動後の図形もまたタイリング可能である必要があるため、移動の前後で同じ同値類に属する位置に移動させる。

逆探索では、木構造を定義するために任意の図形に対してその親が一意に決まるルールである必要がある。そのため、2つ以上の移動候補がある場合は、どれか1つの移動を正当な子から親への移動と決定しなければならない。そこで、複数の移動候補がある場合、次のように追加ルールを設ける。

(追加ルール1) 複数個の単位正三角形を根に近づけられる場合、同値類の番号の一番小さいものを移動させる。

図5のような場合、同値類の3または6を移動し、根の図形へと近づけることができる。この場合は同値類の番号の小さい3の単位正三角形を移動させたものが子から親への正当な移動とする。

(追加ルール2) 単位正三角形の移動候補位置としてレベルの小さくなる位置が複数ある場合、レベルの一番小さくなる候補位置へ移動させる。

図6のような場合、レベル3の単位正三角形はレベル2または0の位置へと移動することができる。この場合はよりレベルの小さいレベル0の位置へ移動させたものが子から親への正当な移動とする。

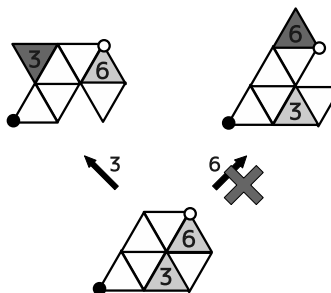


Figure 5: Example of additional rule 1.

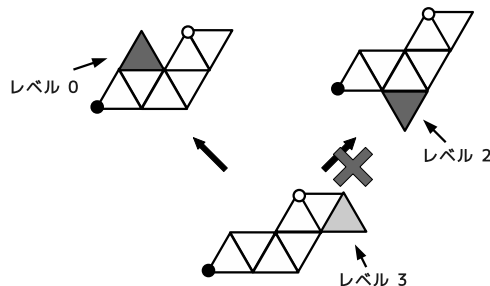


Figure 6: Example of additional rule 2.

p6 タイリングでは連結性を保ちつつ移動するために、次のようなルールを追加する。

(追加ルール3) p6 タイリングの場合、origin を含む単位正三角形は、それに隣接する単位正三角形と同時に動かす。

移動前の図形は origin を含む単位正三角形 A (図7(a) 着色部) をもつ。その同値類の番号を a とおく。A を origin を中心に 60 度回転させると、図7(a) のようになり、A の3近傍の内2つは同じ同値類 a に属する。図形が連結であるためには、A の残り1つの近傍 B が移動前の図形に含まれる必要がある。(図7(b) の着色部が A, B)。また B も同様に 60 度回転させると図7(b) のようになり、A, B のいずれか一方のみを動かすと連結性が

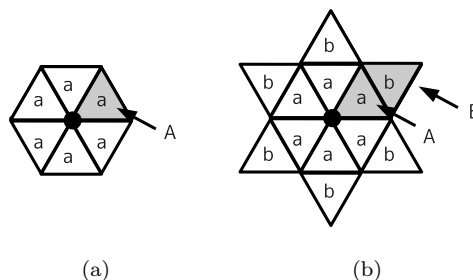


Figure 7: Equilateral triangles around the origin.

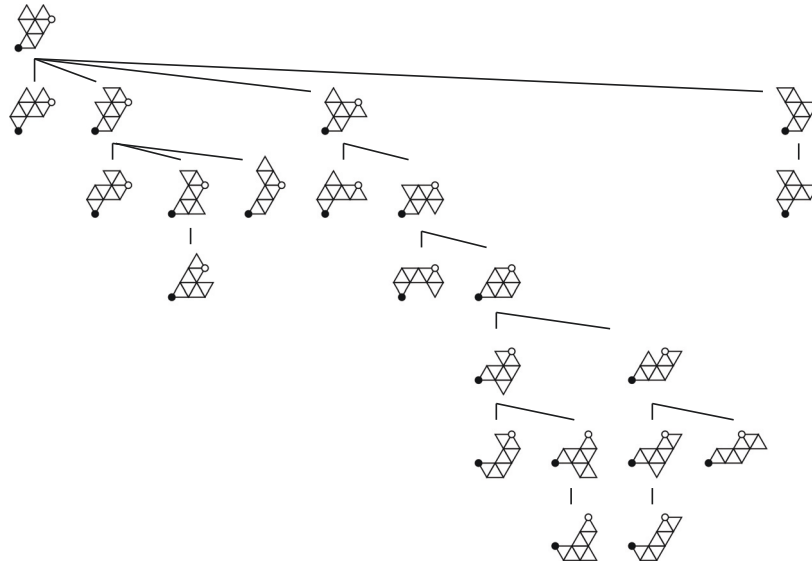


Figure 9: The family tree of 7-iamonds.

保たれないことが分かる. したがって, p6 タイリングでは, 図 8 のように, A と B の 2 つの単位正三角形を同時に動かす移動を子から親への正当な移動とする.

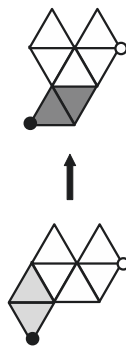


Figure 8: Example of additional rule 3.

以上のルールを適用すると, 図 9 のように家系木ができる. これは 7-iamond の家系木である. 各ノードは p6 タイリング可能な polyiamond であり, 辺は親子関係を表す.

3.2 親から子へのルール

次に, 根の図形から子を求めていくことで対象となる図形をすべて探索するため, 親から子へのルールを設計する. これは, 子から親へのルールの逆となるようなルールにすればよく, 次のようになる.

(親から子への基本ルール) 根の図形から遠ざけるように単位正三角形を 1 つ移動させる.

子から親のルールと同様に, p6 タイリングを生成する場合は origin を含む単位正三角形とそれに隣接する単位正三角形は同時に動かす.

親から子へのルールの, 具体的なアルゴリズムは次のようする. 1. 根の図形から遠ざかるように移動可能な単位正三角形をすべて移動候補とする. 2. 移動後の図形の親を確認し, 移動前の図形と一致すれば, その移動は正当な子への移動であり, その図形を生成する.

以上のルールを根の図形に適用し, 生成した図形すべてに対して上記の手順を繰り返すことにより p6 タイリング可能な polyiamond を生成する. 根の図形は p6 タイリング可能であるため, n 種類の同値類からそれぞれ 1 つずつの単位正三角形を持つ. また, 単位正三角形の移動では, 移動の前後で同じ同値類に属する位置へと移動させているため, 生成する polyiamond はすべて p6 タイリング可能な図形である.

3.3 単位正三角形の分類

図 10(a) は, 図 10(b) を 60 度回転してできる図形である. これら 2 つの図形は, それぞれの回転中心で回転させると平面に敷き詰められた図形は同じものとなる. また, 図 10(c) は, 図 10(a) の

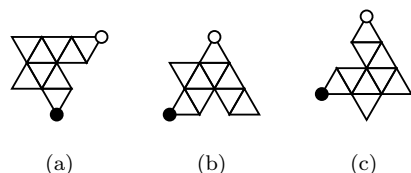


Figure 10: Polyiamonds of the same isohedral tiling with rotation/mirroring.

origin, terminus を結ぶ線の鏡映になっている。これら 2 つの図形は、それぞれの回転中心で回転させると平面に敷き詰められた図形が互いに鏡映の関係になる。

このように、平面に敷き詰められた図形が同じか、鏡映の関係にあるかで polyiamond を同一視するか、区別する分類法があり、(1) 回転や鏡映で同じ図形となるものもすべて区別する fixed, (2) 回転で同じ図形となるものを同一視する one sided, (3) 回転や鏡映で同じ図形となるものは同一視する free の 3 つに分類される。

本稿で生成する polyiamond は、まず初めに回転や鏡映して同一となるものを区別する fixed な分類で polyiamond を生成する。さらにそこから、回転中心で回転させると同じ図形になる冗長な図形を削除し、one sided な polyiamond を生成する。同様に、回転や鏡映により同一となる図形を削除し、free な分類の polyiamond を生成する。

Free や one sided な分類の図形を求めるには、fixed な分類で生成された図形とそれを回転、鏡映した図形とを比較し、自分自身が単位正三角形の位置の辞書順で最大になる場合のみ出力すればよい。この比較にも過去に生成した図形との比較は不要である。

4 実験結果

3章で提案したアルゴリズムを実装し、生成した結果を表 1 に示す。計算時間については主たる生成対象である fixed な polyiamond の生成時間を計測している。実行環境は、CPU: Intel(R) Core(TM)2 Duo 2.66GHz, メモリ: 2GB, Ubuntu Linux である。

Table 1: Generated polyiamonds and the computation time.

n	terminus	fixed	one sided	free	time (sec)
1	(0, 1)	2	1	1	0.01
3	(1, 1)	2	2	1	0.01
4	(0, 2)	6	5	3	0.01
7	(1, 2)	22	20	20	0.01
9	(0, 3)	62	57	29	0.01
12	(2, 2)	412	390	195	0.07
13	(1, 3)	530	504	504	0.10
16	(0, 4)	3,063	2,964	1,453	0.71
19	(2, 3)	10,722	10,357	10,357	2.56
21	(1, 4)	28,276	27,436	27,436	8.37

既存の polyiamond の生成法 [3] では、単位正三角形の個数 $n = 9$ までの free な polyiamond について生成を行っている。本稿で提案したアルゴリズムでは表 1 に示すように、 $n = 21$ までの fixed, one sided, free な polyiamond について生成を行った。提案アルゴリズムでは、既存の手法よりも 3 桁多い個数の polyiamond を生成することに成功した。また、 $n = 9$ までのすべてで既存の手法と同じ図形が生成できていることを確認した。計算時間は $n = 16$ までは 1 秒以内であり、その後も比較的小さい。

既存の手法では、過去に生成した図形すべてを計算領域内に保持する必要がある。そのため、単位正三角形の位置を一つあたり仮に 2 バイトで表せるとして、28,276 個の 21-iamond の保持に、単純には最低でも 1,187 k バイトの領域が必要となる。提案手法では、これを 368 k バイトの領域で生成している。本手法が必要な計算領域は n の増加に対して緩やかであるため、大きな n に対して既存手法との計算領域の差はますます広がる。これはルールに従って次の図形を生成でき、同一性判定が不要という逆探索の特長を表している。本稿での逆探索を用いた polyiamond の生成法は計

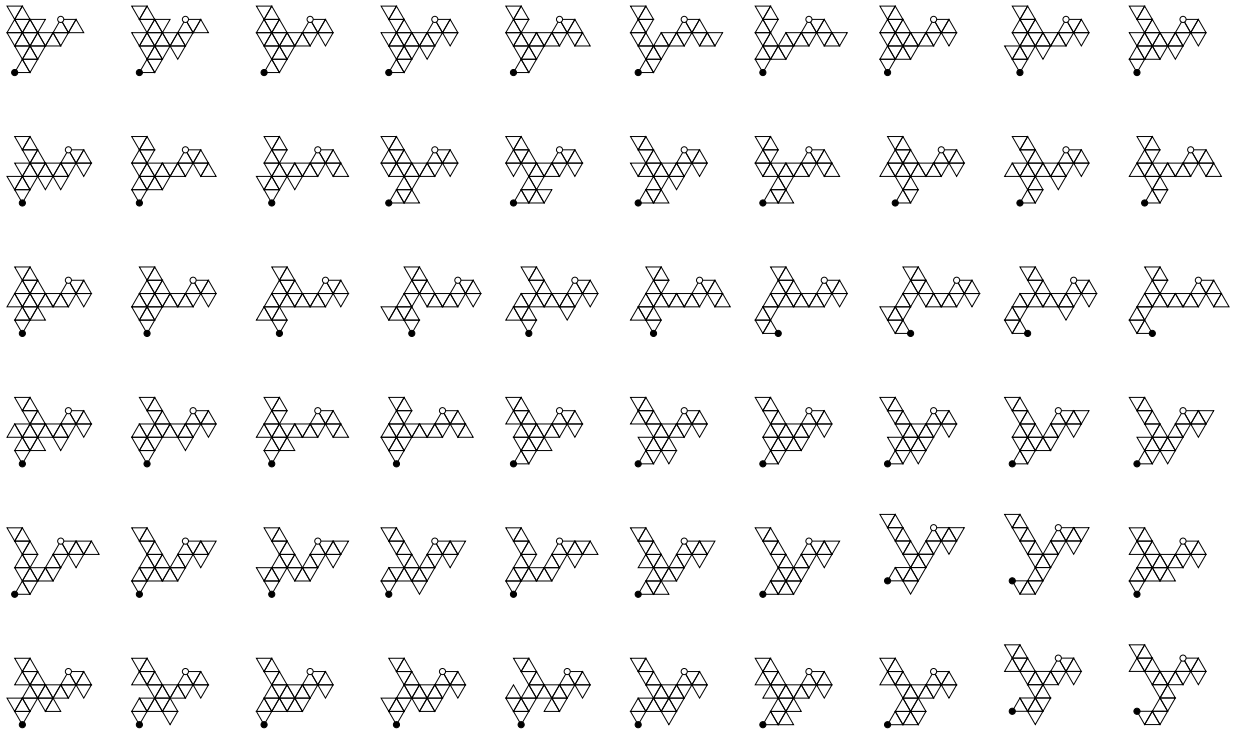


Figure 11: Partial list of 21-iamonds for p6 tiling.

算時間の効率化と計算領域の削減に成功したと言える。実行結果の例として、生成した 21-iamond の一部を図 11 に示す。これらの図形はすべて p6 タイリング可能である。

5 まとめ

本稿では、逆探索により $n = 21$ までの p6 タイリング可能な polyiamond を生成した。既存の生成法では、多大な計算量と、計算領域を要したが、逆探索による生成法を提案したことにより計算時間の効率化と計算領域の削減に成功した。17 種類のタイリング方法すべてに生成法を与えることは、今後の課題である。

参考文献

- [1] D. Avis and K. Fukuda, Reverse Search for Enumeration, *Discrete Appl. Math.*, 6, pp.21–46, 1996.
- [2] H. Fukuda, N. Mutoh, G. Nakamura, and D. Schattschneider, A Method to Generate Polyominoes, Polyiamonds and Polyhexes for Isohedral Tilings with Rotational Symmetry, *Graphs and Combinatorics*, 23, pp.259–267, 2007.
- [3] H. Fukuda, N. Mutoh, G. Nakamura, and D. Schattschneider, Enumeration of Polyominoes, Polyiamonds and Polyhexes for Isohedral Tilings with Rotational Symmetry, *Lecture Notes in Computer Science* 4535, pp.68–78, 2008.
- [4] H. Gürçay, Introducing Iso-Tailer 3D: A 3D Tiling Visualizer, *Hacettepe Journal of Mathematics and Statistics*, 37, pp.107–114, 2008.
- [5] T. Horiyama, M. Samejima, Enumeration of Polyominoes for p4 Tiling, *In Proc. of the 21st Canadian Conference on Computational Geometry*, pp.29–32, 2009.
- [6] C. S. Kaplan, D. H. Salesin, Escherization, *In Proc. of SIGGRAPH*, pp.499–510, 2000.
- [7] C. S. Kaplan, D. H. Salesin, Dihedral Escherization, *In Proc. of Graphics Interface, Canadian Human-Computer Communications Society*, pp.255–262, 2004.
- [8] 甲藤麻衣子, 顔紅梅, 近藤誠造, 三橋俊雄, 河西立雄, 正倉院裂繰り返し紋様の研究 — 数学的手法 (群論) を用いたパターン形成の分析と提案 —, 日本デザイン学会, デザイン学研究. 研究発表大会概要集, 48, pp.396–397, 2001.
- [9] D. E. Knuth, *The Art of Computer Programming*, vol. 4, fascicle 1: Bitwise Tricks & Techniques; Binary Decision Diagrams, Addison-Wesley, 2009.
- [10] 吉田 光由, 塵劫記, 1641 (初版 1627).