Plane-filling curves on all uniform grids

Jörg Arndt, <arndt@jjj.de>
Technische Hochschule Nürnberg

May 31, 2017

Abstract

We describe a search for plane-filling curves traversing all edges of a grid once. The curves are given by Lindenmayer systems with only one non-constant letter. All such curves for small orders on three grids have been found. For all uniform grids we show how curves traversing all points once can be obtained from the curves found. Curves traversing all edges once are described for the four uniform grids where they exist.

Contents

1 Introduction **2**
1.1 Self-avoiding edge-covering curves on a grid . . . 2
1.2 Description via simple Lindenmayer-systems . . . 3
1.3 Some known plane-filling curves . . . 6
1.4 Other descriptions using L-systems . . . 8

2 The search **10**
2.1 Conditions for a curve to be self-avoiding and edge-covering . . . 10
2.2 The shape of a curve . . . 11
2.3 Structure of the program for searching . . . 12
2.4 Format of the files specifying the curves . . . 12
2.5 Numbers of shapes found . . . 14

3 Properties of curves and tiles **14**
3.1 Self-similarity, symmetries, and tiling property . . . 14
3.2 Certain arrangements of curves . . . 19
3.3 Tiles and complex numeration systems . . . 21
3.4 Curves and tiles on the tri-hexagonal grid . . . 26

4 Plane-filling curves on all uniform grids **29**
4.1 Conversions to point-covering curves . . . 30
4.2 Conversions to edge-covering curves . . . 42

5 Multiplying and dividing curves **47**
5.1 Products of curves . . . 47
5.2 Divisions of a curve . . . 51

6 Eye of the beholder **55**
6.1 Triangular grid . . . 56
6.2 Square grid . . . 60
6.3 Tri-Hexagonal grid . . . 66

1 Introduction

1.1 Self-avoiding edge-covering curves on a grid

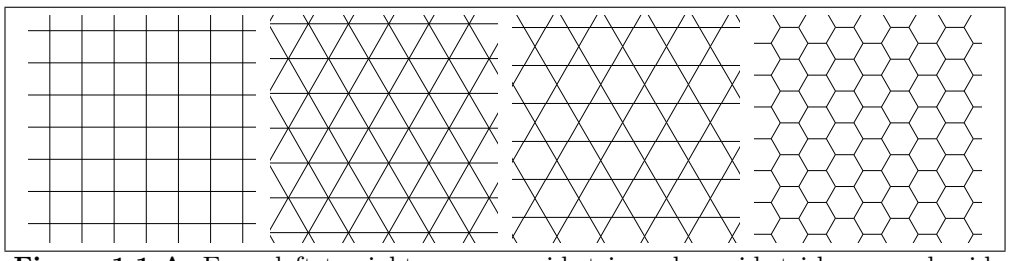

Figure 1.1-A: From left to right: square grid, triangular grid, tri-hexagonal grid, and hexagonal (honeycomb) grid.

We are interested in self-avoiding, plane-filling curves on certain grids. A curve is *self-avoiding* if it neither crosses itself nor has an edge that is traversed twice. It is *edge-covering* if it traverses all edges of some grid.

self-avoiding
edge-covering

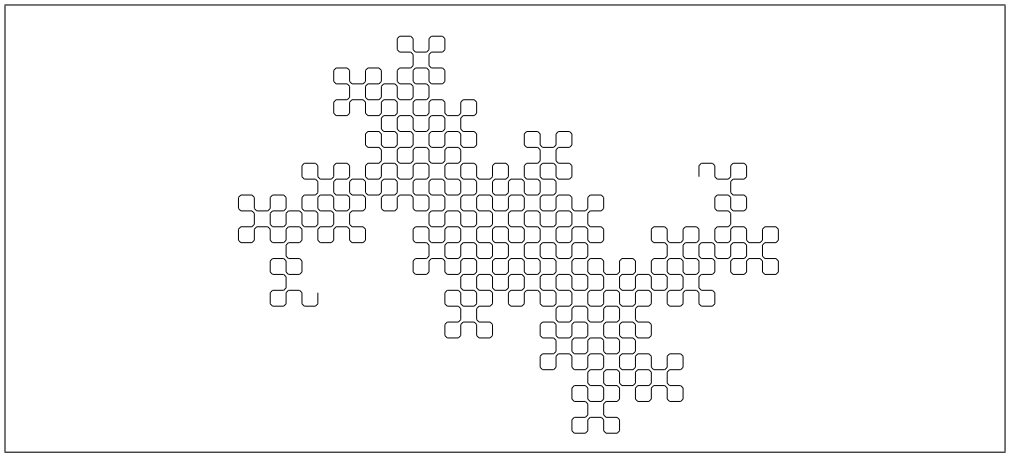

Figure 1.1-B: The R5-dragon, a curve on the square grid (`R5-1`).

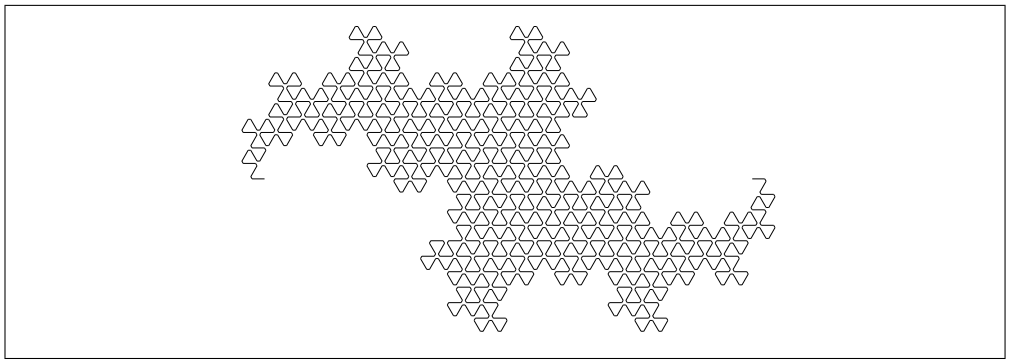

Figure 1.1-C: The terdragon, a curve on the triangular grid (`R3-1`).

The curves traverse three grids: the square grid (Figure 1.1-B), the triangular grid (Figure 1.1-C), and the tri-hexagonal grid (Figure 1.1-D). The ID of each curve such

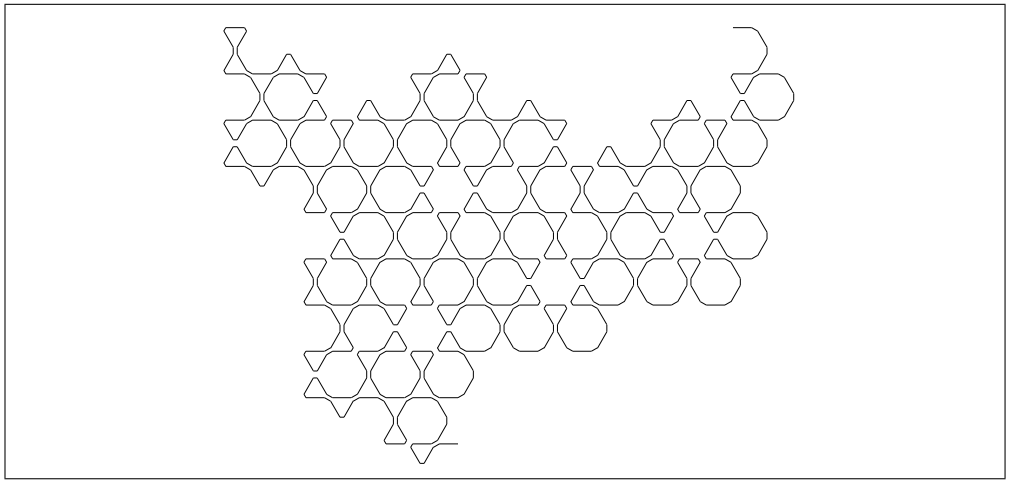

Figure 1.1-D: Ventrella's curve, a curve on the tri-hexagonal grid (`R7-1`).

as `R7-1` is explained in section 2.4.1. The turns are slightly rounded in these figures to make the curves visually apparent.

For edge-covering curves to exist, the grid must have an even number of incident edges at each point. Otherwise a dead end is produced after the point is traversed sufficiently often. This rules out the hexagonal (honeycomb) grid as every point has incidence 3.

1.2 Description via simple Lindenmayer-systems

A *Lindenmayer-system* (L-system, see [22, Section 1.2, pp. 3ff]) is a triple (Ω, A, P) where Ω is an alphabet, A a word over Ω (called the *axiom*), and P a set of maps from letters $\in \Omega$ to words over Ω that contains one map for each letter.

Lindenmayer-system
axiom

The word that a letter is mapped to is called the *production* of the letter. If the map for a letter is the identity, we call the letter a *constant* of the L-system.

production
constant

We specify curves by L-systems interpreted as a sequence of (unit-length) edges and turns. The curves can be rendered via turtle-graphics, see [22, Section 1.3, pp. 6ff]. The initial position and direction are arbitrary. Letters are interpreted as "draw a unit stroke in the current direction", `+` and `-` as turns by $\pm$ a fixed angle ϕ (set to either 60°, 90°, or 120°). We will also use the constant letter `0` for turns by 0° (non-turns).

As an example we take the L-system with alphabet $\Omega = \{$`L, R, +, -`$\}$, axiom $A =$ `L`, and the maps $P = \{$`L` $\mapsto$ `L+R`, `R` $\mapsto$ `L-R`, `+` $\mapsto$ `+`, `-` $\mapsto$ `-`$\}$. There are two constants, `+` and `-`, in this L-system.

Starting with the axiom and repeatedly applying the maps $\in P$, we obtain the words shown in Figure 1.2-A. We call the word obtained after the maps were applied n times ($n \geq 0$) the nth *iterate* of the L-system.

iterate

```
0: L
1: L+R
2: L+R+L-R
3: L+R+L-R+L+R-L-R
4: L+R+L-R+L+R-L-R+L+R+L-R-L+R-L-R
5: L+R+L-R+L+R-L-R+L+R+L-R-L+R-L-R+L+R+L-R+L+R-L-R-L+R+L-R-L+R-L-R
```

Figure 1.2-A: The nth iterates ($n \in \{0, 1, 2, 3, 4, 5\}$) of a Lindenmayer-system.

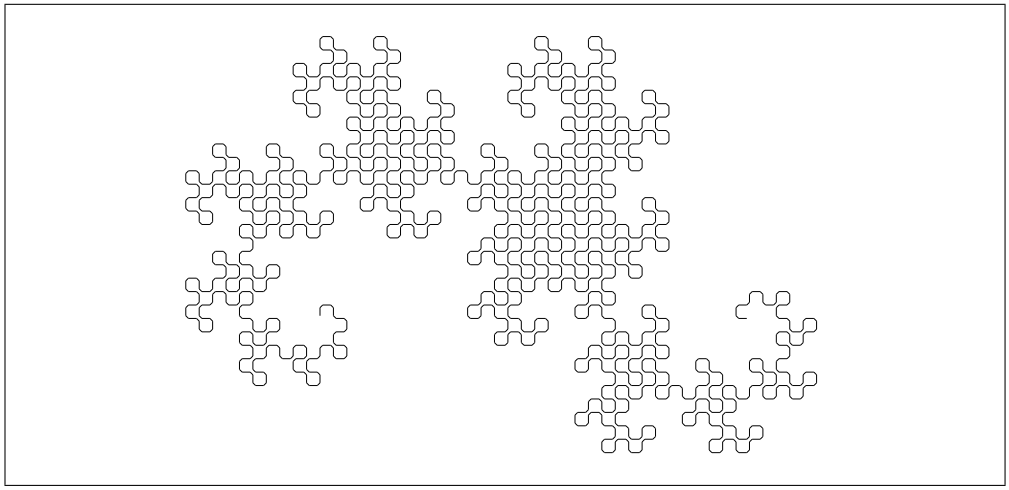

Figure 1.2-B: The Heighway-Harter dragon.

The curve shown in Figure 1.2-B found in the 1960s by John Heighway (sometimes called *Heighway-Harter dragon*) corresponds to the L-system just described: we interpret `+` and `-` as turns by ±90° and both `L` and `R` as edges of unit length. The curve shown corresponds to the 10th iterate of the L-system. A description in terms of the paper-folding sequence is given in [2, Chapter 5, pp. 152ff]. *Heighway-Harter dragon*

The so-called *terdragon* shown in Figure 1.1-C corresponds to the L-system with axiom `F` and maps `F` ↦ `F+F-F`, `+` ↦ `+`, and `-` ↦ `-` where the turns are by 120° and `F` is an edge. Both curves were described by Chandler Davis and Donald Knuth [6] in 1970 (an extended version of the paper is reprinted in [7]). *terdragon*

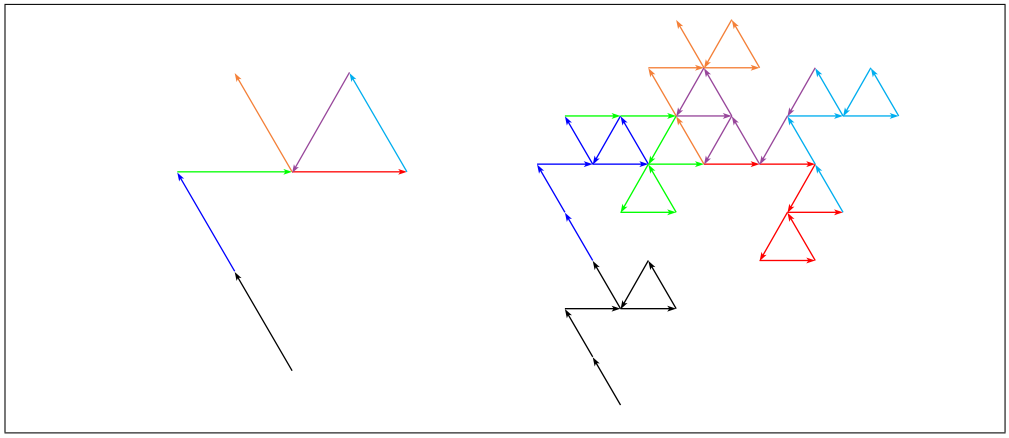

Figure 1.2-C: First iterate (motif) and second iterate of a curve of order 7 (`R7-1`).

We call an L-system *simple* if it has just one non-constant letter. The terdragon corresponds to a simple L-system but the Heighway-Harter dragon does not. Only curves with simple L-systems are considered for the search to keep the search space manageable. *simple*

For simple L-systems we always use `F` for the only non-constant letter. The axiom (`F`) and the maps for the constant letters (`+` ↦ `+`, `-` ↦ `-`) will be omitted.

The *order* R of a curve is the number of `F`s in the production of `F` by the (simple) L-system. For example, the terdragon (Figure 1.1-C) has order $R = 3$ (`F` ↦ `F+F-F`) *order*

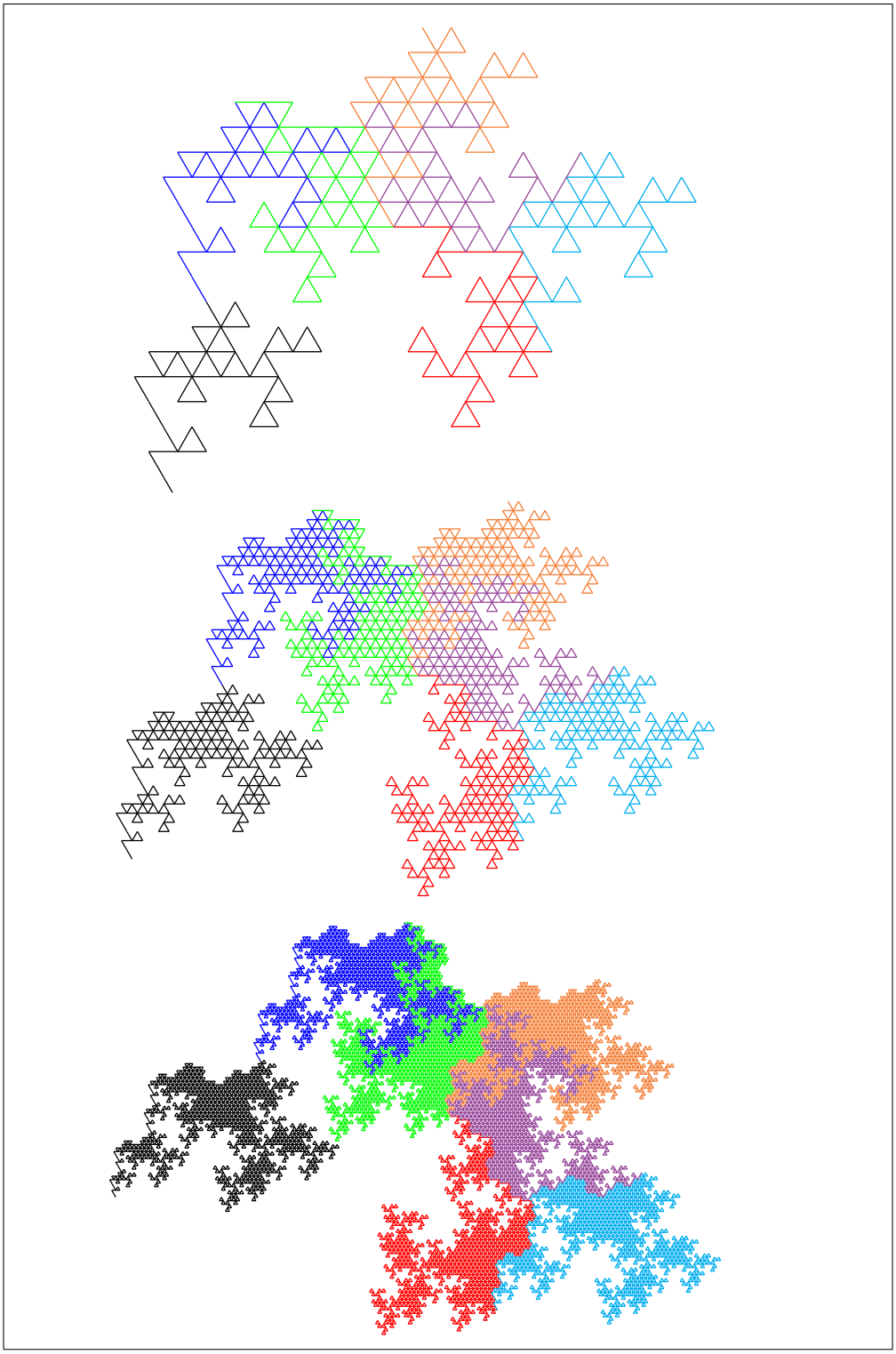

Figure 1.2-D: Third, fourth, and fifth iterate of the curve `R7-1`.

and the curve shown in Figure 1.1-B has order $R = 5$ (`F` $\mapsto$ `F+F+F-F-F`).

We call the curve corresponding to the nth iterate of an L-system the nth *iterate of the curve*. We call the first iterate the *motif* of the curve. Iterate 0 corresponds to a single edge and iterate n is obtained from iterate $n-1$ by replacing every edge by the motive. Figures 1.2-C and 1.2-D show the iterates 1 through 5 of a curve of order 7 on the triangular grid with L-system `F` $\mapsto$ `F0F+F0F-F-F+F`. Here the constant letter `0` stands for "no turn". The motif and fourth iterate of the curve of order 13 on the square grid with L-system `F` $\mapsto$ `F+F-F-F+F+F+F-F+F-F-F-F+F` are shown in Figure 1.2-E.

iterate of the curve
motif

Note that Davis and Knuth use "order" for what we call "iterate" [7, Figure 21, p. 600].

It is our goal to find *all* self-avoiding edge-covering curves with simple L-systems of small orders on the three grids mentioned.

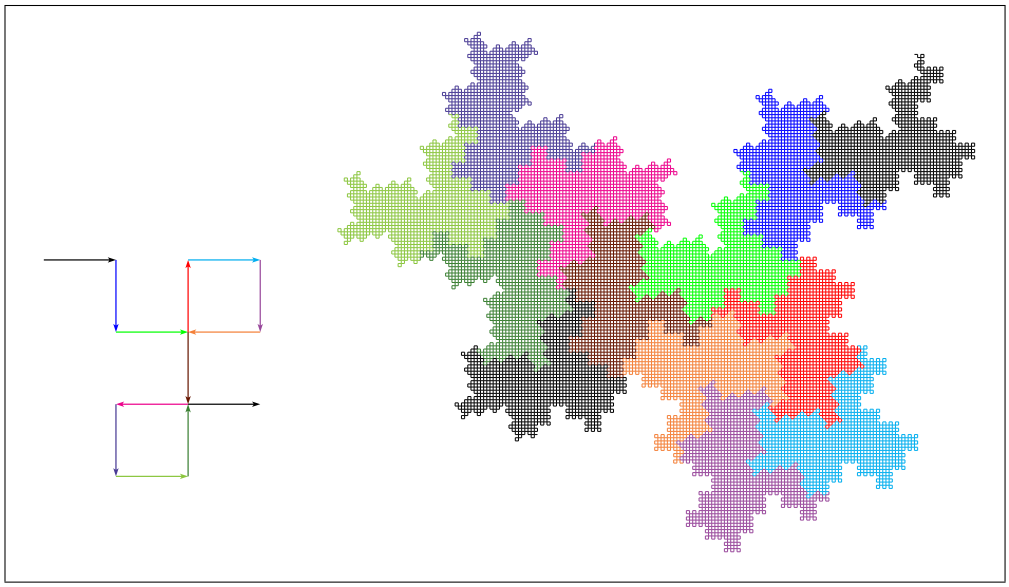

Figure 1.2-E: First iterate (motif) and fourth iterate of a curve of order 13 (`R13-3`).

1.3 Some known plane-filling curves

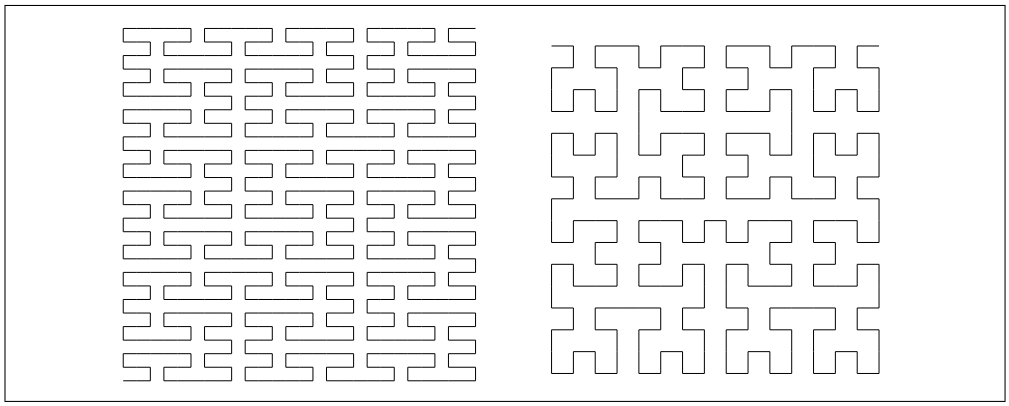

Figure 1.3-A: The Peano curve (left) and the Hilbert curve (right).

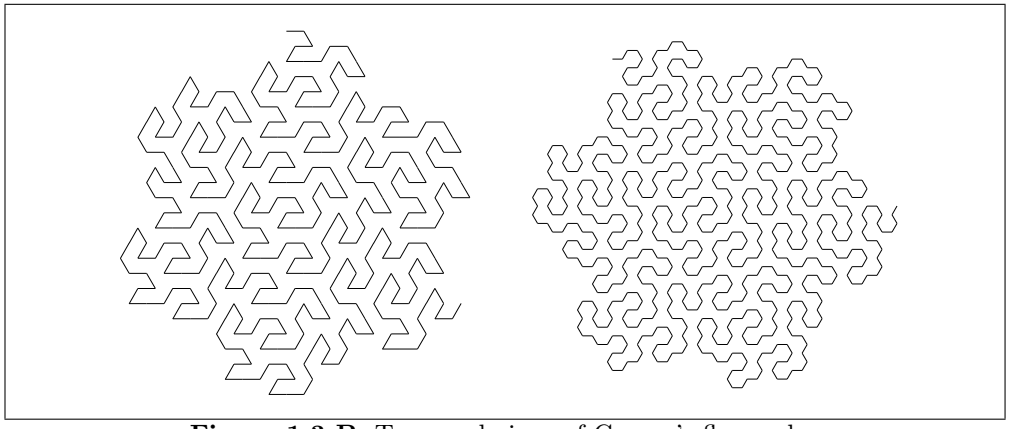

Figure 1.3-B: Two renderings of Gosper's flowsnake.

The earliest example of a plane-filling curve was described by Giuseppe Peano in 1890 [20] (also see [23, Chapter 3, pp. 31ff] and [5] for the construction of n-dimensional Peano curves via the Gray code for base 3). In 1891 David Hilbert described the curve now named after him [15]. Both curves are shown in Figure 1.3-A. *Peano curve* *Hilbert curve*

In the 1970s Bill Gosper discovered his *flowsnake* curve shown in Figure 1.3-B [11]. *flowsnake*

Gosper's curve 1.3-B can be given by the axiom `L` and non-constant maps `L` $\mapsto$ `L+R++R-L--LL-R+` and `R` $\mapsto$ `-L+RR++R+L--L-R` where the turns are by ± 60 degrees. Both `L` and `R` correspond to edges. The rendering on the right is obtained via post-processing replacing all `L` by `L-L+` and all `R` by `-R+R`.

More curves resembling Gosper's flowsnake were given by Jin Akiyama, Hiroshi Fukuda, Hiro Ito, and Gisaku Nakamura in 2007 [1].

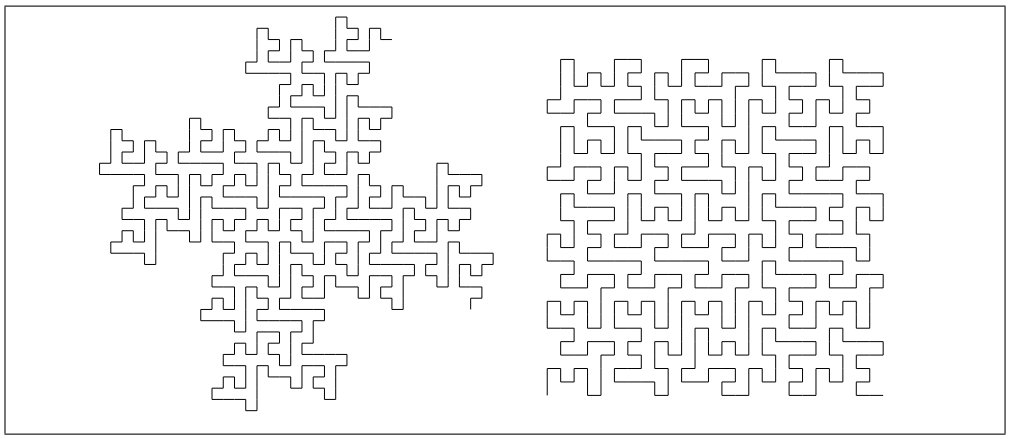

Figure 1.3-C: A curve on the square grid given by Mandelbrot (left) and the E-Curve given by McKenna (right).

Benoit Mandelbrot [18, Plate 49, p. 49] shows a curve with the L-system with axiom `L` and (non-constant) maps `L` $\mapsto$ `-RL+L+R-L`, `R` $\mapsto$ `R+L-R-RL+`, see Figure 1.3-C (left).

Douglas M. McKenna gave what he calls the *E-Curve* in 1994 [19, p. 60], shown in Figure 1.3-C (right). It can be generated by the L-system with maps *E-Curve*

`L` $\mapsto$ `-RR+L+L-R-RL-R+LL+R+L-RLL+R+LR-R-L+L+R-R-LL` and
`R` $\mapsto$ `RR+L+L-R-R+L+LR-L-RRL+R-L-RR-L+RL+L+R-R-LL+`.

None of these curves corresponds to a simple L-system or is edge-covering. They do traverse every point of the grid once. We will call such curves *point-covering*. *point-covering*

The already mentioned curve shown in Figure 1.1-B first appeared in [8] and [9] and later in [3, Section 1.31.5, p. 95ff] (therein called *R5-dragon*) and [25, p. 84]. *R5-dragon*

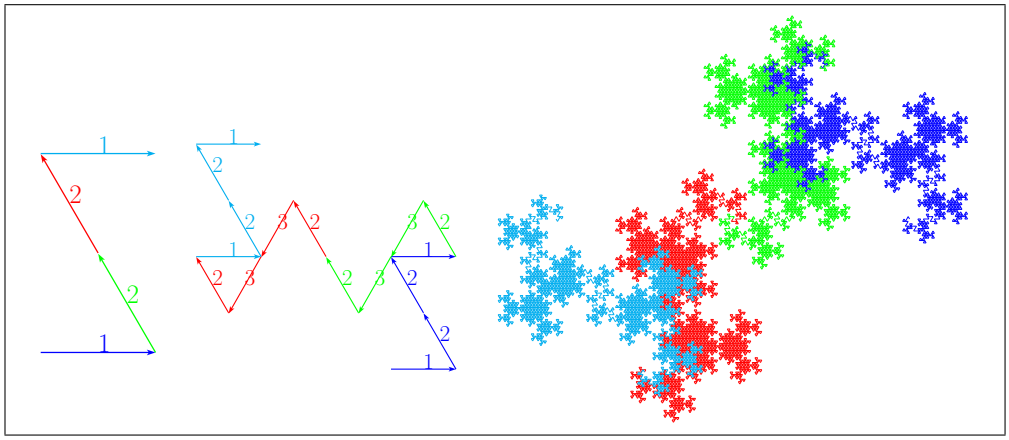

Figure 1.3-D: First, second, and seventh iterate of a curve of order 4 on the triangular grid (`R4-1`). The numbers (left and middle) correspond to the directions of the edges.

A lesser known curve on the triangular grid with L-system `F` ↦ `F+FOF-F`, shown in Figure 1.3-D, is called the *crab* by Gilbert Helmberg [13, Section 1.1.5.2, p. 24ff]. Note that Helmberg calls "generator" what we call "motif". *crab*

A great number of plane-filling curves are given in Jeffrey Ventrella's 2012 book "Brain Filling Curves" [25]. Some of the curves correspond to simple L-systems. He especially gives the first curve on the tri-hexagonal grid (page 105 in [25], see Figure 1.1-D) and an asymmetric curve on the triangular grid (top of page 107 in [25], see Figures 1.2-C and 1.2-D). The latter curve has already been found by Bill Gosper in the 1970s, but there seems to be no publication prior to Ventrella describing it.

We will only find edge-covering curves in our search. Methods to obtain point-covering curves from the ones found are given in section 4.1 on page 30.

1.4 Other descriptions using L-systems

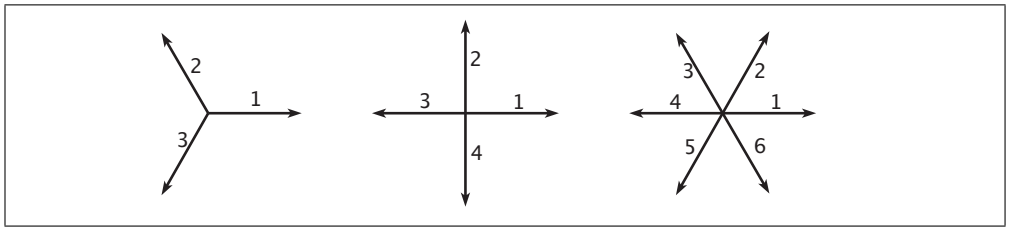

Figure 1.4-A: Numbering for the directions in the triangular grid (left), square grid (middle), and the hexagonal and tri-hexagonal grid (right).

Description by directed edges. The curves with simple L-systems can also be described by L-systems specifying the directions of the edges in the drawing. There are 3 possible directions on the triangular grid. We assign the letters `1` (right), `2`

(upward left), and 3 (downward left) to them, as shown in Figure 1.4-A. The word for the motif (left image in Figure 1.3-D), here 1221, is the production of its leftmost letter, giving the map 1 ↦ 1221. The other maps corresponding to the remaining rotations are 2 ↦ 2332 and 3 ↦ 3113. The map for $k \geq 2$ is obtained by incrementing all numbers in the map for $k-1$, taking the result modulo 3. The image in the middle of Figure 1.3-D shows the second iterate with a coloration that allows to identify the corresponding parts in the motif.

The L-system for the R5-dragon shown in Figure 1.1-B is F ↦ F+F+F-F-F and the maps for the edge directions are 1 ↦ 12321, 2 ↦ 23432, 3 ↦ 34143, and 4 ↦ 41214. The interpretations as directions are 1 (right), 2 (up), 3 (left), and 4 (down).

The maps of these L-systems by definition commute with the map σ that replaces k by $k+1 \bmod n$ where n is the number of directions. Dekking's *folding morphisms* [10, Section 2] commute with $\sigma\tau$ where τ is the map that reverses the productions. These curves can also be described by L-systems with two non-constant maps for, say, L and R as follows. The production of L starts with L and the letters L and R appear alternatingly, with turns (+ and -) between them. The production of R is obtained from that of L by reversing and swapping L with R and + with -. For example, the maps L ↦ L+R-L-R and R ↦ L+R+L-R correspond to the curve shown in figure 3.2-C on page 20.

folding morphisms

The intersection of the sets of Dekking's curves and the curves with simple L-systems consists of the curves with two-fold symmetry.

Description by turns. Yet another way to describe curves with simple L-systems is by an L-system for the succession of turns. The maps for the turns T (where $T \in \{$ +, -, 0 $\}$ and 0 denotes a turn by 0°, a non-turn) are the turns in the production of F (in the simple L-system) followed by T. Edges are drawn at the start and after each turn. For example, the curve R4-1 in Figure 1.3-D has the L-system F ↦ F+F0F-F, the sequence of turns is +0-, and we get the maps + ↦ +0-+, - ↦ +0--, and 0 ↦ +0-0. For the R5-dragon with map F ↦ F+F+F-F-F, we get the maps + ↦ ++--+ and - ↦ ++---. The choice of the axiom does not matter as the maps are identical except for the last turn. This description is equivalent to the method of looking up the nth turn ($n \geq 1$) from the lowest non-zero digit in the base-R expansion of n indicated in [3, Section 1.31.5, pp. 95ff].

L-systems for the Hilbert curve. The following descriptions of the Peano and Hilbert curves by L-systems are slightly more complicated than for other curves shown. The Peano curve can be described by the L-system with axiom L and (non-constant) maps L ↦ LtRtL-t-RtLtR+t+LtRtL, R ↦ RtLtR+t+LtRtL-t-RtLtR. Here only the letter t corresponds to an edge, both L and R are ignored in the drawing process. For the Hilbert curve a possible L-system with axiom L and (non-constant) maps L ↦ +Rt-LtL-tR+, R ↦ -Lt+RtR+tL- can be used (only t corresponds to an edge).

Again, the descriptions by strokes and turns is by no means the only way to specify these curves. For the Hilbert curve the L-system with axiom A and non-constant maps A ↦ DrAuAlC, B ↦ ClBdBrD, C ↦ BdClCuA, D ↦ AuDrDdB can be used. The drawing is obtained by ignoring the capital letters and using r, d, l, and u as moves respectively "right", "down", "left", and "up", see [3, Figure 1.31-C, p. 85]. Bader [4, Section 3.3, pp. 37ff] gives a similar description for the Peano curve.

2 The search

2.1 Conditions for a curve to be self-avoiding and edge-covering

Michel Dekking's 2012 paper "Paperfolding morphisms, planefilling curves, and fractal tiles" [10] has been crucial for this project.

Let X be the production of `F` in a simple L-system of order R. Let C_n be the nth iterate of the curve corresponding to the L-system and C be the set of all iterates C_n of the curve $n \geq 0$. We say C is self-avoiding or edge-covering if every curve in C has the respective property.

2.1.1 Sufficient conditions

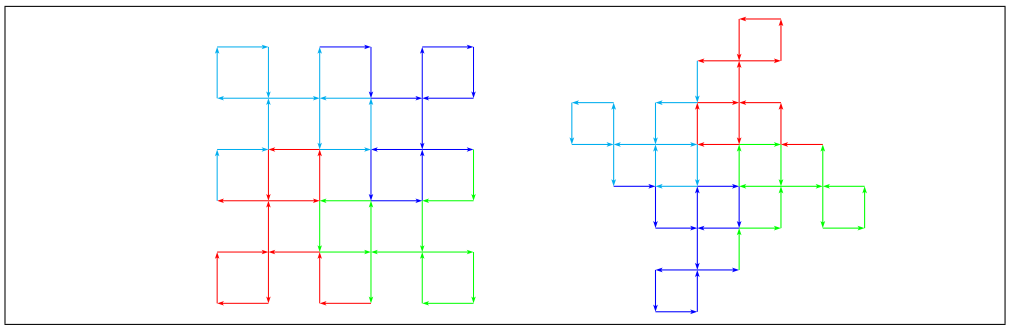

Figure 2.1-A: Tiles Θ_{+1} (left) and Θ_{-1} (right) for the curve of order 13 on the square grid with L-system `F` $\mapsto$ `F+F-F-F+F+F+F-F+F-F-F-F+F` (`R13-3`).

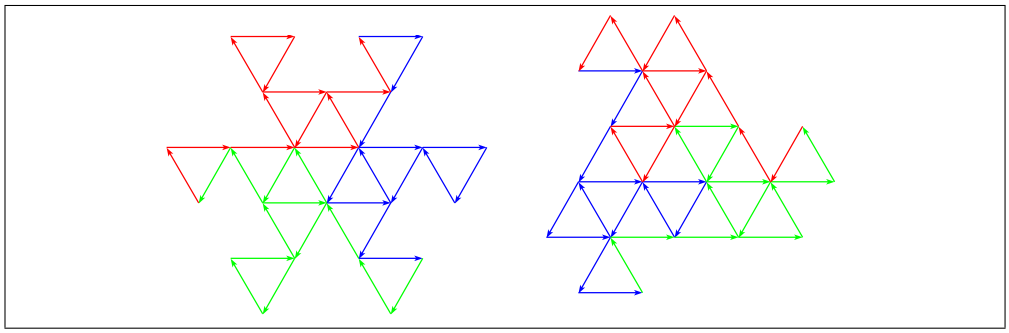

Figure 2.1-B: Tiles Θ_{+1} (left) and Θ_{-1} (right) for the curve of order 13 on the triangular grid with L-system `F` $\mapsto$ `F+F0F0F-F-F+F0F+F+F-F0F-F` (`R13-15`).

We need the concept of a *tile* for the following facts. Here we define them only for the square and the triangular grid, the tiles for the tri-hexagonal grid are described in section 3.4 on page 26. *tile*

For the square grid, let Θ_{+1} be the (closed) curve corresponding to the first iterate of the map of the L-system with axiom `F+F+F+F` and Θ_{-1} for the axiom `F-F-F-F`. Here the turns are by $\phi = 90°$, see Figure 2.1-A for an example. For the triangular grid, the respective axioms are `F+F+F` and `F-F-F`, and the turns are by $\phi = 120°$, see Figure 2.1-B.

The tiles of edge-covering curves do indeed tile the grid: infinitely many disjoint translations of them do cover all edges of the grid, see section 3.1.

Fact 1 (Tiles-SA)**.** *C is self-avoiding if and only if both tiles Θ_{+1} and Θ_{-1} are self-avoiding.*

Dekking's proof for the curves described by folding morphisms [10, Theorem 1, p. 24] can be adapted for the curves with simple L-systems [Dekking, personal communication, March 2016]. Note that for Dekking's curves on the square grid, it suffices that one of the two tiles is self-avoiding

We call a tile edge-covering if all edges in its interior are traversed once.
Fact 2 (Tiles-Fill)**.** *C is edge-covering if and only if both tiles Θ_{+1} and Θ_{-1} are edge-covering.*

The proof for Theorem 2 in [10, p. 27] can be modified to show this [Dekking, personal communication, March 2016].

What is called tile (for a self-avoiding, edge-covering curve) here is called a (maximally simple) Θ-loop by Dekking [10, Definition 3, p. 24].

2.1.2 Necessary conditions

The motif must obviously be self-avoiding:
Fact 3 (Obv)**.** *For C to be self-avoiding, C_1 must be self-avoiding.*
Fact 4 (Turn)**.** *For C to be self-avoiding and edge-covering, the net rotation of the curve C_1 must be zero.*

That is, the number of `+` and `-` in X must be equal. If this condition does not hold, the tiles cannot be self-avoiding.
Fact 5 (Dist)**.** *For C to be self-avoiding and edge-covering, the squared distance between the start and the endpoint of the curve must be equal to R.*

If this condition does not hold, the tiles cannot be both edge-covering.

Also, if the dimension of the limiting curve is $\log R/\log d = 2\log R/\log d^2$ where d is the distance (see [21, Section 1.1.4, pp. 28ff]), we need $R = d^2$ for dimension 2.

For the square grid, the possible orders are the numbers of the form $x^2 + y^2$, sequence A001481 in [24], otherwise numbers of the form $x^2 + x\,y + y^2$ (equivalently, numbers of the form $3\,x^2 + y^2$), sequence A003136 in [24].

The curves on the square grid must turn at every point, otherwise two dead ends would be created. As we do not allow turns as first or last letters in X, we will find only curves of odd orders, sequence A057653 in [24].

2.2 The shape of a curve

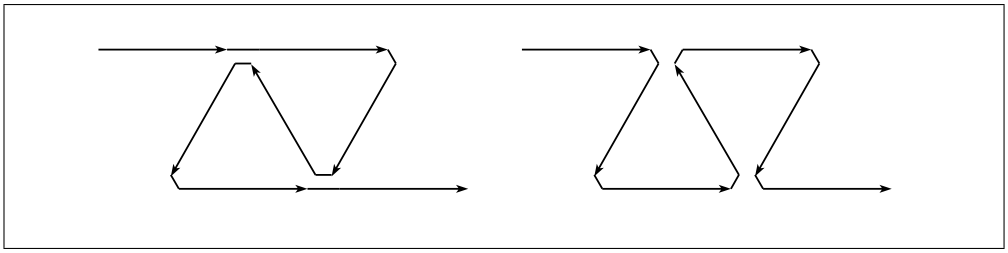

Figure 2.2-A: Two different curves of order 7 with same shape (`R7-2` and `R7-5`).

The *shape* of a curve is the set of grid points traversed in the first iterate. Different *shape*

curves can have the same shape, as shown in Figure 2.2-A. The L-systems are respectively `F` $\mapsto$ `F0F+F+F-F-F0F` and `F` $\mapsto$ `F+F-F-F+F+F-F`. Here the letter `0` (interpreted as "no turn") is used to keep the maps for one order of fixed length. These two curves were already given in [3, Figures 1.31-P and 1.31-Q, pp. 97-98].

We consider two curves to be of the same shape whenever any transformation of the symmetry of the underlying grid (rotations and flips) maps one shape into the other. If two curves have the same shape, we call them *similar*. *similar*

Rendering only one curve for each shape makes the results (visually) much more manageable, as for higher orders more and more curves tend to have the same shape.

2.3 Structure of the program for searching

The program consists of the following parts: generation of the L-systems, testing of the corresponding curves, detection of similarity to shapes seen so far, and detection of symmetries.

The generation of the L-systems is equivalent to counting in base 2 or 3: For the square grid the allowed turns are by $+90°$ and $-90°$. For the triangular grid turns by $+120°$, $-120°$, and $0°$ are possible. For the tri-hexagonal, two values again suffice, see section 3.4.

Testing for self-avoidance and the edge-covering property uses the conditions of section 2.1 in some order where the criteria cheaply testable are checked before the more costly ones. Condition 4 [Turn] is tested before the curve is even computed (as a sequence of points), then 5 [Dist], followed by 3 [Obv], 1 [Tiles-SA], and 2 [Tiles-Fill].

Every curve found is assigned a positive ID. The triple (grid, order, ID) identifies a curve.

The shape of the curve is computed as a sorted sequence of points without duplicates. For each curve found, the shape is compared to the shapes found so far. If a match is detected, the new curve is marked a duplicate shape. If a curve is a duplicate, the lowest ID of a curve of the same shape and the transformations to match the shapes are documented.

The symmetries (if any) of each curve are computed and documented.

The generation of all L-systems is very simple but certainly not very elegant. Still, after careful optimization, the tests and the generation of the L-systems each use about half of the CPU time, and approximately one million curves per second are checked. Due to the combinatorial explosion, searches for large orders would take very long. The largest orders searched and their approximate duration were $R = 53$ for the square grid (7 days), $R = 31$ for the triangular grid (6 days), and $R = 61$ for the tri-hexagonal grid (76 days). Parallelization of the existing program is not planned, instead a completely different method for searching will be used.

2.4 Format of the files specifying the curves

Figure 2.4-A shows the output format for the curves of order 17 on the square grid.

The descriptions of the curves found are available for download at `http://jjj.de/3frac/` in the file `short-lsys.tar.xz` (of size 4.3 MB) which unpacks into a directory named `short-lsys/` of size 136 MB. Therein the subdirectories `c3/`, `c4/`, and `c6/` contain respectively the descriptions of the curves in the triangular, square, and tri-hexagonal grids. The file for an order, say $R = 17$, is named `lsys-r17.txt`.

```
F F+F+F-F+F-F-F-F+F-F+F+F+F-F+F-F-F  R17-1  # # symm-dr
F F+F+F-F-F+F+F+F-F+F-F-F-F+F+F-F-F  R17-2  # # symm-dr
F F+F+F-F-F+F+F+F-F-F+F+F-F-F-F+F-F  R17-3  #
F F+F+F-F-F-F+F+F+F-F+F+F-F-F-F+F-F  R17-4  # # symm-r ## same = 1 P R
F F+F-F+F+F+F-F-F+F+F-F-F-F+F+F-F-F  R17-5  # ## same = 3 R X
F F+F-F+F+F+F-F-F+F-F+F+F-F-F-F+F-F  R17-6  # # symm-dr
F F+F-F+F+F+F-F-F+F-F-F-F+F+F+F-F-F  R17-7  # # symm-r ## same = 1 P R
F F+F-F+F+F+F-F-F-F+F+F+F-F-F-F+F-F  R17-8  # # symm-dr ## same = 1 P R
F F+F-F+F+F-F+F+F+F-F-F-F+F-F-F+F-F  R17-9  # # symm-dr ## same = 1 P R
F F+F-F+F+F-F+F+F-F-F-F+F+F-F-F-F+F  R17-10  #
F F+F-F+F+F-F+F-F+F+F-F-F-F+F-F-F+F  R17-11  #
F F+F-F-F+F-F-F-F+F+F-F+F-F+F+F-F+F  R17-12  # ## same = 11 Z T
F F+F-F-F-F+F+F-F-F-F+F+F-F+F+F-F+F  R17-13  # ## same = 10 Z T
```

Figure 2.4-A: Descriptions of the curves of order 17 on the square grid.

2.4.1 L-system and ID

Each line starts with the non-constant map for the L-system: `F X` where `X` is the production of `F`. No listed L-system has a map with initial turn `-`: these can be omitted because such a map is either the reversal of some map with last turn `-` or can be obtained from a map with `+` both as initial and final turn by swapping `-` and `+`.

The following string (like `R17-6`) is of the form `R<order>-<ID>`. The order of the curves is constant within a file and `ID` identifies a curve among all curves of the same order on the same grid.

2.4.2 Information about symmetries of the curve

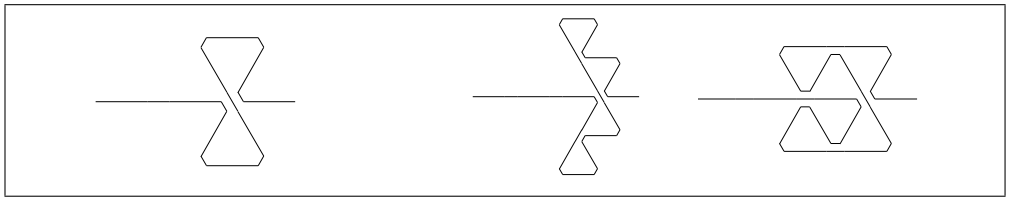

Figure 2.4-B: Motifs of curves on the triangular grid with shape invariant to reflection of the y-axis (`R9-2`, `R16-1`, and `R16-4`).

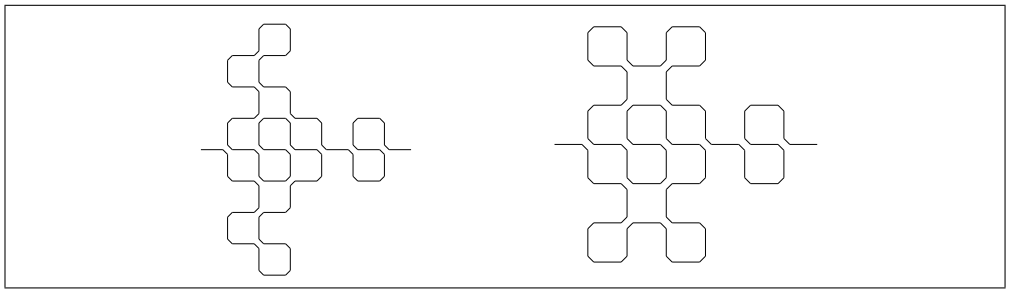

Figure 2.4-C: Motifs of two curves of order 49 on the square grid invariant to reflection of the y-axis (`R49-14164` and `R49-15059`).

Information about symmetry and similarity of the shape is given after the character `#`.

If a curve has any non-trivial symmetries then these are specified as letters following `symm-`. A `d` indicates that the word `X` is invariant under the map reversing it and

swapping `-` and `+`. These curves do have a 2-fold (rotational) symmetry. An `r` indicates 2-fold symmetry, so whenever there is a `d` there is an `r`.

In the following it is assumed that the first edge goes in positive x-direction. The letter `m` stands for invariance under reflection of the y-axis. Curves with this symmetry are shown in Figures 2.4-B (orders 9 and 16) and 2.4-C (order 49). Curves with just this symmetry on the triangular grid exist for orders $R = n^2$ where $n \pmod 3 \not\equiv 2$, and on the square grid for $R = n^2$ with n odd.

The specifications `symm-mrqz` (or `symm-dmrqz`) are given for curves whose shapes are perfect squares, these exists only on the square grid and for orders that are squares.

2.4.3 Information about similarities to shapes of other curves

If a curve has a shape similar (via any transformation of the grid) to any curve found earlier then this is documented as `same = ID` where `ID` refers to the earlier curve. Among multiple possible choices of similar curves the one with the lowest ID is taken.

The letters following specify which transformations map the shape of the new curve into the shape of the earlier curve. Here `P` is put if the two shapes are identical (without any transformation), `M` for reflection of the y-axis, `R` for reversing directions of all edges, `Z` for reversing directions and reflection of the y-axis, `T` if the turns of the curves after reversing one L-system are identical, and `X` if the turns of the curves after reversing one L-system and swapping signs are identical. For shapes with non-trivial symmetries more than one letter may be given.

2.5 Numbers of shapes found

We now give the numbers n of shapes found for the orders R (up to the search limits) where any curve exists as lists of entries $R : n$.
Triangular grid, sequence A234434 in [24]:
3:1, 4:1, 7:3, 9:5, 12:10, 13:15, 16:17, 19:71, 21:212, 25:184, 27:543, 28:842, 31:1848.
Square grid, sequence A265685 in [24]:
5:1, 9:1, 13:4, 17:6, 25:33, 29:39, 37:164, 41:335, 49:603, 53:2467.
Tri-hexagonal grid, sequence A265686 in [24]:
7:1, 13:3, 19:7, 25:10, 31:63, 37:157, 43:456, 49:1830, 61:8538.

The number of curves is much greater than the number of shapes. For example, for order $R = 53$ on the square grid there are 2467 shapes and 401738 curves, so about 162 curves share one shape on average. The corresponding file lists the 200869 curves where the map for `F` does not start with F-F. These are counted by adding the number of entries ending in F-F (131111) to the number (69758) of entries beginning and ending in F+F (which after swapping signs give yet unseen L-systems).

3 Properties of curves and tiles

3.1 Self-similarity, symmetries, and tiling property

All curves are self-similar by construction: every curve of order R can be decomposed into R disjoint rotated copies of itself. Figure 3.1-A shows this for a curve of order 13 with L-system `F` $\mapsto$ `F+F0F0F-F-F+F0F+F+F-F0F-F` on the triangular grid.

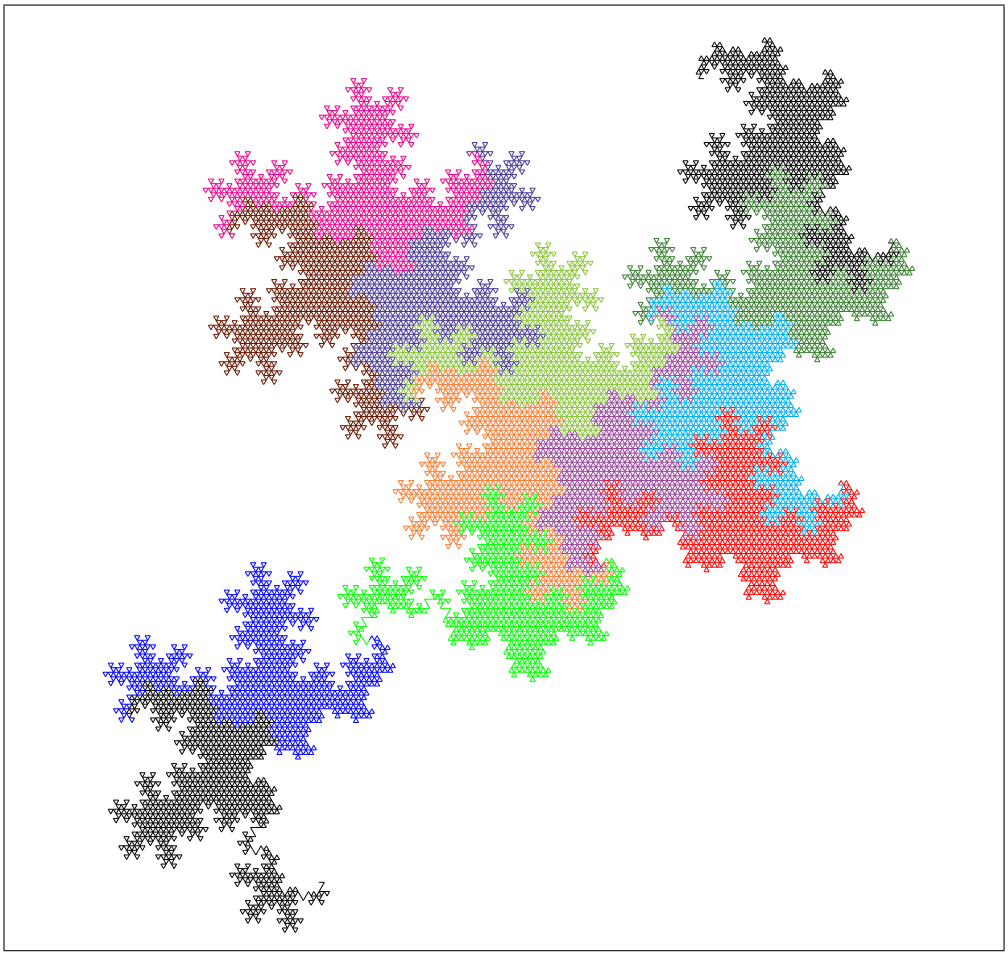

Figure 3.1-A: Self-similarity of the order-13 curve `R13-15` on the triangular grid.

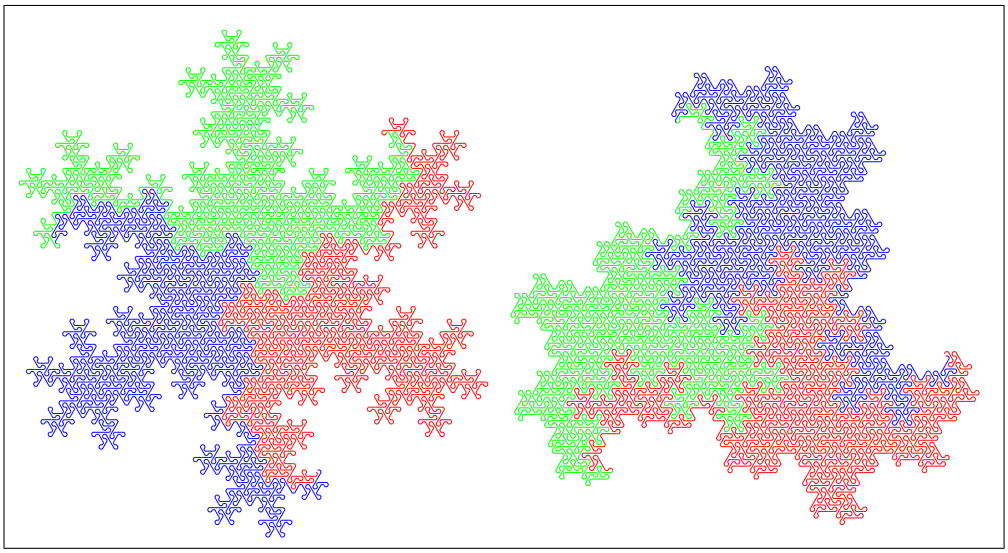

Figure 3.1-B: The two tiles Θ_{+3} and Θ_{-3} of the curve in the previous figure (`R13-15`).

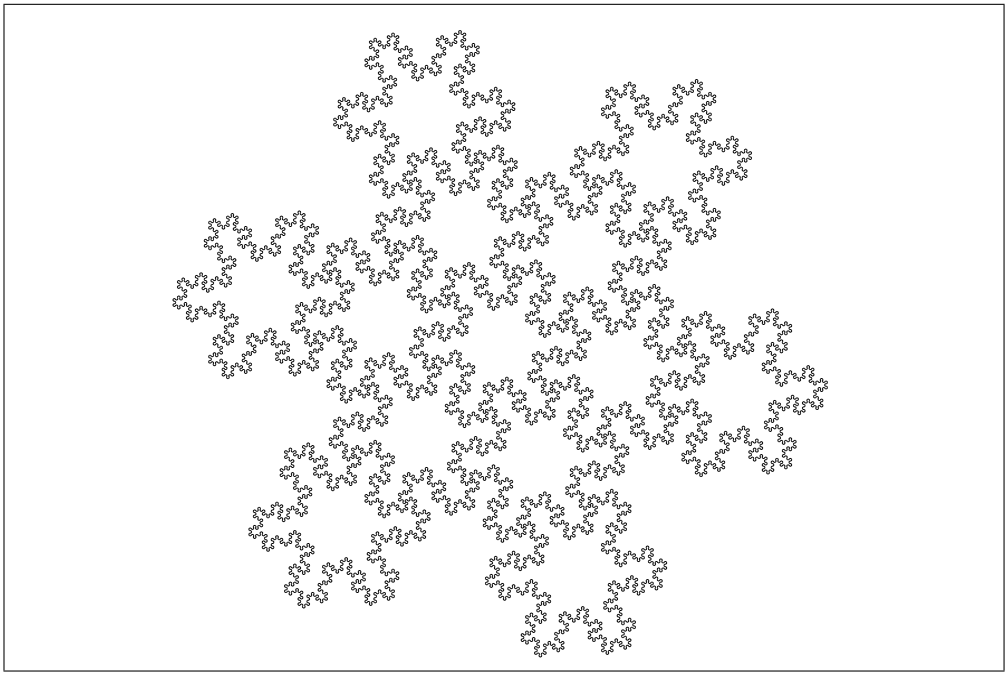

Figure 3.1-C: The shape $\Theta_{+\infty}$ of the tile at the left in the previous figure, decomposed into 13 smaller rotated copies of itself.

Let Θ_{+k} and Θ_{-k} be the tiles for the kth iterate of a curve, and $\Theta_{+\infty}$ and $\Theta_{-\infty}$ the limiting shape of the tiles. The tiles Θ_{+3} and Θ_{-3} of this curve are shown in Figure 3.1-B (for Θ_{+1} and Θ_{-1} see Figure 2.1-B on page 10). The tile can be decomposed into 13 small copies of itself as shown in Figure 3.1-C.

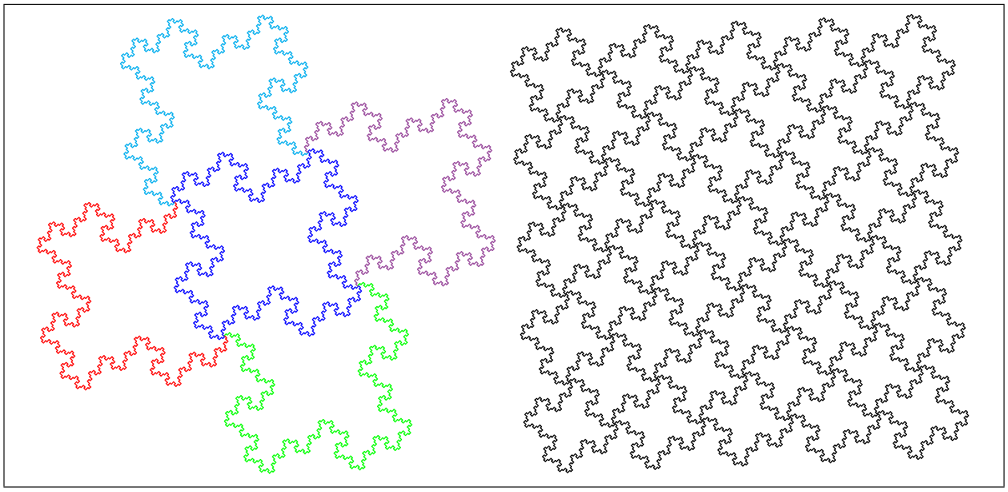

Figure 3.1-D: Decomposition of the tile $\Theta_{+\infty}$ of the R5-dragon (`R5-1`) into rotated copies of itself (left) and the tiling of the plane by such tiles (right).

Such a statement is true for all tiles: the tiles of a curve of order R can always be decomposed into R scaled (by $1/\sqrt{R}$) rotated copies of itself. Figure 3.1-D (left) shows this for the R5-dragon, the tiling of the plane by translations of the tile is shown on the right. The curves tile the plane by translations and rotations.

Our tiles are always connected, and the tilings are lattice tilings (the positions of the tiles are on a lattice).

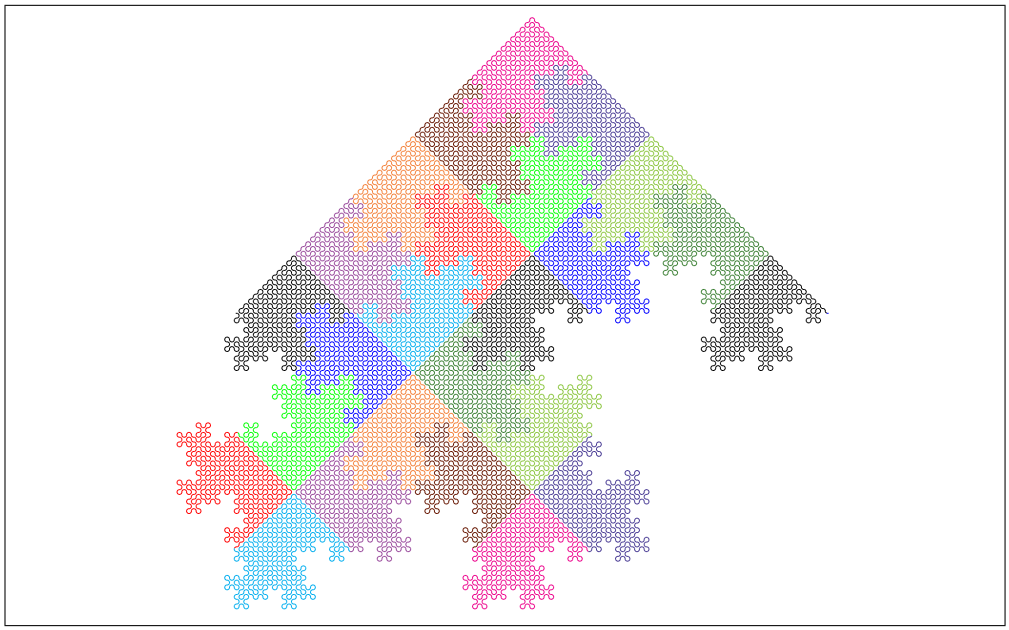

Figure 3.1-E: Self-similarity of a curve of order 25 on the square grid (`R25-46`).

The tiles for the curves on the square grid always have 4-fold rotational symmetry. Figure 3.1-E shows the self-similarity of the curve `R25-46` on the square grid. The tiles appear as groups of four smaller curves.

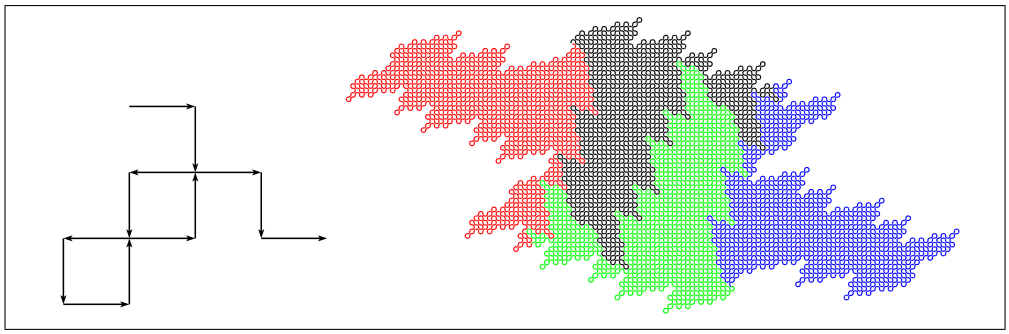

Figure 3.1-F: Motif of a curve of order 13 with an L-system with two non-constant letters `L` and `R` (left) and third iterate of the tile with axiom `L+R+L+R` (right).

While not found in our search, curves on the square grid with tiles having only 2-fold rotational symmetry exist. Figure 3.1-F shows the motif of such a curve of order 13 generated by an L-system with two non-constant letters `L` and `R` (left) and the tile from the axiom `L+R+L+R` (right). The maps for the curve are `L` $\mapsto$ `L+R+L-R+L-R-L-R+L-R+L+R-L` and `R` $\mapsto$ `R+L-R-L+R-L+R+L+R-L+R-L-R`, the production for `R` can be obtained from that for `L` by reversing and swapping all `+` and `-` and all `L` and `R`.

On both grids the two tiles Θ_{+j} and Θ_{-j} for $j \geq 1$ (and $j = \infty$) have different shapes in general (the only exception are tiles that are perfect squares). If the curve has 2-fold rotational symmetry, then both tiles are identical.

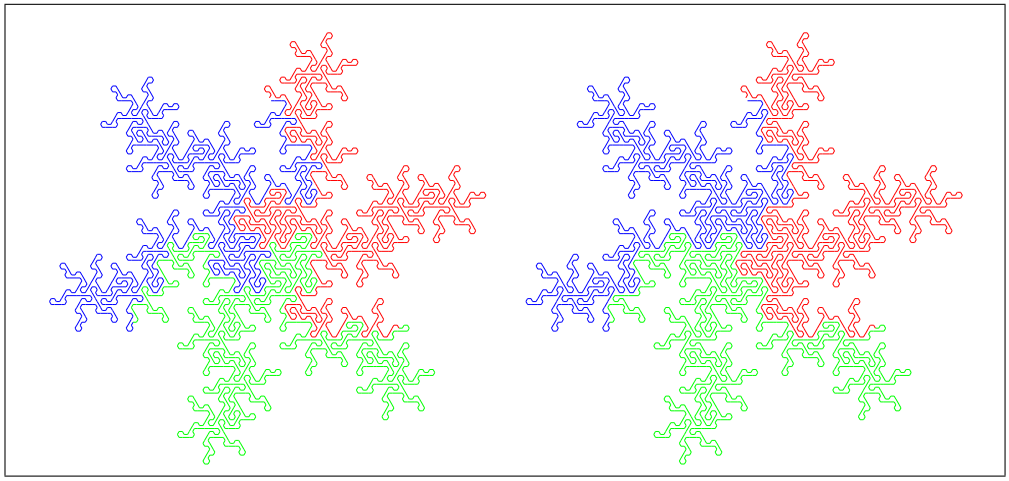

Figure 3.1-G: Two curves with different shapes but same shape of the tiles Θ_{-k} for all k, shown are the tiles Θ_{-2} (`R25-247` and `R25-248`).

Note that curves of different shapes can lead to the same shape of tiles, see Figure 3.1-G for an example with curves of order 25 on the triangular grid.

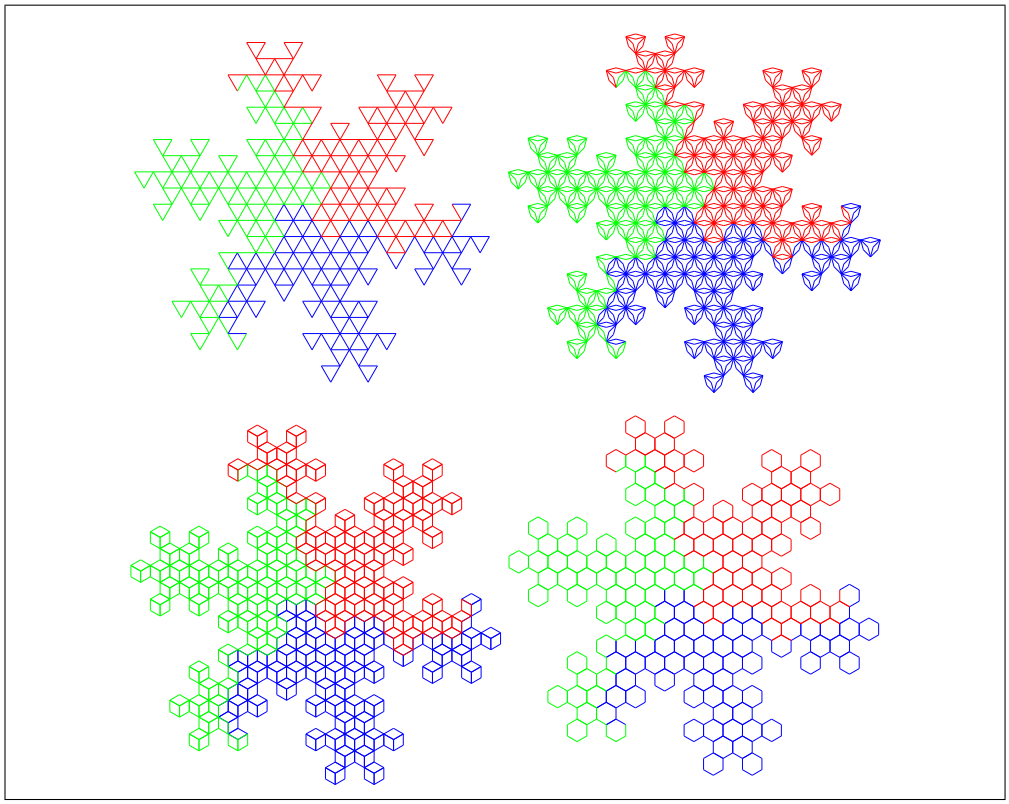

Figure 3.1-H: Rendering a tile with 6-fold symmetry, see text.

The tiles for the curves on the triangular grid always have 3-fold rotational symmetry. Some tiles appear to have 6-fold rotational symmetry but this is only true in the limit. With some artistic license such tiles can be rendered with 6-fold symmetry, we use Θ_{+2} of `R13-15` (for Θ_{+3} see Figure 3.1-B). The upper left of Figure 3.1-H shows that

the tile does not actually have a 6-fold symmetry. In the rendering to the right each line has been replaced with a thin lozenge. Choosing 60° for the acute angle of the lozenges leads to the rendering in the lower left, an apparent arrangement of cubes. If we keep just the outline of the cubes (lower right), we obtain the arrangement of hexagons corresponding to a 6-symmetric tile of the hexagonal grid.

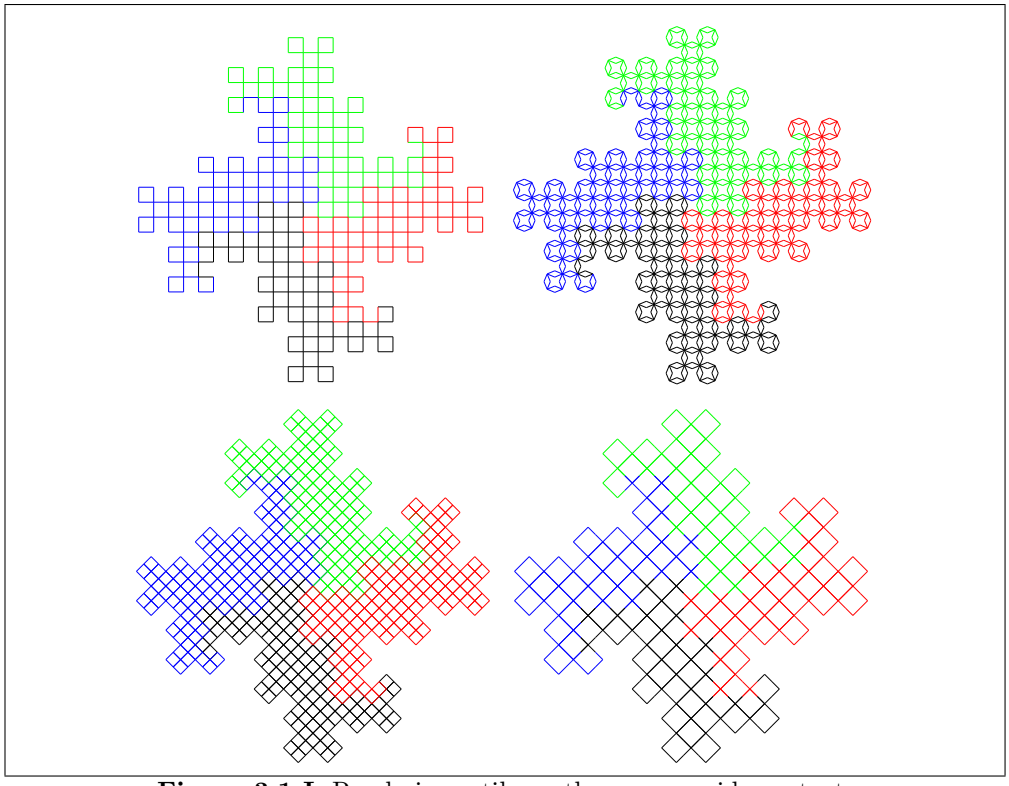

Figure 3.1-I: Rendering a tile on the square grid, see text.

The analogous rendering for the square grid is shown in Figure 3.1-I for the tile Θ_{+3} of the R5-dragon. Here the full "lozenges" shown in the lower left are squares that are rotated by 45° against the original grid.

3.2 Certain arrangements of curves

In the tilings of the triangular grid there are points where six curves end. These curves give an arrangement with 3-fold symmetry as shown for the curve `R7-1` in Figure 3.2-A.

For curves with 2-fold symmetry, such as the terdragon, the arrangement has 6-fold symmetry as shown in Figure 3.2-B. This arrangement is also shown in [7, Figure 19, p. 596], also see [10, Section 8, pp. 29-33] where these arrangements are called "carousels", and curves for which the arrangements fill the plane are called "perfect". An example of a curve that is not perfect is shown in Figure 3.2-C, called "alternate dragon curve" in [7, Figure 10, p. 590].

For curves on the square grid these arrangements have a 2-fold symmetry and a 4-fold symmetry for curves with 2-fold symmetry, as shown in Figure 3.2-D for the curves `R17-11` and `R17-6`.

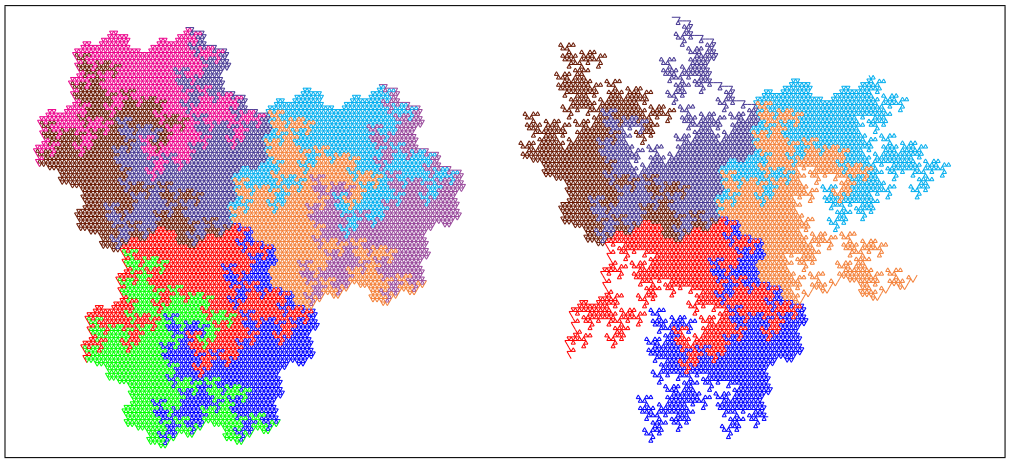

Figure 3.2-A: Three tiles of the curve `R7-1` in the tiling of the plane (left) and the arrangement of the curves meeting at the center (right).

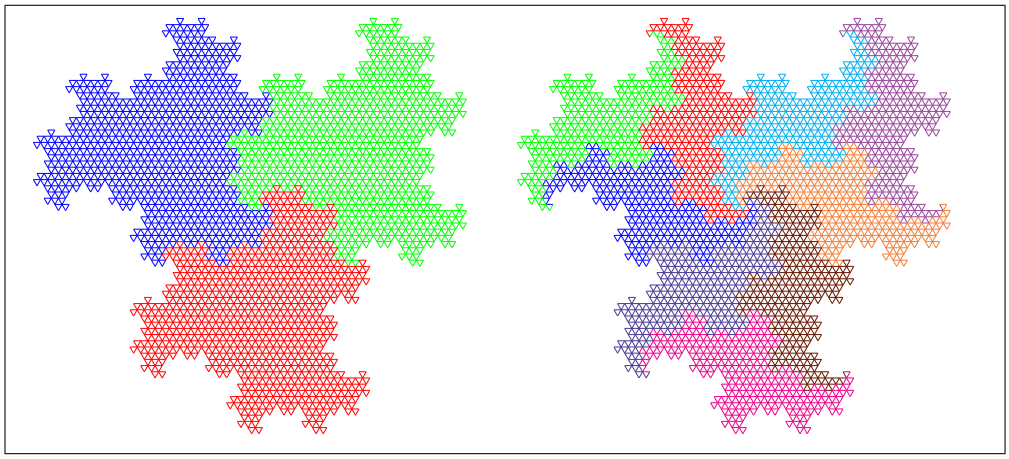

Figure 3.2-B: Three tiles of the terdragon in the tiling of the plane (left). The six curves meeting at the center give an arrangement with 6-fold symmetry (right).

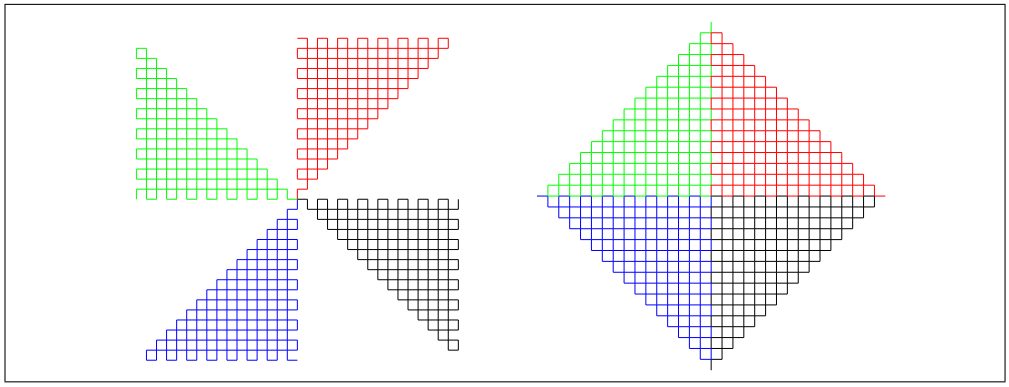

Figure 3.2-C: Only one of the arrangements of the curve with maps `L` $\mapsto$ `L+R-L-R` and `R` $\mapsto$ `L+R+L-R` contains a neighborhood of the origin.

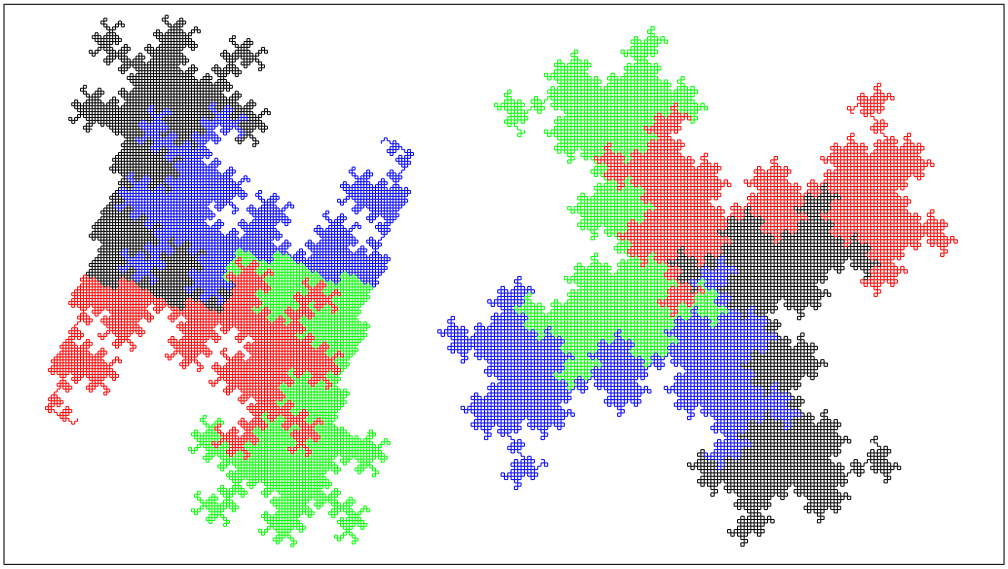

Figure 3.2-D: The 2-symmetric arrangement of curves `R17-11` on the square grid (left) and the 4-symmetric arrangement of 2-symmetric curves `R17-6` (right).

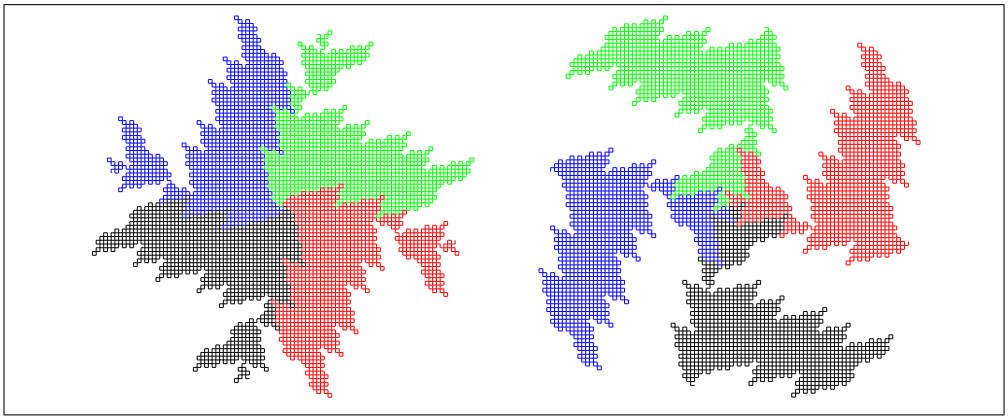

Figure 3.2-E: The two 4-symmetric arrangements of curves of order 13 on the square grid generated by an L-system with two non-constant letters.

Unsymmetric curves with arrangements of 4-fold symmetry exist. The arrangements for the curve already shown in Figure 3.1-F on page 17 are shown in Figure 3.2-E, the curves on the right are flipped versions of those on the left. This happens for curves with L-systems with two non-constant letters, say `L` and `R`, and maps `L` $\mapsto$ `f(L)`, `L` $\mapsto$ `g(L)` where the letters `L` and `R` alternate in `f(L)` and `g(R)` is the reversed word `f(L)` with letters `L` and `R` swapped. If such a curve has 2-fold symmetry, a curve of the same shape with simple L-system exists.

3.3 Tiles and complex numeration systems

A *numeration system* is a pair (B, D) where B is the base and D the set of digits. The expansion of a number in a numeration system has the form $\sum_{k=-\infty}^{m} d_k\, B^k$ where $d_k \in D, m \in \mathbb{N}$. We call the part left of the radix point $(\sum_{k=0}^{m} d_k\, B^k)$ the *integer part* and the part right of the radix point $(\sum_{k=-\infty}^{-1} d_k\, B^k)$ the *fractional part*.

numeration system

integer part

fractional part

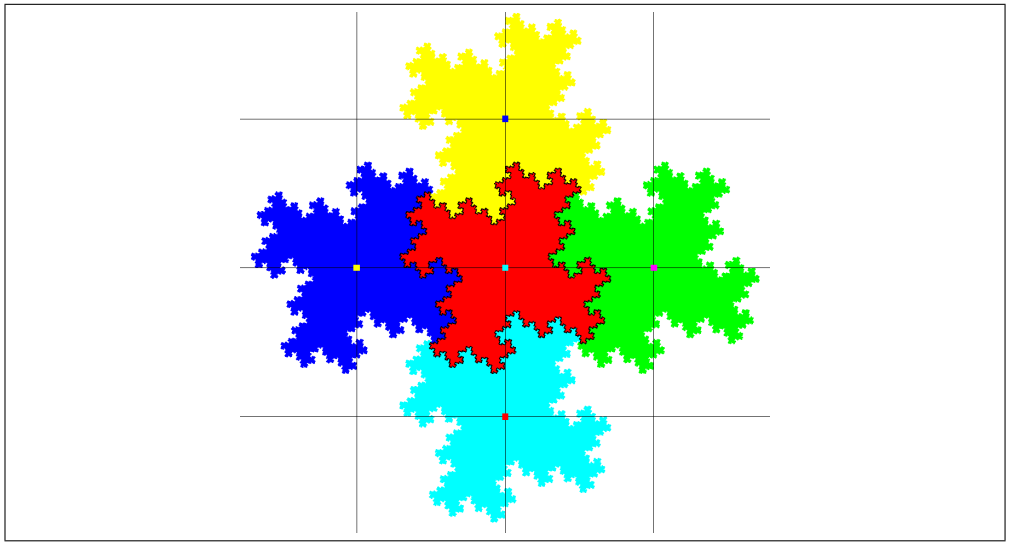

Figure 3.3-A: The fundamental region for the complex numeration system with base $2+i$ and digit set $\{0,+1,-1,+i,-i\}$ where $i=\sqrt{-1}$. The set is scaled up by the factor $\sqrt{5}$ to let the digits (small squares) lie in the five subsets.

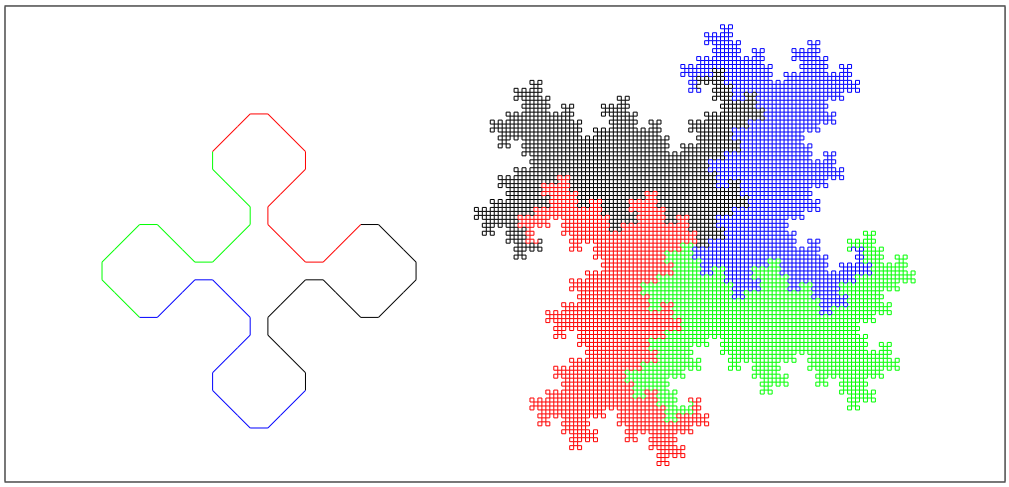

Figure 3.3-B: The tiles Θ_{+1} (left) Θ_{+5} (right) of the curve `R5-1` on the square grid.

The well-known positional numeration systems are defined by an integer base $B \geq 2$ and a digit set D, usually $D=\{0,1,\ldots,B-1\}$. We call a numeration system *integral* if each integer has a unique representation as $\sum_{k=0}^{m} d_k\, B^k$ (a representation without fractional part). The positional numeration systems with digits $D=\{0,1,\ldots,B-1\}$ are integral. All numbers can be represented in an integral numeration system [17, Section 7.4, pp. 257ff]. *integral*

We consider complex numeration systems where both the base B and the digits in D are in general complex. Let $\omega_k = \exp(2\pi i/k)$ where $i=\sqrt{-1}$ be a primitive complex kth root of unity, we will only use the values $\omega_3=(-1+i\sqrt{3})/2$, $\omega_4=i$, and $\omega_6=(1+i\sqrt{3})/2$. The integers of interest are the Gaussian integers (numbers $x+\omega_4\, y = x+i\, y$ where $x,y\in\mathbb{Z}$, points on the square grid) and the Eisenstein integers (numbers $x+\omega_3\, y$ where $x,y\in\mathbb{Z}$, points on the triangular grid). We require that both the base and all digits are such integers.

Following Knuth [16, Section 4.1] we call the set of numbers of the form $\sum_{k=-\infty}^{-1} d_k\, B^k$ (that is numbers of the form $0.*$ where $*$ is any sequence of digits) the *fundamental region* of the numeration system. *fundamental region*

An example for the square grid is the numeration system with base $B = 2 + i$ and digit set D=$\{0, +1, -1, +i, -i\}$ whose fundamental region is shown in Figure 3.3-A. The set consists of five subsets that are similar to the whole set: the subsets shown in red, green, blue, yellow, and cyan are respectively the subsets of numbers of the form $0.(0)*$, $0.(+1)*$, $0.(-1)*$, $0.(+i)*$ $0.(-i)*$. The positions of the values of the digits are indicated by the small squares inside the subsets (for this purpose the set is shown scaled up by $\sqrt{\mathrm{abs}(B)} = \sqrt{5}$).

The tiles of our curves correspond to complex numeration systems. The shape of the tile Θ_{+5} of the curve `R5-1` on the square grid (with L-system `F` $\mapsto$ `F+F+F-F-F`) shown at the right of Figure 3.3-B matches (up to rotation) the form of the fundamental region shown in 3.3-A. The values of the digits correspond to the positions of the squares surrounded by the tile Θ_{+1} in the left of Figure 3.3-B.

For every tile we can find a complex numeration system and identify the tile with the fundamental region as follows. Let R be the order of the underlying curve. There exists a complex base B such that $\mathrm{abs}(B)^2 = R$ and a set of digits $\{d_0, d_1, \ldots, d_{R-1}\}$ corresponding to the squares surrounded by Θ_{+1} such that the shape of the tile $(\Theta_{+\infty})$ matches the fundamental region of the numeration system. We take the digit 0 for the center of the tile where possible: on the square grid the orders are of the form $R = 4k + 1$ and there always is a central digit, on the triangular grid there is a central digit for orders of the form $R = 3k + 1$, otherwise (orders $R = 3k$) we choose one of the three digits around the center. The subdivision into sets for numbers of the forms $0.d_j*$ for $0 \le j < R$ reproduces the tiling of the fundamental region (and of $\Theta_{+\infty}$) into R smaller rotated copies of itself. The same holds for Θ_{-1} and $\Theta_{-\infty}$, giving a different numeration system in general.

The digits form a *complete residue system* (D contains exactly one representative of each coset modulo B). Note that in general there is more than one choice for the base. *complete residue system*

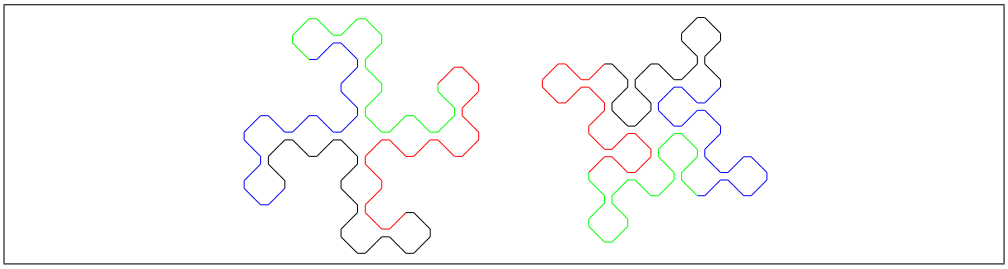

Figure 3.3-C: The tiles Θ_{+1} of the curves `R13-1` (left) and `R13-4` (right) on the square grid.

We give two more examples for the square grid.

The tiles Θ_{+1} for the order-13 curves with maps `F` $\mapsto$ `F+F+F-F+F+F-F+F-F-F+F+F-F` and `F` $\mapsto$ `F+F-F-F-F+F+F-F-F+F+F+F-F` are shown in Figure 3.3-C. The 13 squares inside the tiles give the digit sets, $D = \{0, \pm1, \pm i, \pm2, \pm2i, +2{+}i, -2{-}i, +1{-}2i, -1{+}2i\}$ for the curve `R13-1` and $D = \{0, \pm1, \pm i, \pm1{\pm}i, -2{+}i, +2{-}i, +1{+}2i, -1{-}2i\}$ for the curve `R13-4`. The base $B = 3 + 2i$ can be chosen for both numeration systems. The fundamental regions of the numeration systems are shown next to the tiles Θ_{+3} in Figures 3.3-D and 3.3-E.

The fundamental region for the numeration system with base $B = \sqrt{3}\,i$ and digit

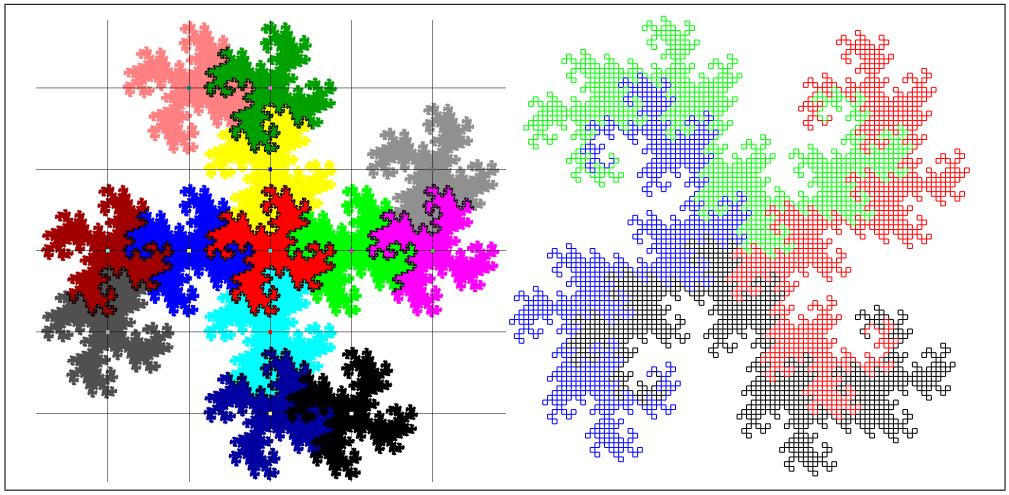

Figure 3.3-D: Fundamental region of a numeration system (left, scaled up by $\sqrt{13}$) and the tile Θ_{+3} of the curve `R13-1` on the square grid (right).

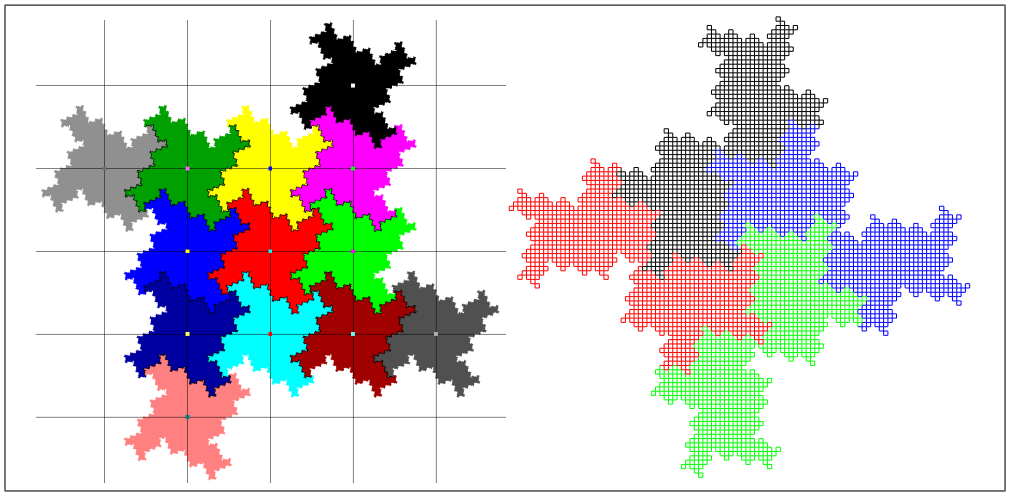

Figure 3.3-E: Fundamental region of a numeration system (left, scaled up by $\sqrt{13}$) and the tile Θ_{+3} of the curve `R13-4` on the square grid (right).

set D=$\{0, +1, \omega_6\}$ (the number ω_6 is an Eisenstein integer) is shown in Figure 3.3-F. The set consists of the three subsets shown in red, green, and blue, respectively for numbers of the form $0.(0)*$, $0.(1)*$, and $0.(\omega_6)*$.

Helmberg [14, Section 4, pp. 374ff] shows that the tile for the curve in Figure 1.3-D corresponds to the numeration systems with $B = -2$ and digits $D = \{0, 1, \omega_3, \omega_3^2\}$.

The numeration systems mentioned so far are all integral. A sufficient condition for that is that a non-empty neighborhood of the origin (the digit 0) is eventually contained in the tile.

Note that for integral numeration systems there is neither a separation into real and imaginary parts nor is a sign needed.

For the triangular grid, tiles exists which do not contain a neighborhood of zero, we now give one example.

A complex numeration system corresponding to the tile Θ_{-1} of the curve `R7-1` on the

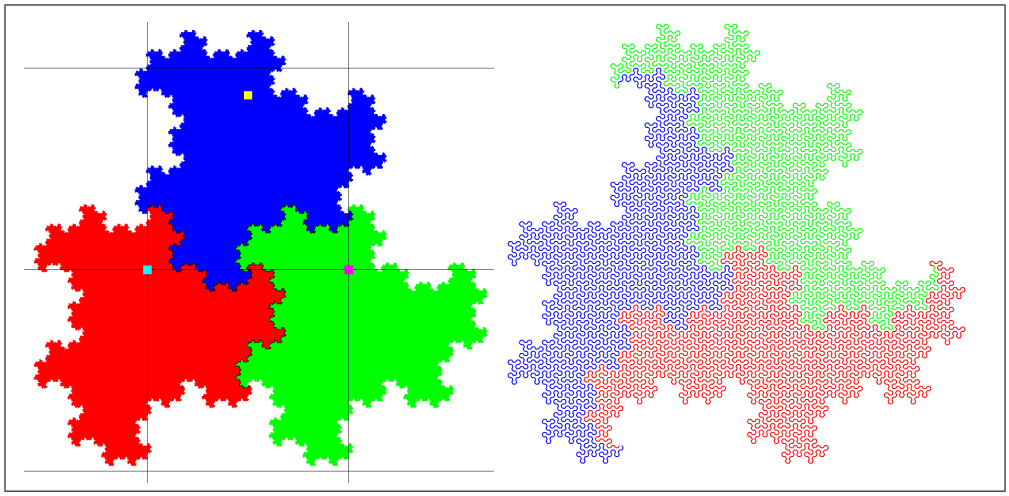

Figure 3.3-F: The fundamental region (left) for the complex numeration system with base $\sqrt{3}\,i$ and digit set $\{0, +1, \omega_6\}$ where $\omega_6 = \exp(+2\pi i/6) = (1 + i\sqrt{3})/2$ (the set is scaled up by the factor $\sqrt{3}$ to let the digits (small squares) lie in the three subsets) and the tile Θ_{+7} for the terdragon (right).

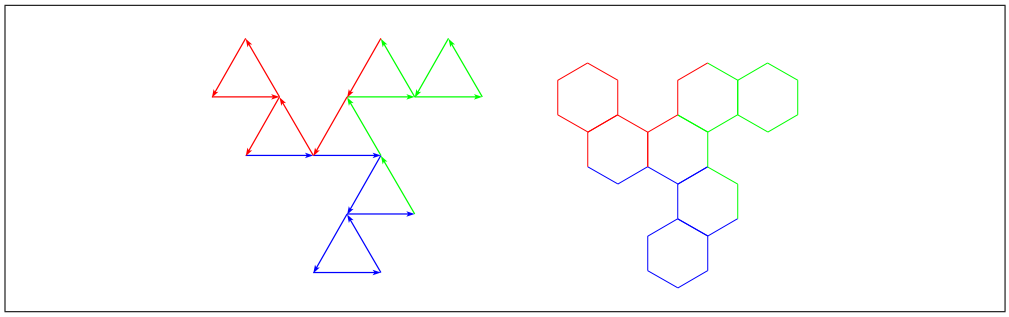

Figure 3.3-G: The tile Θ_{-1} of the curve `R7-1` (left), the same rendered as in Figure 3.1-H (right).

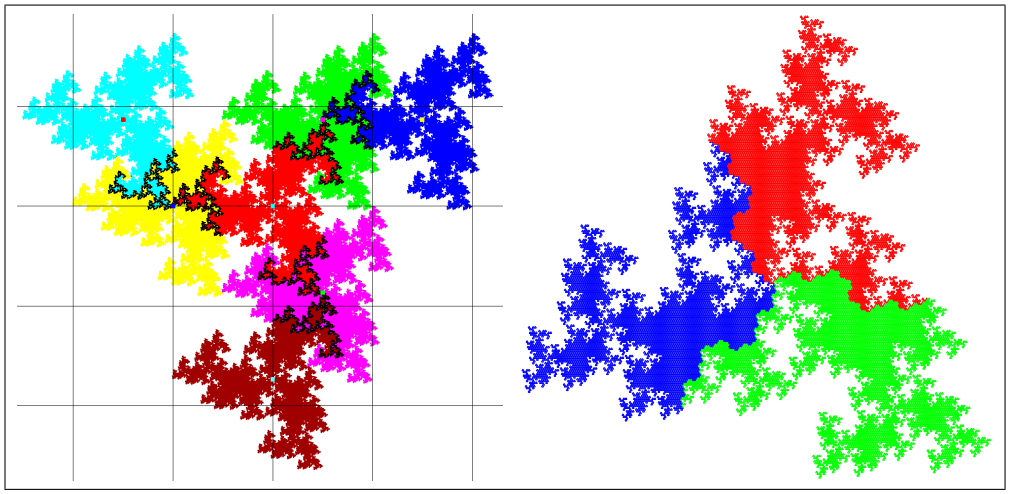

Figure 3.3-H: Fundamental region of a numeration system (left, scaled up by $\sqrt{7}$) and fifth iterate Θ_{-5} of the tile for the curve `R7-1` (right).

triangular grid can be obtained as follows. The $R = 7$ digits are points on the grid that are enclosed by the tile Θ_{-1}, Figure 3.3-G gives two renderings of the tile. We choose the center as digit 0, the other digits are ω_6 and $1 + \omega_6$ (upper right arm), -1 and $-1 + \omega_3$ (upper left arm), and $-\omega_3$ and $-\omega_3 - \omega_6$ (lower arm).

The choice of base $B = 2 - \omega_3 = \frac{1}{2}(5 - i\sqrt{3})$ ($\mathrm{abs}(B)^2 = 7$ and B lies on the triangular lattice) leads to the fundamental region as shown left in Figure 3.3-H, after rotation it matches the tile Θ_{-5} on the right. This tile (and fundamental region) does not contain a neighborhood of zero.

Tiles with 6-fold symmetry necessarily contain all digits adjacent to 0, so the corresponding numeration systems are always integral. For example, the (filled out) shape shown in Figure 3.1-C, after rotation such that 5 of the smaller tiles are on the real axis, corresponds to the complex base $1 + i\sqrt{12}$ and digit set $D = \{0, \pm 1, \pm 2, \pm\omega_6, \pm 2\omega_6, \pm\omega_3, \pm 2\omega_3\}$, an integral numeration system.

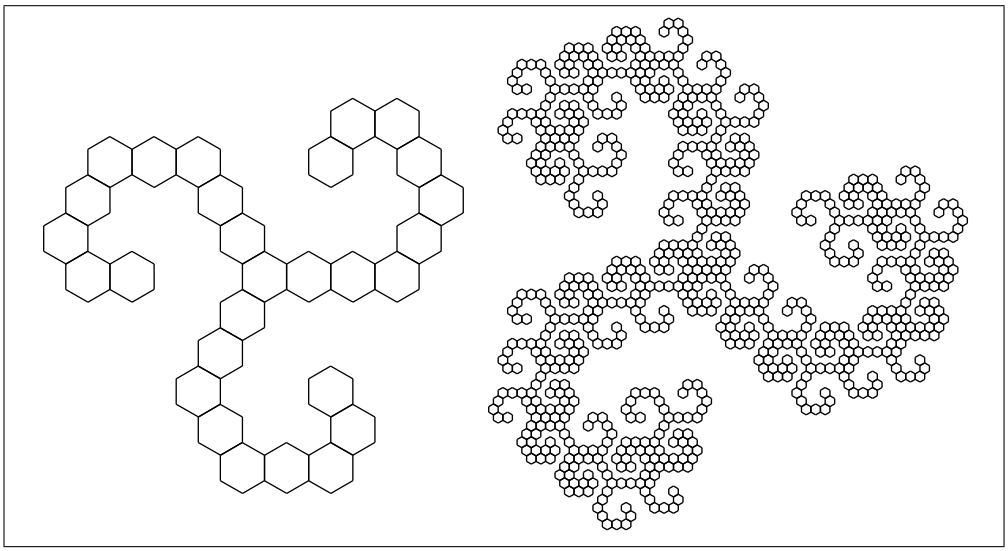

Figure 3.3-I: The tiles Θ_{+1} and Θ_{+2} of the curve `R28-10790` on the triangular grid.

The tiles Θ_{+1} and Θ_{+2} for a curve of order 28 on the triangular grid are shown in Figure 3.3-I. The central hexagon is not surrounded by others on Θ_{+1}, but in Θ_{+2} and all tiles Θ_{+k} for $k \geq 2$. The following *appears* to hold true. If there is a k_0 such that for all tiles Θ_k with $k \geq k_0$ the central polygon is surrounded by other polygons of the tile then $k_0 \leq 2$. Here we dropped the sign of the subscript in Θ_k, being arbitrary but fixed.

The subdivision of the tiles into smaller rotated copies of itself corresponds to an iterated function system (IFS) with R affine maps of the form $M_j(\vec{v}) = \vec{t}_j + \alpha\, T\, \vec{v}$ for $0 \leq j < R$ where α is a scalar with $\mathrm{abs}(\alpha)^2 = 1/R$, and T an orthogonal matrix. Neither α nor T do depend on j, the linear part $\alpha\, T$ of the map is a scaled rotation.

3.4 Curves and tiles on the tri-hexagonal grid

Curves on the tri-hexagonal grid (see Figure 3.4-A) must have non-zero turns between adjacent edges, otherwise two dead ends remain.

We use turns by 60° in the turtle graphics and represent 120° turns by `--` or `++` in the L-systems. For example, the curve of order 19 shown in Figure 3.4-B is given by the L-system `F` $\mapsto$ `F+F+F+F+F--F+F+F+F--F+F+F--F--F+F+F--F+F--F`.

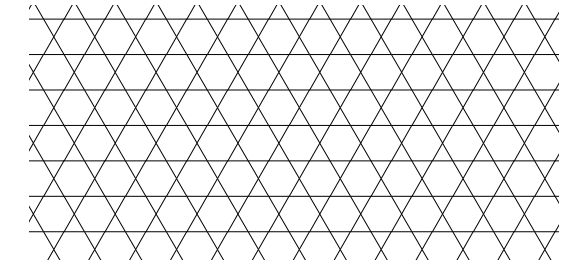

Figure 3.4-A: The tri-hexagonal grid.

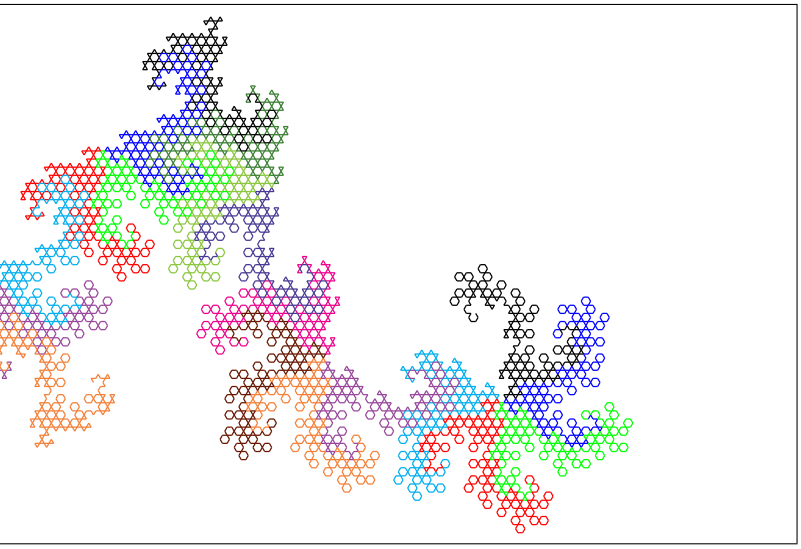

Figure 3.4-B: Third iterate of a curve of order 19 on the tri-hexagonal grid (`R19-1`). The coloration emphasizes the self-similarity.

```
F F+F+F+F+F--F+F+F--F--F+F+F+F--F  R13-1  #
F F+F+F+F+F--F+F--F--F+F+F+F--F  R13-2  #
F F+F+F+F--F+F+F+F--F--F+F--F+F  R13-3  #
F F+F--F+F--F--F+F+F+F--F+F+F+F  R13-4  # ## same = 3 Z T
```

Figure 3.4-C: Descriptions of the curves of order 13 on the tri-hexagonal grid.

Observe that all turns by 60° are in one direction (`+`) while all turns in the other direction are by 120° (`--`), see Figure 3.4-C for the file giving the L-system for the curves of order 13.

This is true for all curves on this grid: at any step turns `++` or `-` would give an edge ending in the middle of a hexagon. Obviously swapping the signs would give a similar curve whose shape is obtained by flipping the original curve, so we only allow turns `+` and `--`.

There is no curve with an L-system beginning and ending with `F--F` as neither tile would satisfy all conditions from section 2.1.

Curves on the tri-hexagonal grid exist only for orders $R = 6k + 1$ where R is of the form $x^2 + x\,y + x^2$, see sequence A260682 in [24].

The curves never have any non-trivial symmetry.

The two tiles Θ_+ and Θ_- for each curve have axioms `F+F+F+F+F+F` and `F--F--F`. The

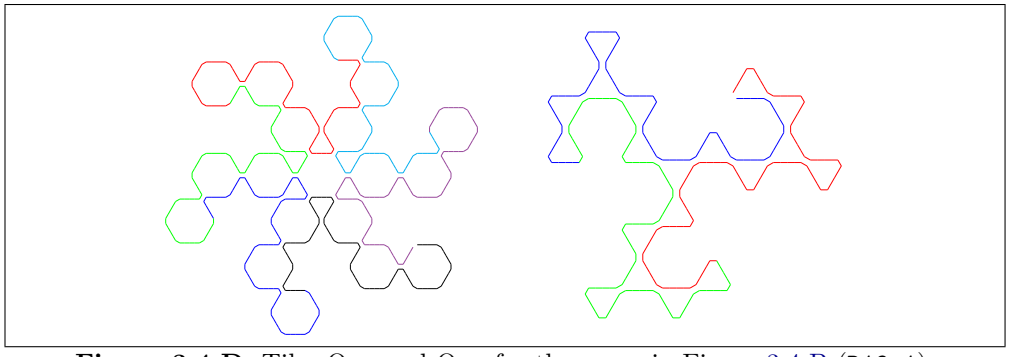

Figure 3.4-D: Tiles Θ_{+1} and Θ_{-1} for the curve in Figure 3.4-B (`R19-1`).

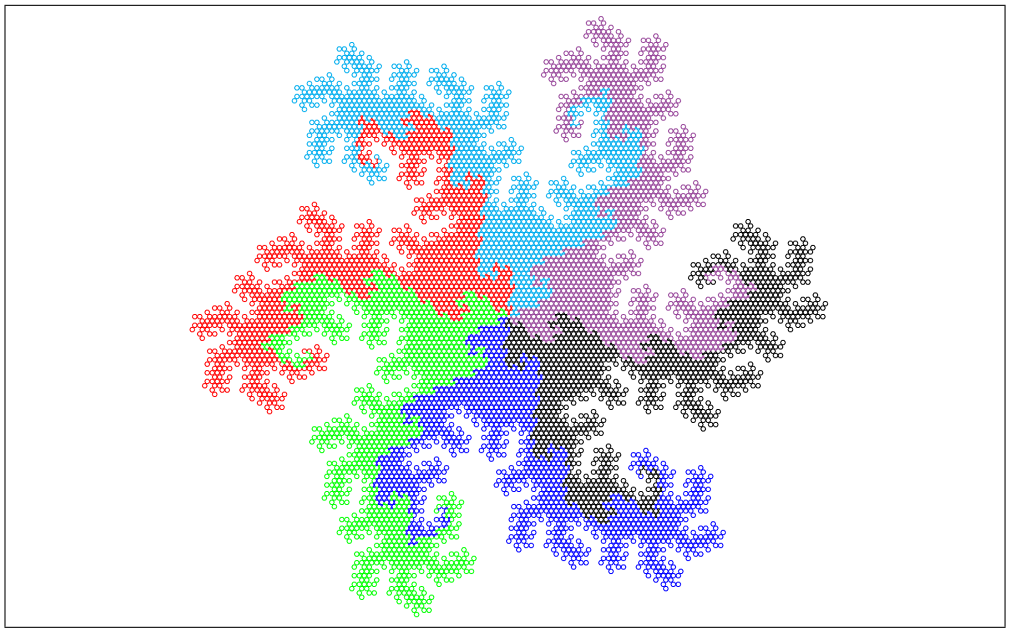

Figure 3.4-E: Tile Θ_{+3} for the curve in Figure 3.4-B (`R19-1`).

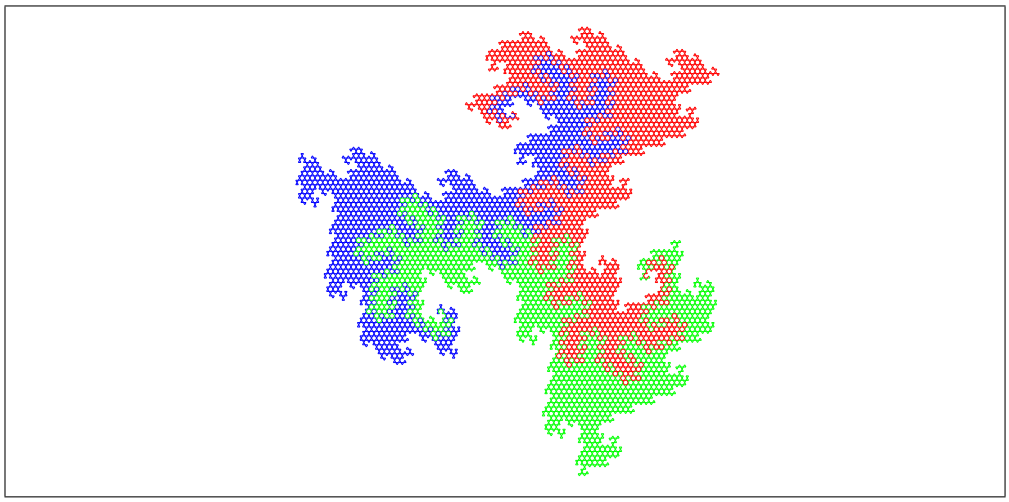

Figure 3.4-F: Tile Θ_{-3} for the curve in Figure 3.4-B (`R19-1`).

tiles Θ_{+1} and Θ_{-1} for the curve mentioned are shown in Figure 3.4-D. All tiles Θ_{+j} are centered at a hexagon and have 6-fold symmetry while all tiles Θ_{-j} are centered at a triangle and have 3-fold symmetry. Figures 3.4-E and 3.4-F show the third iterates.

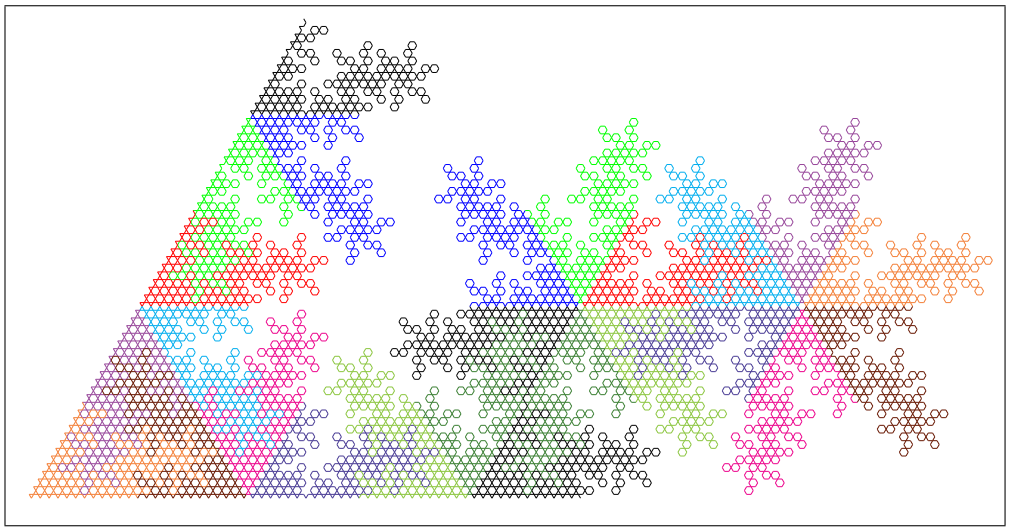

Figure 3.4-G: Self-similarity of a curve of order 25 (`R25-11`).

The tile Θ_+ again tiles the grid with translated copies. The tile Θ_-, however, tiles in the way the triangular grid is tiled by triangles, half of them are rotated by 120°. This can be seen in Figure 3.4-G for a curve of order 25: the right part shows how the tiles Θ_+ match and on the left the triangular form of the tiles Θ_- is obvious. Curves whose tile Θ_- has the shape of an equilateral triangle exist only if the order is a perfect square.

The tiles Θ_+ always give integral complex numeration systems, their shapes $\Theta_{+\infty}$ are the same as the shapes of tiles with 6-fold symmetry (in the limit) appearing for curves on the triangular grid of the same order. The tiles Θ_- on the tri-hexagonal grid are never equal to any shape of a tile on the triangular grid.

4 Plane-filling curves on all uniform grids

The curves corresponding to simple L-systems are edge-covering (traverse each edge of the underlying grid once) and necessarily traverse each point more than once (hence are not point-covering). Here we give methods to convert these curves into point-covering curves on all uniform grids and to edge-covering curves on two uniform grids.

We specify the type of grid by the symbols for the vertex type for the corresponding uniform edge to edge tilings of regular polygons as in [12, p. 63], also see [26] and [27]: the symbol is the (least cyclic shift of the) list of numbers of edges of the polygons around any of the points.

The symbols for the grids so far seen are (3.3.3.3.3.3) = (3^6) for the triangular grid, (4.4.4.4) = (4^4) for the square grid, (6.6.6) = (6^3) for the hexagonal grid, and (3.6.3.6) for the tri-hexagonal grid. We will encounter the grids (3.3.3.4.4) = $(3^3.4^2)$, (4.8.8) = (4.8^2), and (3.3.4.3.4) = $(3^2.4.3.4)$, shown in Figure 4.0-A, and (4.6.12), (3.3.3.3.6) = $(3^4.6)$ (there are two enantiomers of this grid), and (3.12.12) = (3.12^2), shown in Figure 4.0-B. The remaining grid (3.4.6.4) is shown in section 4.2.1.

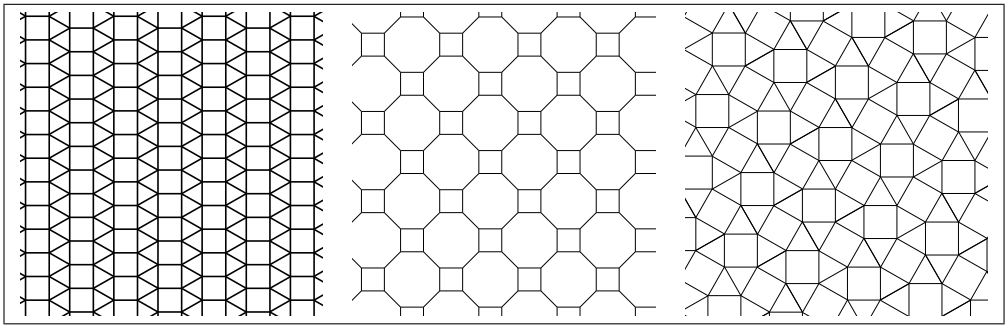

Figure 4.0-A: The grids (3.3.3.4.4) = ($3^3.4^2$), (4.8.8), and (3.3.4.3.4).

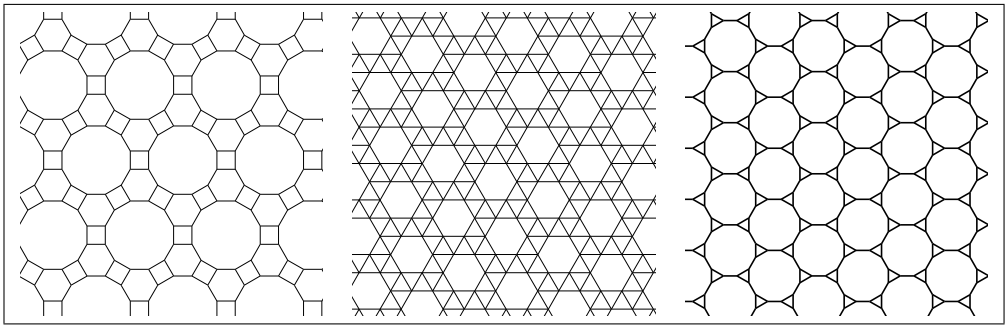

Figure 4.0-B: The grids (4.6.12), (3.3.3.3.6) = ($3^4.6$), and (3.12.12) = (3.12^2).

We call a curve (G)-PC if it is point-covering on the grid (G). For example, both the Peano and the Hilbert curve (Figure 1.3-A) are (4^4)-PC and Gosper's flowsnake (Figure 1.3-B) is (3^6)-PC (left rendering), but (6^3)-PC when rendered as in the right. Similarly, we use (G)-EC to denote edge-covering on the grid (G).

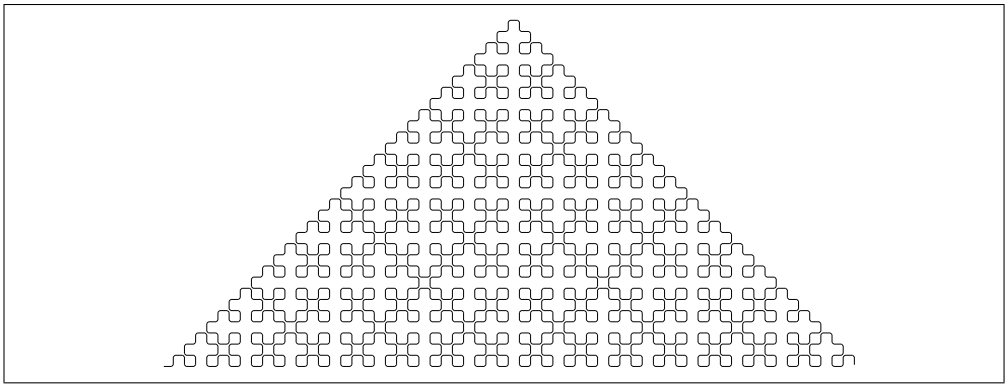

Figure 4.0-C: A plane-filling curve on the square grid that is neither EC nor PC: all points are traversed, but some twice and not all edges are traversed.

The edge-covering and point-covering curves are just two corner cases of the plane-filling curves on a grid, Figure 4.0-C shows a curve that is in neither class.

4.1 Conversions to point-covering curves

We organize the following sections by the grids whose edge-covering curves are used to compute point-covering curves on some uniform grids.

4.1.1 Triangular grid: curves for (6^3), (3.6.3.6), $(3^3.4^2)$, (3^6), (3.12.12), (3.4.6.4), (4.6.12), $(3^4.6)$, and (4^4)

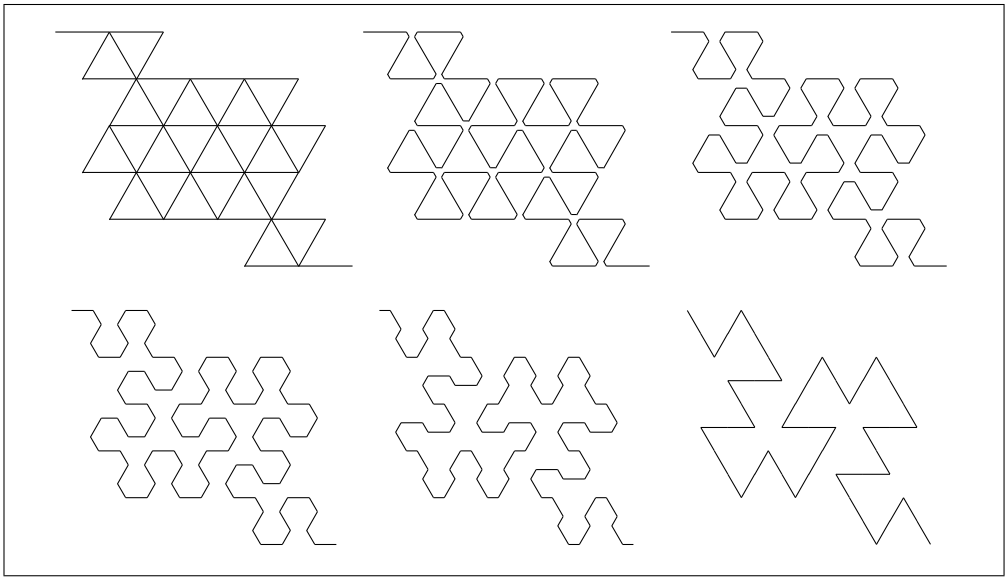

Figure 4.1-A: Renderings of the curve `R7-5` with map `F` $\mapsto$ `F+F-F-F+F+F-F` on the triangular grid for rounding parameter $e \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$.

The constructions given here only work for curves of two types. We call curves where all turns are by a non-zero angle *wiggly*, and curves with the same numbers of `+`, `-`, and `0` in the production of `F` *balanced*. Wiggly curves exist only for odd orders, see sequence A266836 in [24]. The curve `R7-5` with map `F` $\mapsto$ `F+F-F-F+F+F-F` whose motif is shown on the right of 2.2-A on page 11 is a wiggly curve. Unless otherwise specified our methods only work for wiggly curves. Balanced curves only exits for orders of the form $3k+1$, sequence A202822 in [24]. An example is the curve `R7-1` with map `F` $\mapsto$ `F0F+F0F-F-F+F`, it is shown in Figure 1.2-D on page 5. In balanced curves every point is traversed once by two edges with a non-turn between them.

wiggly
balanced

We have sometimes used rounded corners to make the curves visually apparent: at each turn a fraction of $e > 0$ of both edges are skipped and a straight line is used to connect the two new end points. Figure 4.1-A shows renderings of a curve of order 7 on the triangular grid for $e \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$.

The rendering in the lower left ($e = 1/3$) and lower right ($e = 1/2$) are respectively a (6^3)-PC curve and a (3.6.3.6)-PC curve. The turning points of the rounded corners are indicated in Figure 4.1-B.

The words for these curves can be computed using a post-processing step on the final iterate of the L-system. For the (6^3)-PC curve, replace all `+F` by `+F+F`, all `-F` by `-F-F`, and use turns by 60°. For the (3.6.3.6)-PC curve, drop all `F`, replace all `+` by `+F+`, all `-` by `-F-`, and use turns by 60°. The corresponding curves with the ends of all edges marked are shown in Figure 4.1-C.

Curves that are (3^6)-PC can be obtained by post-processing as well: drop all `F`, replace all `+` by `++F`, all `-` by `-F-`, and use turns by 60°. The third iterate of the resulting curve is shown in Figure 4.1-D. We can generate (3^6)-PC curves from balanced curves by dropping all `F`, replacing all `+` by `+F+`, all `-` by `-F-`, and all `0` by `F0F`. Turns by 60° are used. The curve from `R7-1` is shown in Figure 4.1-E.

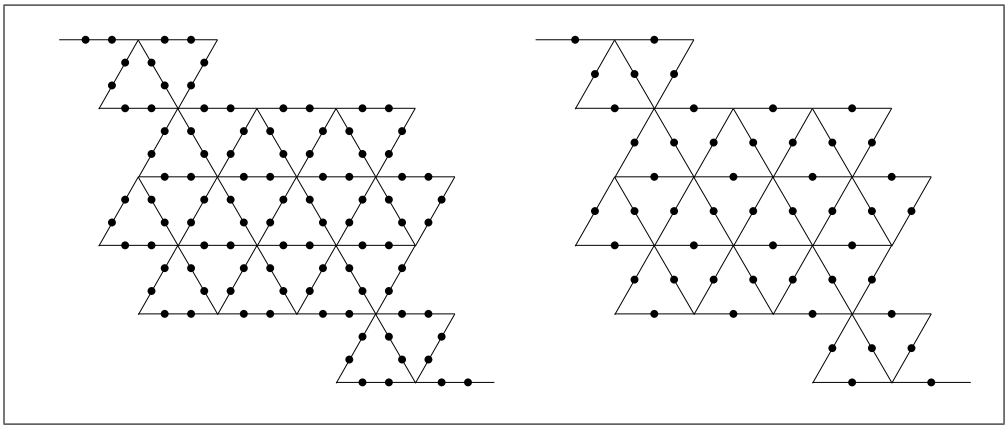

Figure 4.1-B: The turning points with rounded corners for $e = 1/3$ (left) and $e = 1/2$ (right) respectively give the hexagonal and tri-hexagonal grid.

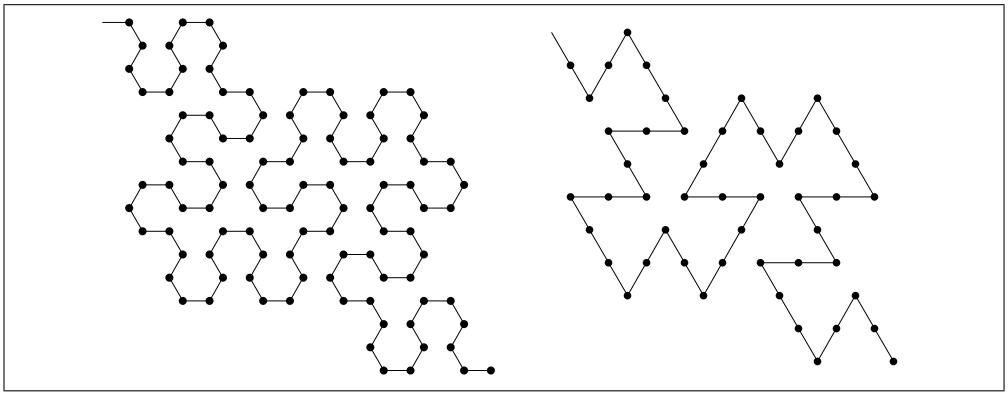

Figure 4.1-C: The curves obtained by post-processing corresponding to $e = 1/3$ (left), a (6^3)-PC curve, and $e = 1/2$, a (3.6.3.6)-PC curve (right).

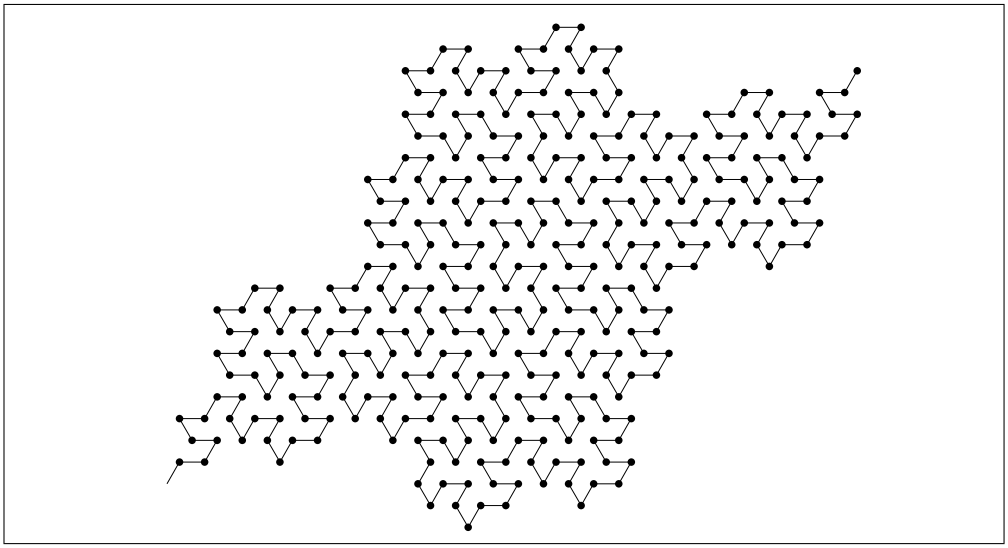

Figure 4.1-D: The (3^6)-PC curve from the third iterate of `R7-5`.

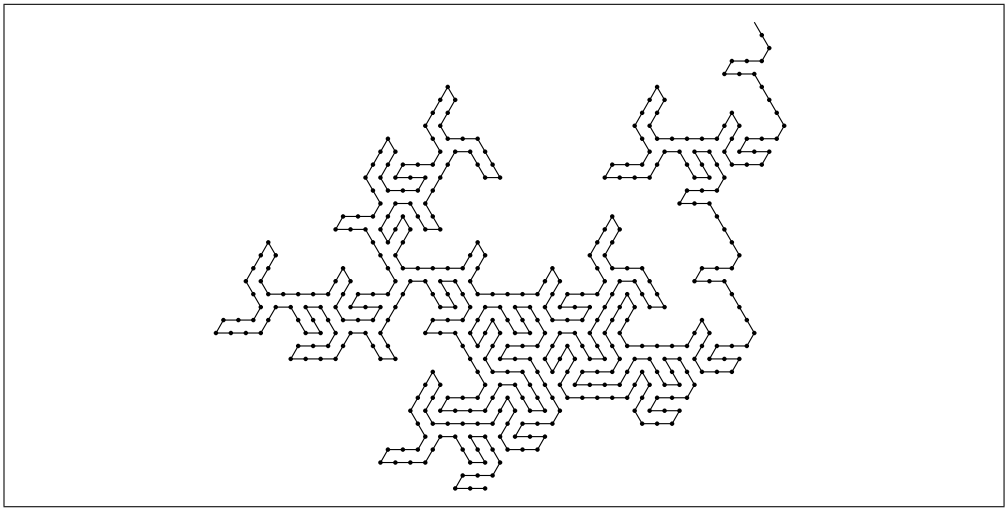

Figure 4.1-E: The (3^6)-PC curve from the third iterate of the balanced curve `R7-1`.

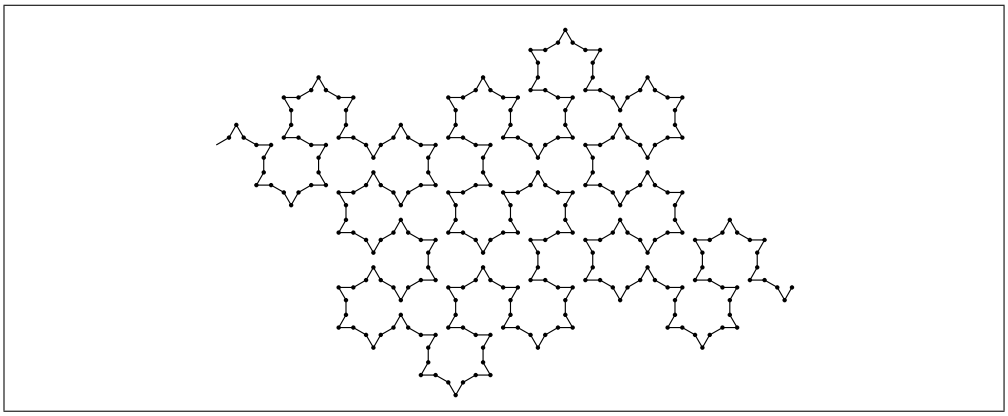

Figure 4.1-F: The (3.12.12)-PC curve from the second iterate of `R7-5`.

Curves that are (3.12.12)-PC like the one shown in Figure 4.1-F can be obtained as follows. Drop the initial F, then replace all `+F` by `X` and all `-F` by `Y`, then replace all `X` by `F-F++++F-F-F++++F-` and all `Y` by `F+F----F+F+F----F+`, use turns by 30°.

Two (3.4.6.4)-PC curves are shown in Figure 4.1-G. For the curve on the left, drop all `F`, replace all `+` by `p` and all `-` by `m`, then replace all `p` by `++F+++F----F+++F+++F----F+` and all `m` by `--F---F++++F---F---F++++F-`, use turns by 30°. For the curve on the right, drop all `F`, replace all `+` by `p` and all `-` by `m`, then replace all `p` by `+++F----F+++F+++F----F+++F` and all `m` by `+++F----F-F-F----F+++F`, again use turns by 30°.

For balanced curves, (3.4.6.4)-PC curves are obtained by replacing all `+` by `p` and all `-` by `m`, then all `Fp` by `+F++F+`, all `Fm` by `--F--`, all `F0` by `+F-F-F+`, use turns by 30°. The curve from `R7-1` is shown in Figure 4.1-H.

For curves that are (4.6.12)-PC (Figure 4.1-I) drop all `F`, replace all `+` by `p` and all `-` by `m`, then replace all `p` by `+F+F+F+F` and all `m` by `+F---F---F+F`. Use turns by 30°.

For balanced curves, (4.6.12)-PC curves are obtained by replacing all `F+` by `F+F+F+F+`, all `F-` by `F--F--`, and all `F0` by `F+F+F--F--F+F+`, again using turns by 30°. See figure 4.1-J for the curve from the third iterate of `R71-1`.

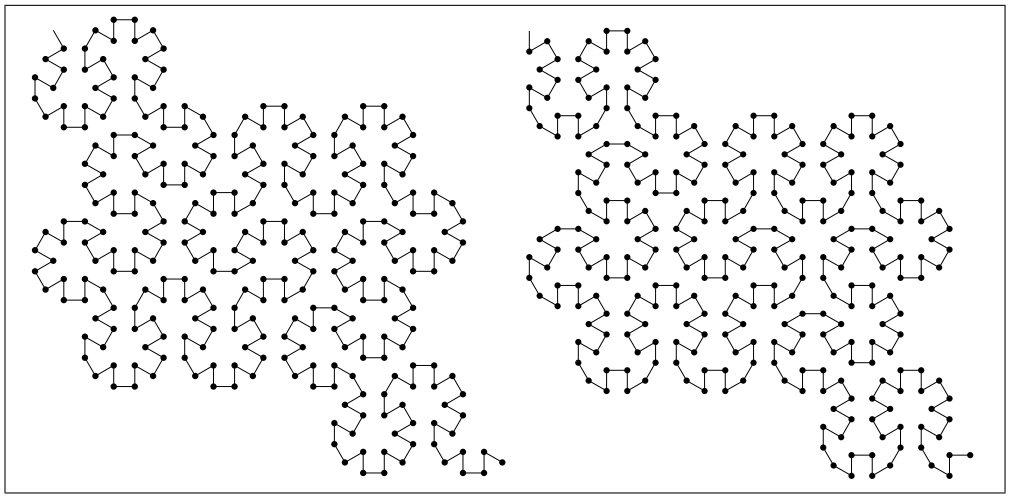

Figure 4.1-G: Two (3.4.6.4)-PC curves from the second iterate of `R7-5`.

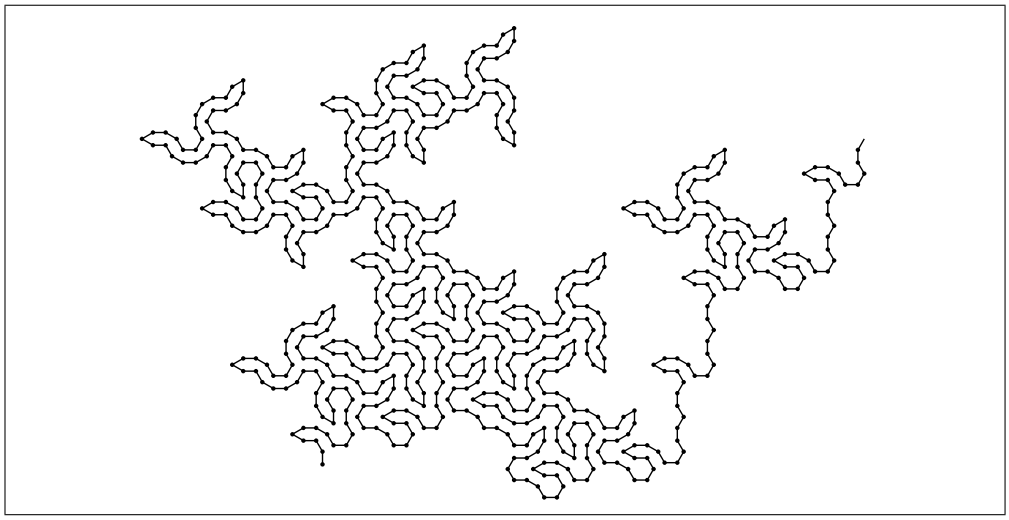

Figure 4.1-H: (3.4.6.4)-PC curve from the third iterate of the balanced curve `R7-1`.

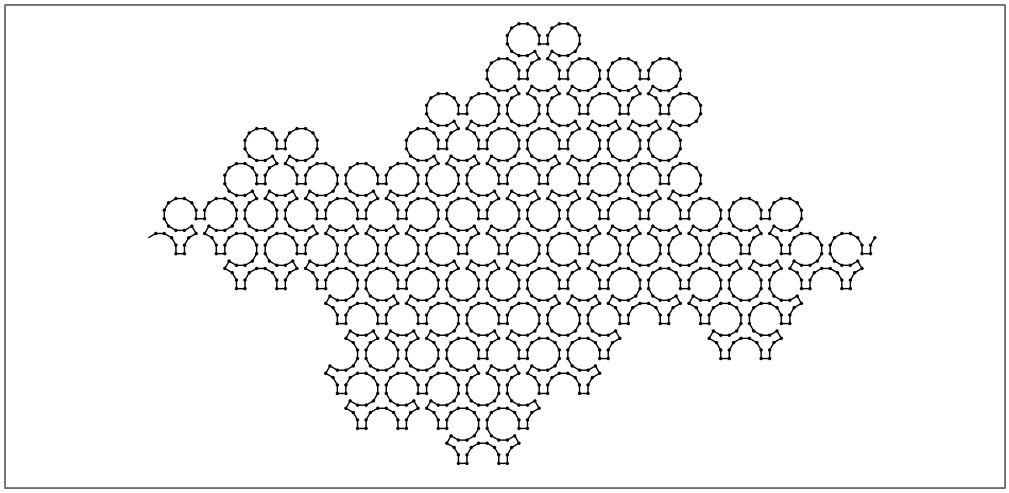

Figure 4.1-I: The (4.6.12)-PC curve from the third iterate of `R7-5`.

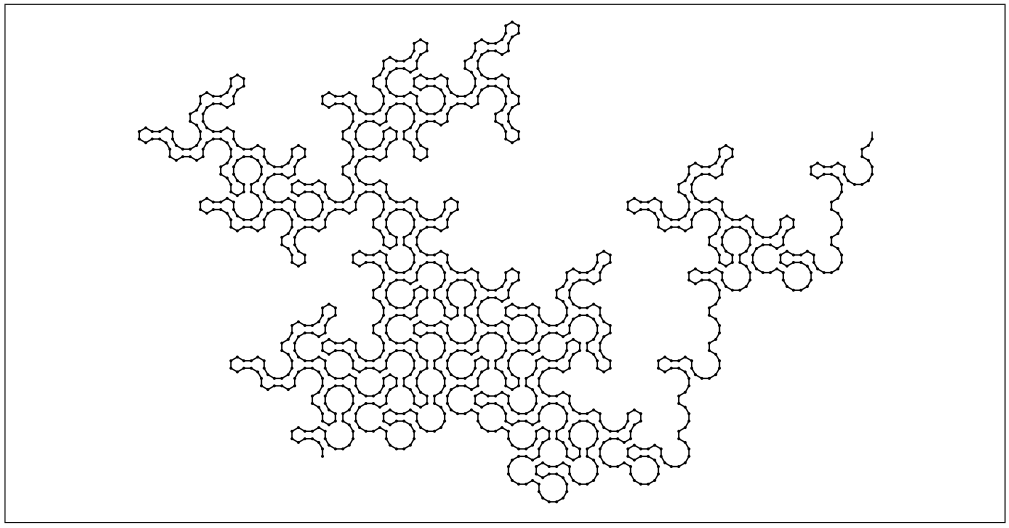

Figure 4.1-J: (4.6.12)-PC curve from the third iterate of the balanced curve `R7-1`.

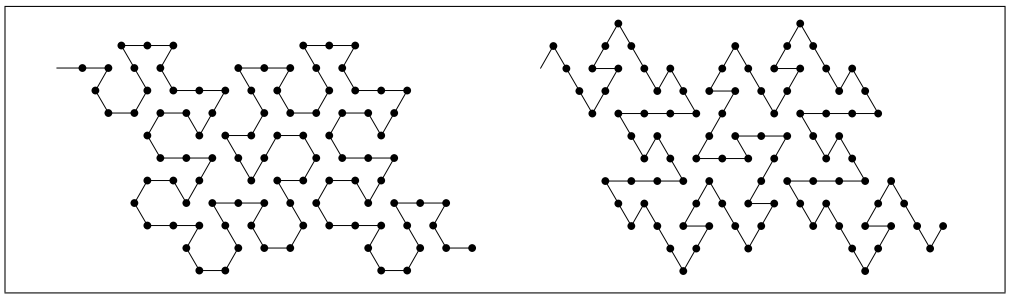

Figure 4.1-K: Two (3^4.6)-PC curves from the second iterate of `R7-5`.

Curves that are (3^4.6)-PC are obtained by replacing all `+` by `F++`, all `-` by `-F-`, and using turns by 60°. Figure 4.1-K (left) shows the curve from the second iterate. For the curve for the other enantiomer of the grid, change the first replacement to `++F`. Replacing all `+` by `+T+` and all `-` by `-T-`, then all `F` by `-T+`, drawing edges for `T` using turns by 60° gives the curve shown on the right of Figure 4.1-K. Change the second replacement to `+T-` for the other enantiomer.

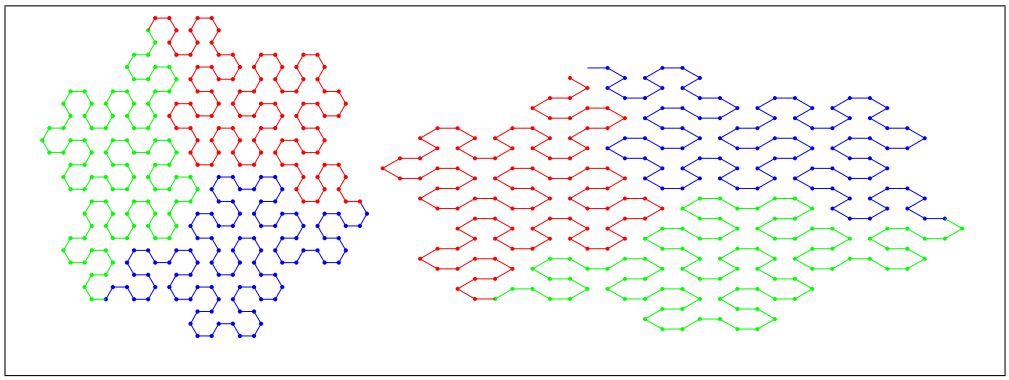

Figure 4.1-L: The second iterate of the tile of `R7-5` (Θ_{+2}, left) and the ($3^3.4^2$)-PC curve derived from it by redirecting non-horizontal edges (right).

Figure 4.1-L shows the $(3^3.4^2)$-PC curve obtained from the curve at the left by adjusting the directions of the edges that are not horizontal.

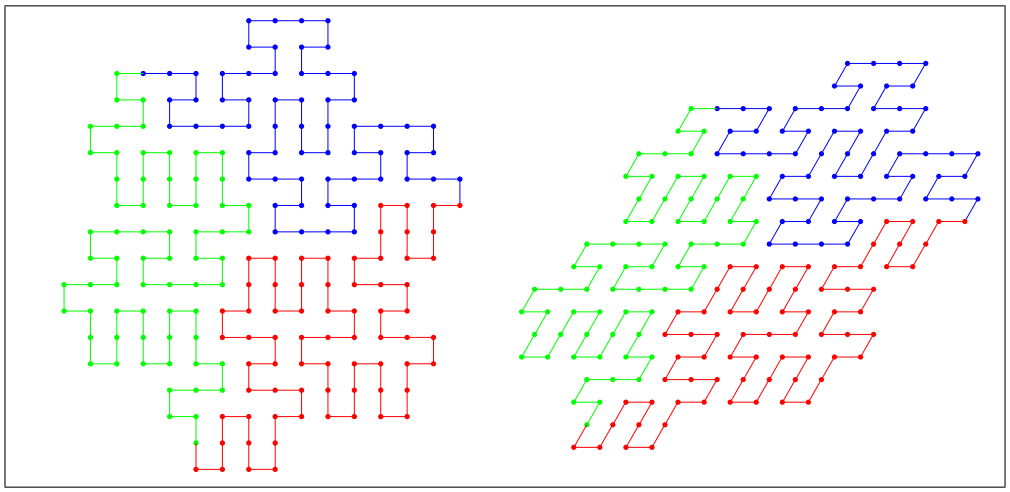

Figure 4.1-M: Rendering of the tile Θ_{+2} as (4^4)-PC (left) and (3^6)-PC curves (right).

(4^4)-PC curves can be obtained by changing all non-horizontal edges in a (6^3)-PC curve, that is, mapping the directions 1, 2, ..., 6 shown at the right of Figure 1.4-A on page 8 respectively to directions 1, 2, 2, 3, 4, 4 on the square grid. The resulting curves are stretched in the vertical direction. Dropping the directions 3 and 5 first gives less distorted curves, as shown at the left in Figure 4.1-M. Turning all vertical edges in the (4^4)-PC curve gives the (3^6)-PC curve at the right.

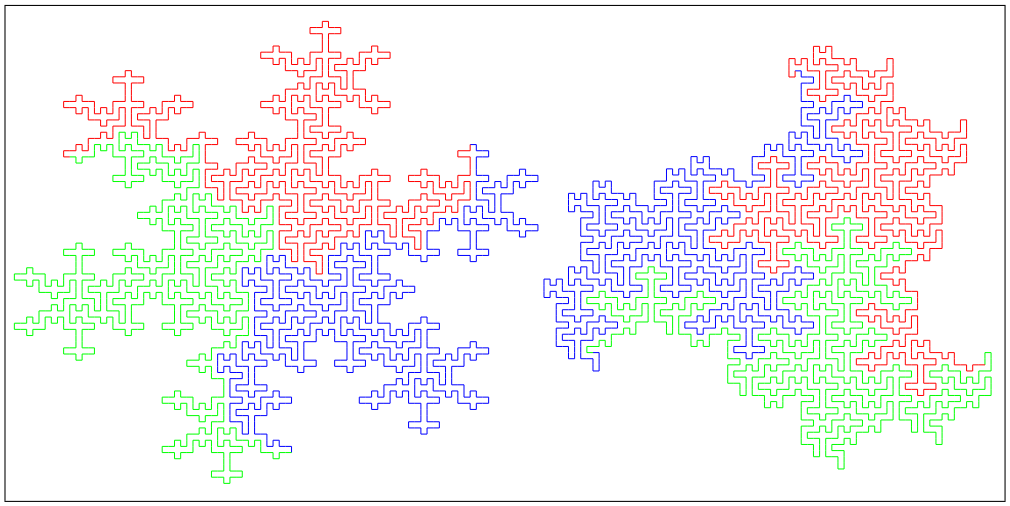

Figure 4.1-N: Point-covering curves on the square grid from the tiles Θ_{+2} (left) and Θ_{-2} (right) of the balanced curve `R13-15` on the triangular grid.

Balanced curves can be rendered as (4^4)-PC curves such as shown in Figure 4.1-N by first converting them to (3.6.3.6)-EC curves as described in section 4.2.2 on page 43 and then applying the conversion used for Figure 4.1-X on page 41.

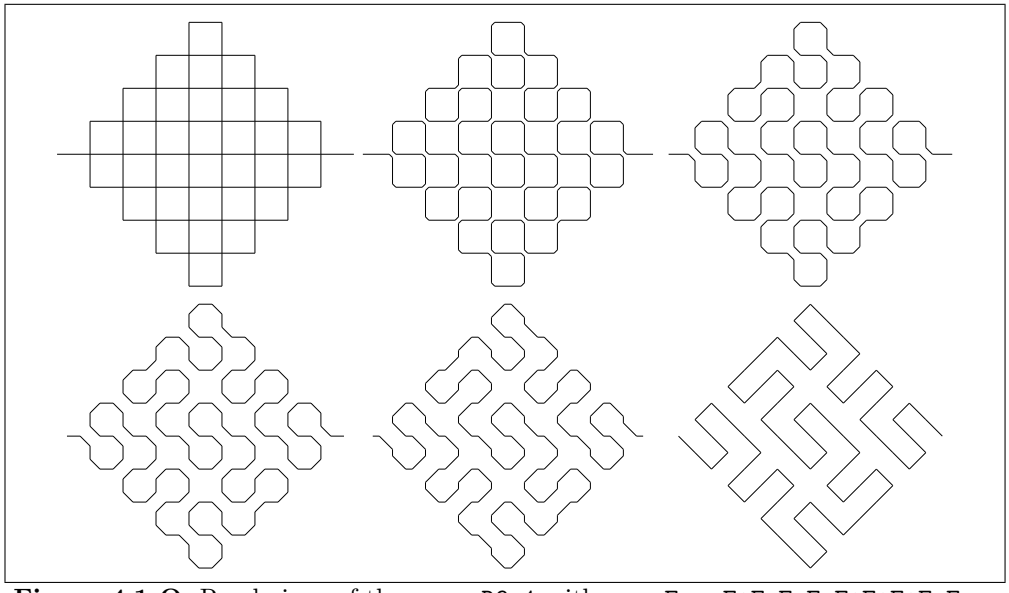

Figure 4.1-O: Renderings of the curve `R9-1` with map `F` $\mapsto$ `F+F-F-F-F+F+F+F-F` on the square grid for rounding parameter $e \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$.

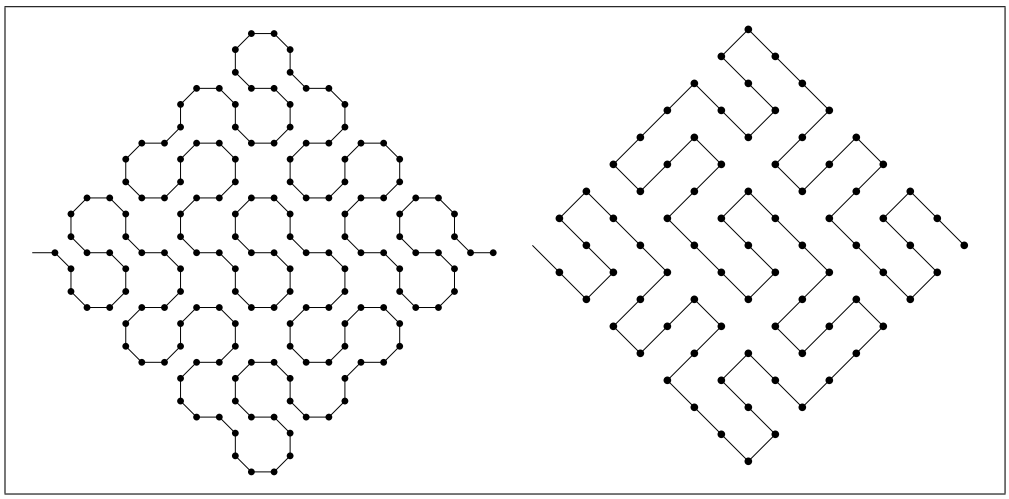

Figure 4.1-P: Points traversed by the curves corresponding to $e = 1/(2+\sqrt{2}) \approx 0.292893\ldots$, a (4.8.8)-PC curve (left), and $e = 1/2$, a (4^4)-PC curve (right).

4.1.2 Square grid: curves for (4.8.8), (4^4), (3.3.4.3.4), and (3^6)

Figure 4.1-O shows renderings of the curve `R9-1` with map `F` $\mapsto$ `F+F-F-F-F+F+F+F-F` for rounding parameter $e \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$. This curve is sometimes referred to as Peano curve, see [23, Chapter 3, pp. 31ff].

The curve for $e = 1/3$ is *almost* a (4.8.8)-PC curve: for all edges to be of equal length, the remaining edges must have the length of the edges shortening the corners, so we must have $1 - 2\,e = \sqrt{2}\,e$, which gives $e = 1/(2+\sqrt{2}) \approx 0.292893\ldots$

For $e = 1/2$ (lower right) we obtain a (4^4)-PC curve, the points are the mid-points of the edges of the original curve.

The curves can again be obtained by post-processing steps. For $e = 1/(2+\sqrt{2}) \approx 1/3$

replace all `+F` by `+F+F`, all `-F` by `-F-F`, and use turns by 45°. For $e = 1/2$ drop all `F`, replace all `+` by `+F+`, all `-` by `-F-`, and use turns by 90°. Both curves are shown in Figure 4.1-O.

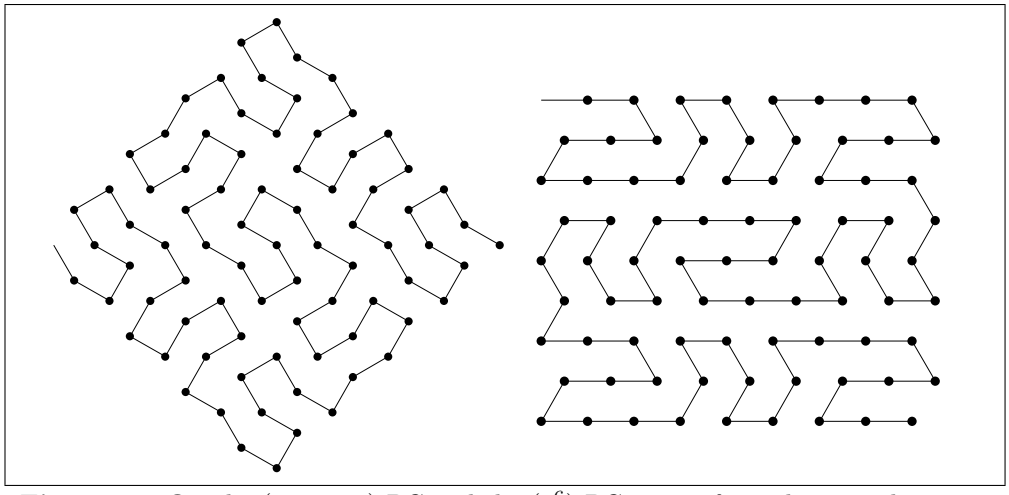

Figure 4.1-Q: The (3.3.4.3.4)-PC and the (3^6)-PC curves from the second iterate.

For the (3.3.4.3.4)-PC curve shown on the left in Figure 4.1-Q, drop all `F`, replace all `+` by `++F+`, all `-` by `--F-`, and use turns by 30°.

The (3^6)-PC curve shown on the right in Figure 4.1-Q results from rendering the (4^4)-PC curve (right of Figure 4.1-P) such that one set of parallel edges (horizontal in the image) is kept and the other (vertical) edges are alternatingly turned by ±30° against their original direction.

4.1.3 Tri-hexagonal grid: curves for (4.6.12), (3.4.6.4), $(3^4.6)$, and (4^4)

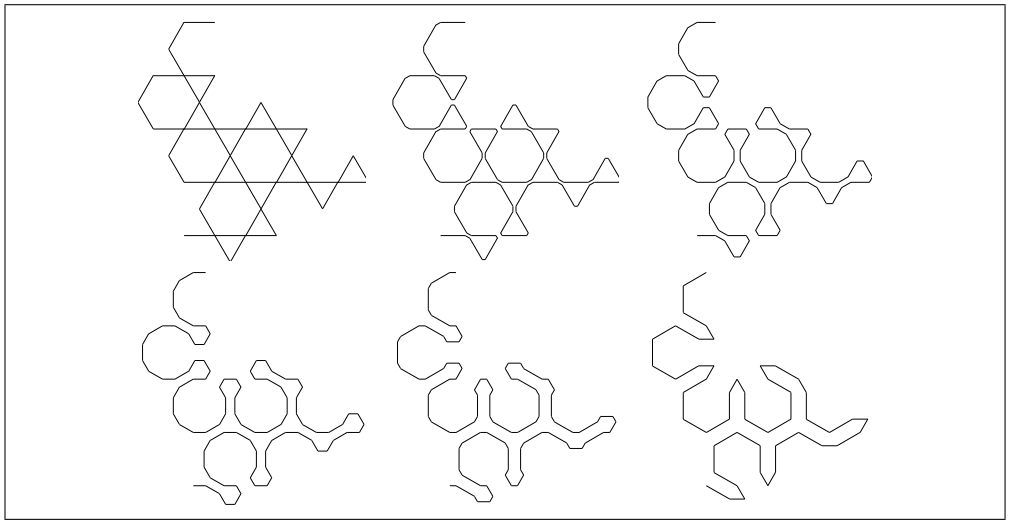

Figure 4.1-R: Renderings of the second iterate of Ventrella's curve (`R7-1`) for rounding parameter $e \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$.

For our examples we use Ventrella's curve (`R7-1`) with the map `F` ↦ `F+F--F--F+F+F+F` on the tri-hexagonal grid already seen in Figure 1.1-D on page 3. The renderings

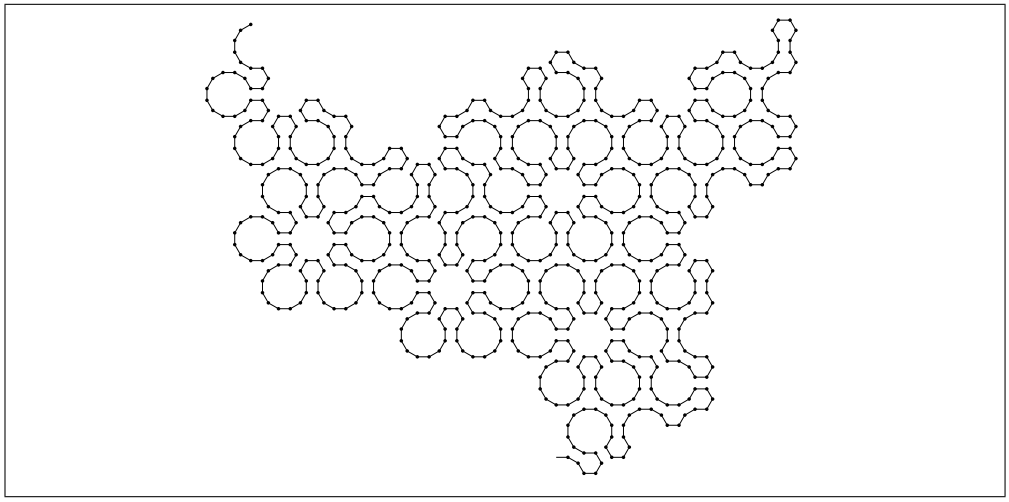

Figure 4.1-S: A (4.6.12)-PC curve from the third iterate.

of the second iterate for the rounding parameter $e \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$ are shown in Figure 4.1-R.

The rendering for $e = 1/3$ (lower left in Figure 4.1-R) shows a distorted (4.6.12)-PC curve. For the (undistorted) curve drop all `F`, replace all `+` by `F+F+` and all `--` by `F--F--`, use turns by 30°. Figure 4.1-S shows the curve obtained from the third iterate.

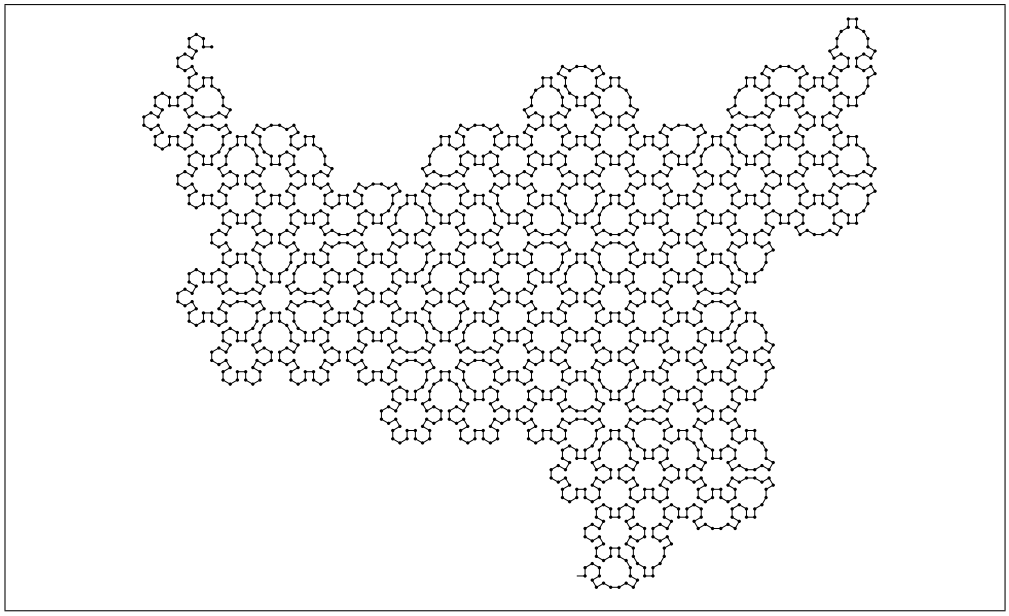

Figure 4.1-T: Another (4.6.12)-PC curve from the third iterate.

A different (4.6.12)-PC curve is shown in Figure 4.1-T, for the curve replace all `+` by `p` and all `--` by `m`, then replace all `p` by `---F++F++F++F++F---` and all `m` by `---F++F-F-F++F---`, use turns by 30°.

The rendering $e = 1/2$ (lower right in Figure 4.1-R) similarly suggests the following construction of a (3.4.6.4)-PC curve, shown in Figure 4.1-U. Drop all `F`, replace all `+` by `+F+` and all `--` by `--F--`, and use turns by 30°.

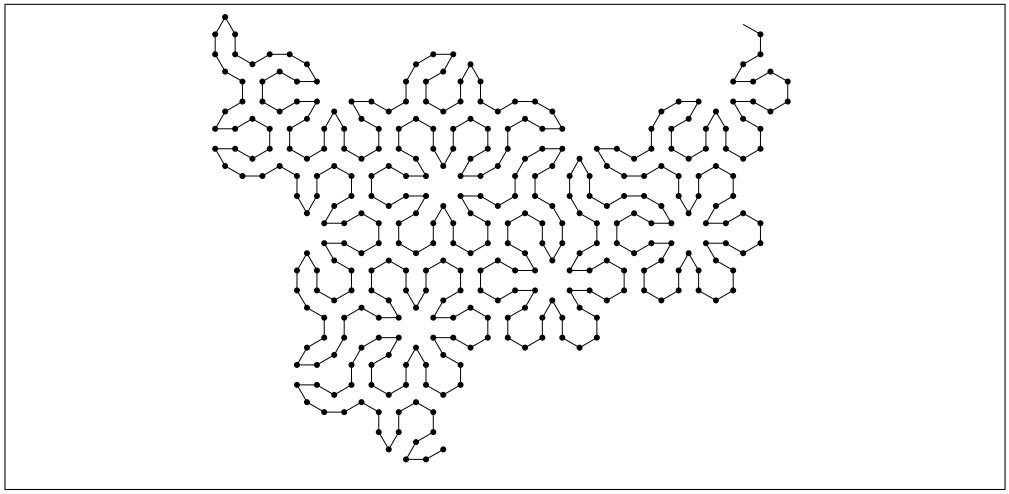

Figure 4.1-U: The (3.4.6.4)-PC curve from the third iterate.

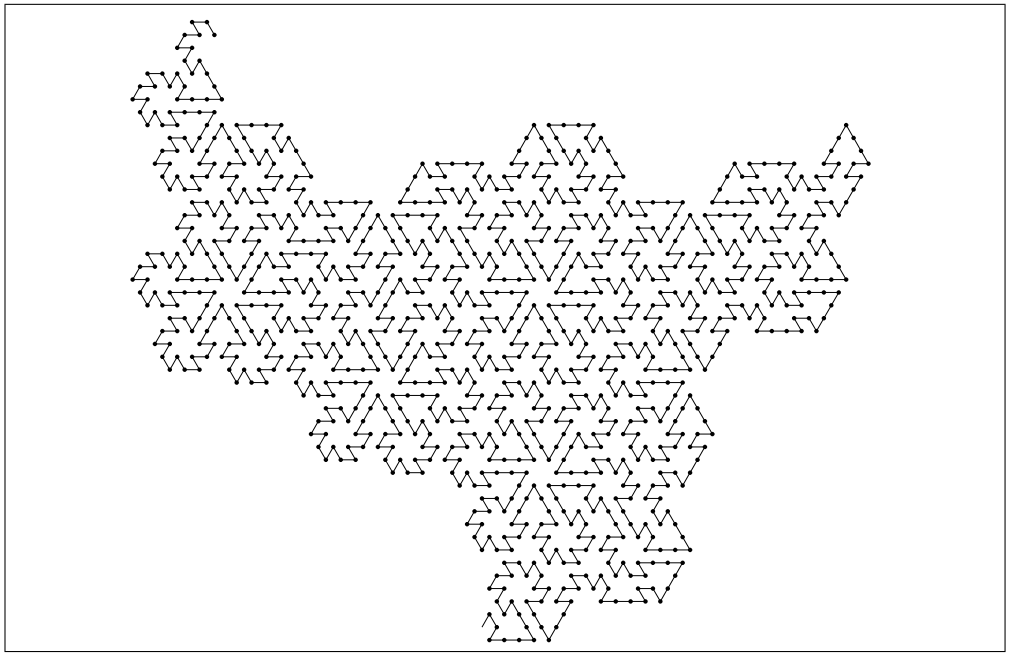

Figure 4.1-V: A $(3^4.6)$-PC curve from the third iterate.

For the $(3^4.6)$-PC curve shown in Figure 4.1-V, drop all `F`, replace all `+` by `p` and all `--` by `m`, then replace all `p` by `--F++++F++F--` and all `m` by `--FFF--`, use turns by 30°.

For the $(3^4.6)$-PC curve shown in Figure 4.1-W, drop all `F`, replace all `+` by `+F` and all `--` by `-F-`, and use turns by 60°. Using the replacements `-F` for `+` and `+F+` for `-` gives the other enantiomer.

In the (4^4)-PC curve shown in Figure 4.1-X all hexagons in the tile shown on the left are replaced by crosses whose horizontal bars are longer than the vertical ones (right). For the conversion we keep track of the direction modulo 3 (not 6!) which shall be held in a variable d. In the following method d has to be incremented or decremented modulo 3 with every `+` or `-` in the input stream. All letters `F` are dropped. When a `+` is read, print `++FF---F+++` if $d = 0$, `F+++F---F++` if $d = 1$, and `--F+++F+` if $d = 2$.

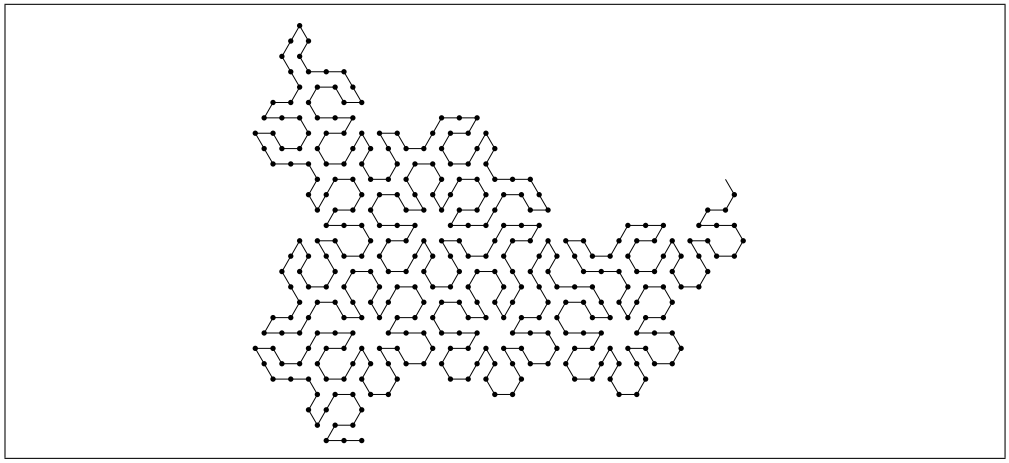

Figure 4.1-W: Another $(3^4.6)$-PC curve from the third iterate.

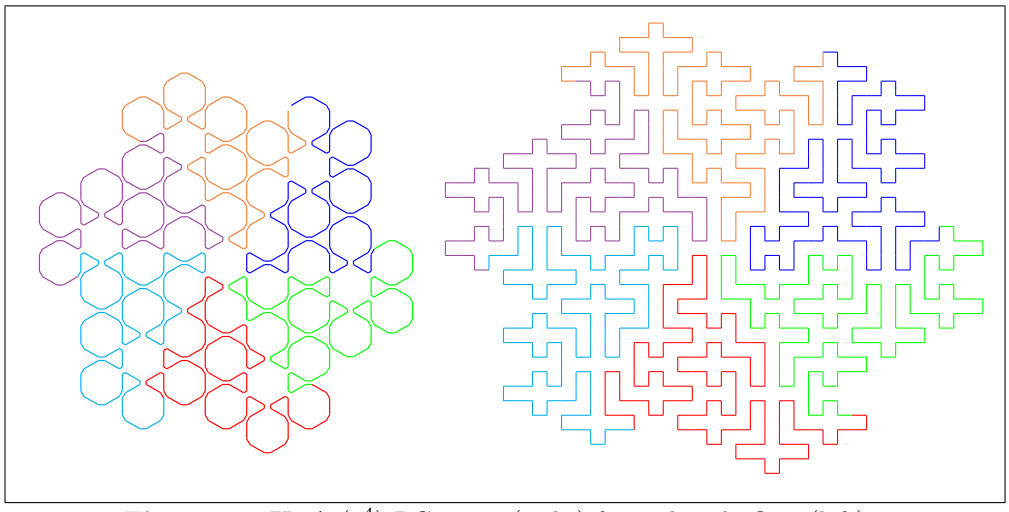

Figure 4.1-X: A (4^4)-PC curve (right) from the tile Θ_{+2} (left).

When a `-` is read, print `-F---F---F+++` if $d = 0$, `---FF---F++` if $d = 1$, and `-----FF+` if $d = 2$. Use turns by 30°, the resulting turns are all by 0 or ±90°.

The two (4^4)-PC curves shown in Figure 4.1-Y are renderings with rounding parameter $e = 1/2$ of the (4^4)-EC curves computed as in section 4.2.3 on page 46.

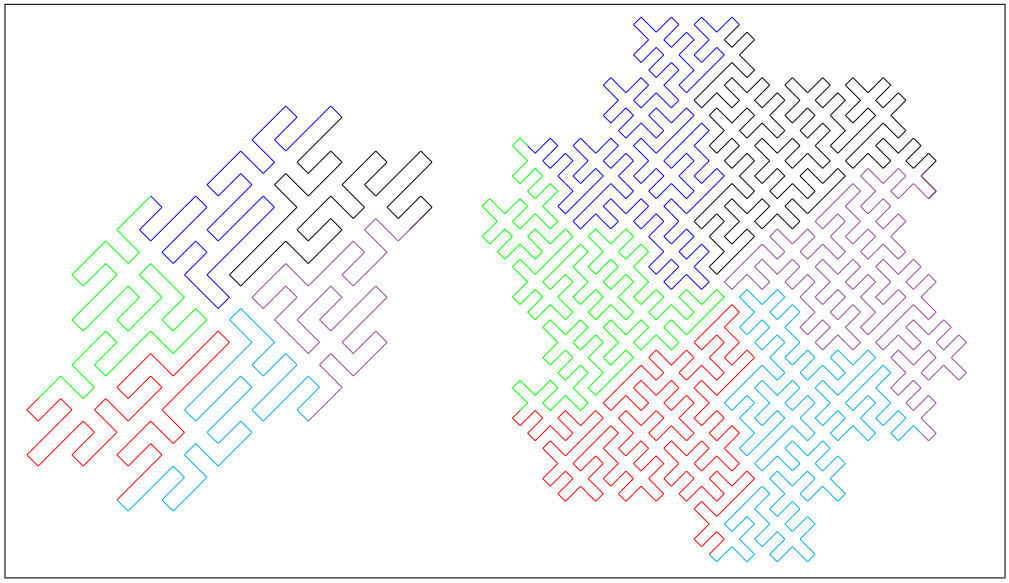

Figure 4.1-Y: Two (4^4)-PC curves from the tile Θ_{+2}.

4.2 Conversions to edge-covering curves

The following sections are ordered by the grid of the edge-covering curves obtained.

4.2.1 Edge-covering curves on (3.4.6.4) from wiggly (3^6)-EC curves

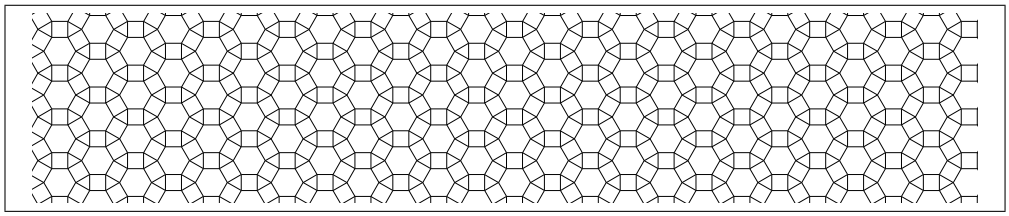

Figure 4.2-A: The (3.4.6.4) grid.

The grid (3.4.6.4) (see Figure 4.2-A) has even incidences on all points, so edge-covering curves do exist. While the grid has been left out in the search, we give two constructions to obtain (3.4.6.4)-EC curves from edge-covering curves on the triangular grid. The grids (3^6), (3.6.3.6), (4.4), and (3.4.6.4) are the only uniform grids where edge-covering curves can exist, so the last gap is closed.

The constructions work only for wiggly curves. We use the terdragon (with map `F` $\mapsto$ `F+F-F`) for our example.

First method. The initial step gives an incomplete curve. Drop all `F`, replace all `+` by `p` and `-` by `m`, then replace all `p` by `---F++++F++++F---F---F++++F++++F---F` and all `m` by `---F++++F---F++F---F---F---F++++F++++F---F`, render with turns by 30°. This is shown in Figure 4.2-B where the colors help with the identification of the replacements. Note that the edges of some hexagons are still missing.

There seems to be no way of including the hexagons by replacements as simple as those just described. Instead we consider the successive edges of the curves as a walk from the

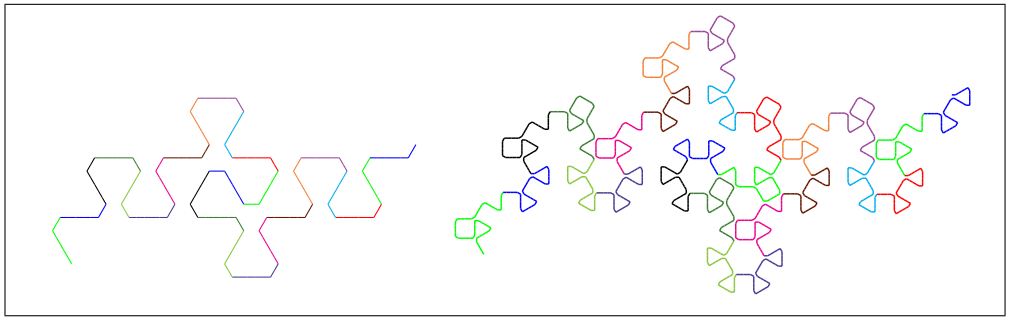

Figure 4.2-B: Third iterate of the terdragon with rounded corners (left) and the corresponding curve on (3.4.6.4) omitting the edges of some hexagons (right).

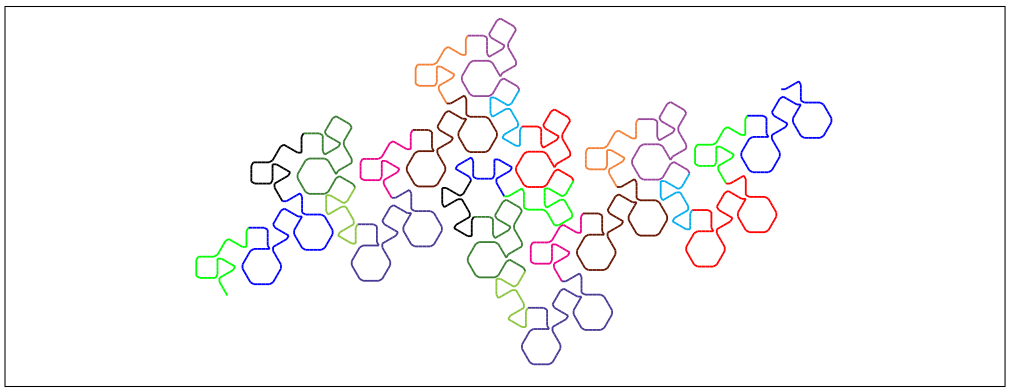

Figure 4.2-C: The completed curve on (3.4.6.4).

left to the right in Figure 4.2-B, visiting the hexagons on the right side of the current edge. This corresponds to a tree traversal visiting all nodes of a tree which are located where hexagons are missing. To fill these in, we keep track of the directions modulo 3, using representatives 0, 1, and 2. We add two hexagons (letter `H`) by changing the replacement for `p` to `---F++++FH++++F---F---F++++FH++++F---F` if the direction is 0 (see the blue replacement of second edge from left in Figure 4.2-C) and one hexagon by changing the replacement for `m` to `---FH++++F---F++F---F---F---F++++F++++F---F` if the direction is 2 (see the olive replacement of fourth edge from left in Figure 4.2-C). Finally, `H` is replaced by `---F++F++F++F++F++F-------`.

Figure 4.2-D shows the curve corresponding to the tile Θ_{-3} of the terdragon.

Second method. For the incomplete curve shown in middle of Figure 4.2-E, drop all `F`, replace all `+` by `++F++F` and all `-` by `---F---F++F`, and use turns by 30°. To fill in the missing triangles (letter `T`), we change the replacement for `-` to `---F---FTF` if the direction is 1, and the replacement for `+` to `TFTF` if the direction is 2. Finally, `T` is replaced by `---F++++F++++F---`.

Figure 4.2-F shows the curve corresponding to the tile Θ_{-4}.

4.2.2 Edge-covering curves on (3.6.3.6)

Conversion from wiggly (3^6)-EC curves. Curves that are (3.6.3.6)-EC can be obtained from wiggly curves on the triangular grid in a similar fashion. Drop all `F` and replace all `+` by `+F+F`, replace all `-` by `--F`, unless the direction is that of the

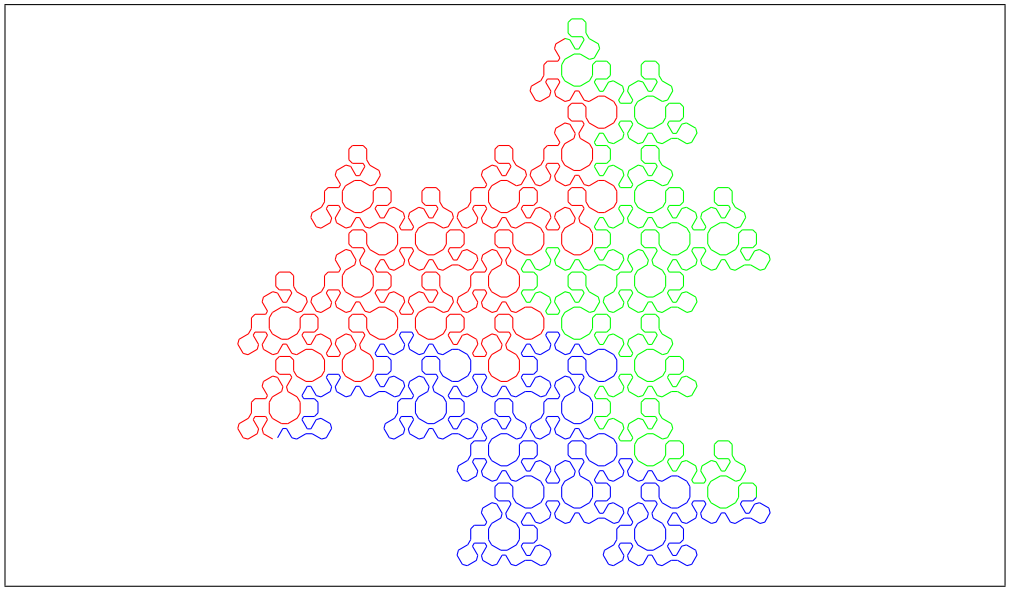

Figure 4.2-D: An edge-covering curve on (3.4.6.4) from the tile Θ_{-3} of the terdragon.

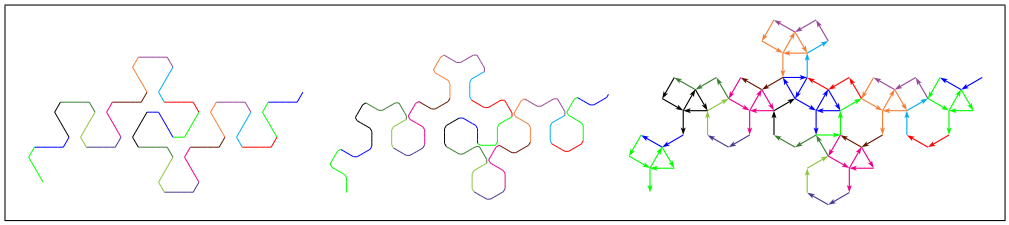

Figure 4.2-E: Third iterate of the terdragon (left), the corresponding curve on (3.4.6.4) omitting the edges of some triangles (middle), and the completed curve (right).

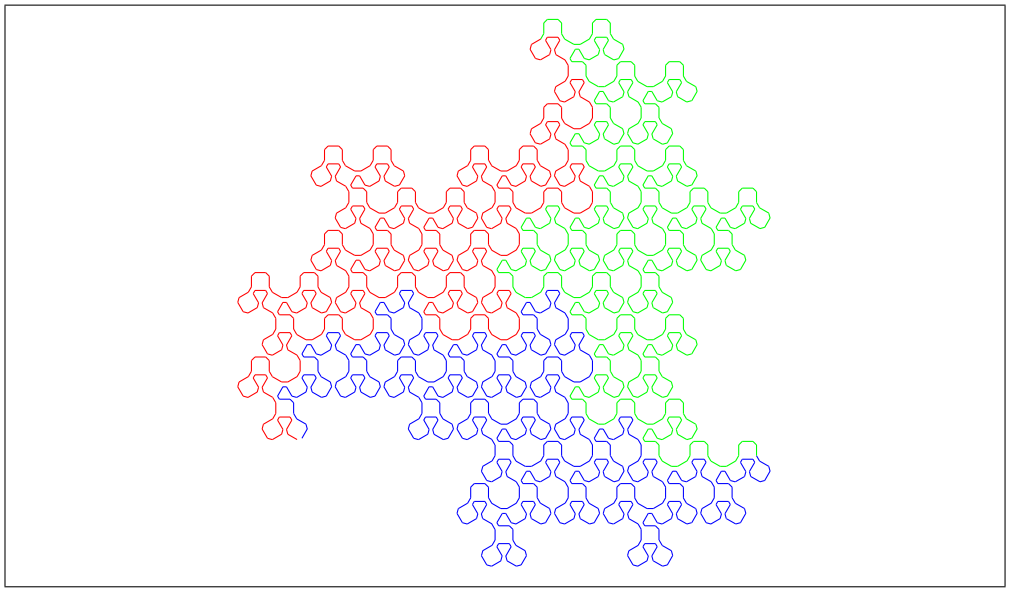

Figure 4.2-F: An edge-covering curve on (3.4.6.4) from the tile Θ_{-4} of the terdragon.

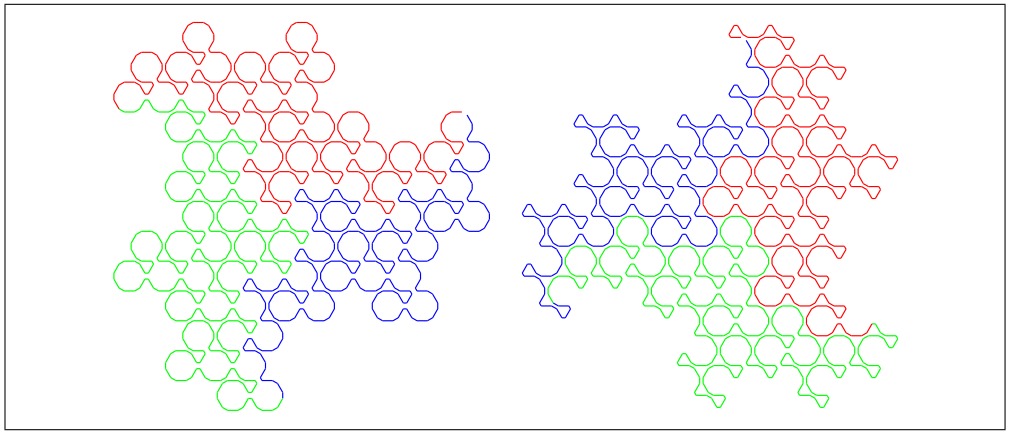

Figure 4.2-G: Edge-covering curves on (3.6.3.6) from the tiles Θ_{+4} (left) and Θ_{-4} (right) of the terdragon.

initial edge, then the replacement for `-` is `+F--F--F+F`. Figure 4.2-G shows the curves obtained from the tiles of the terdragon.

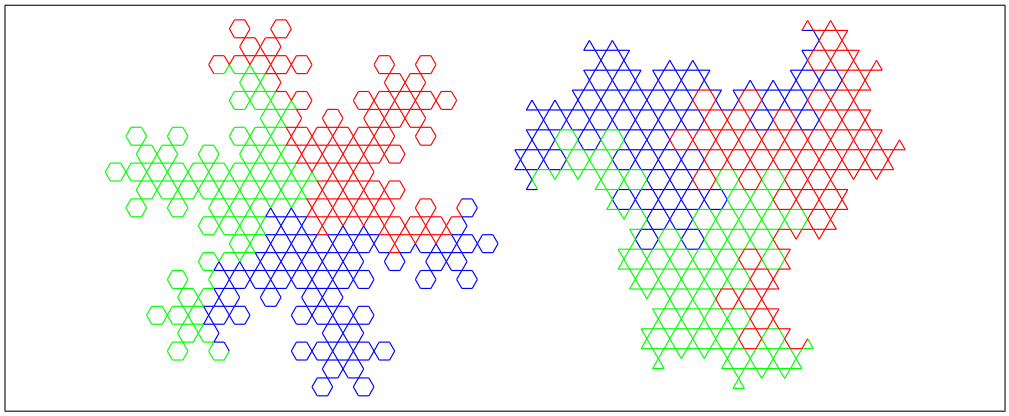

Figure 4.2-H: Edge-covering curves on (3.6.3.6) from the tiles Θ_{+2} (left) and Θ_{-2} (right) of the balanced curve `R13-15` on the triangular grid.

Conversion from balanced (3^6)-EC curves. Balanced curves can be rendered as (3.6.3.6)-EC curves by replacing all `F+` by `F+F+`, `F-` by `F--`, and `F0` by `F+F--F+`. The rendering for the curve `R13-15` with rule `F` $\mapsto$ `F+F0F0F-F-F+F0F+F+F-F0F-F` is shown in Figure 4.2-H.

A conversion from (4^4)-EC curves. Figure 4.2-I shows two (3.6.3.6)-EC curves corresponding to the tiles Θ_{+3} (left) and Θ_{-3} (right) of the curve `R5-1` on the square grid. The conversion is surprisingly easy. Let $k \in \{1, 2, \ldots, 6\}$ denote an edge from 0 to the sixth roots of unity, k standing for $\exp(2\pi\, i\,(k-1)/6)$, as shown in Figure 1.4-A. Horizontal edges are kept, moves to the right and left are respectively mapped to 1 and 4. Vertical edges up and down are respectively mapped to 65 and 32.

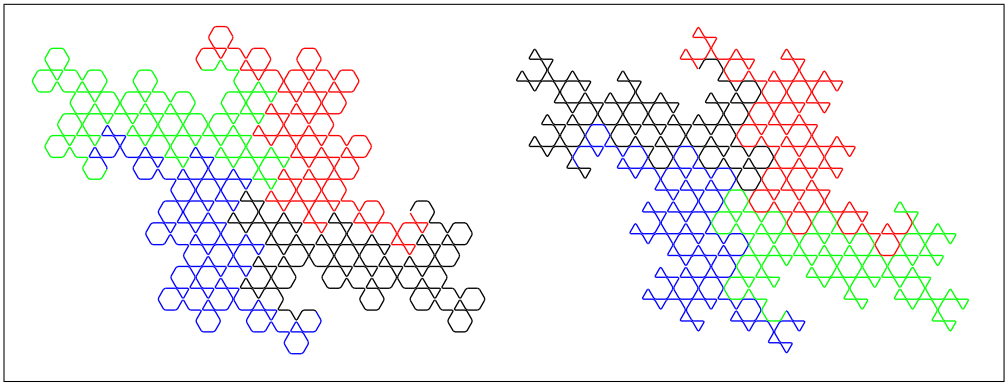

Figure 4.2-I: Edge-covering curves on (3.6.3.6) corresponding to Θ_{+3} (left) and Θ_{-3} (right) of the (4^4)-EC curve `R5-1` whose tile Θ_{+5} is shown in Figure 3.3-B.

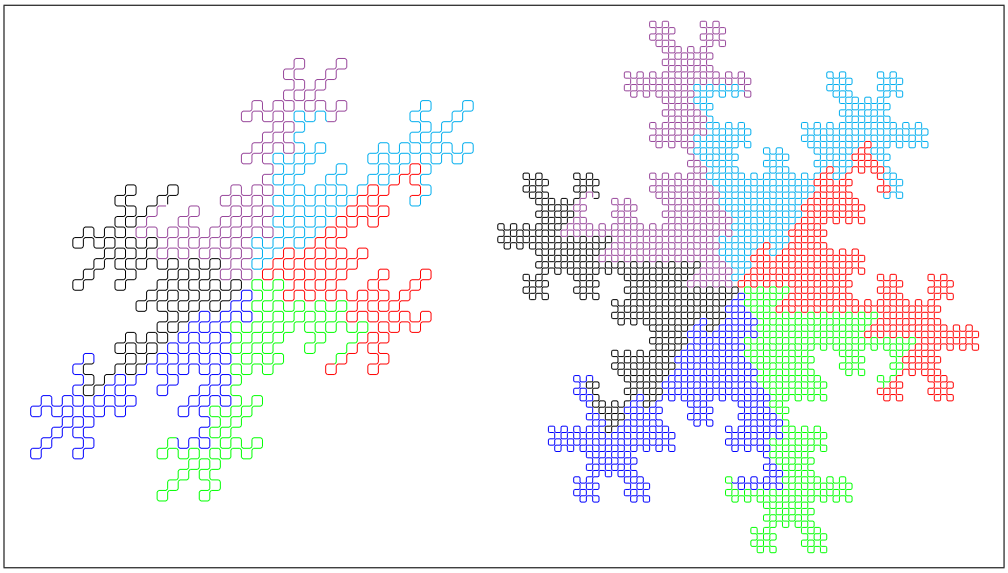

Figure 4.2-J: Two edge-covering curves on the square grid corresponding to Θ_{+2} of the curve `R13-1` on the tri-hexagonal grid.

4.2.3 Edge-covering curves on (4^4) from (3.6.3.6)-EC curves

A simple mapping of the directions on the tri-hexagonal grid to those of the square grid gives the (4^4)-EC curve shown at the left of Figure 4.2-J: map the directions (as shown in Figure 1.4-A on page 8) 1, 2, ..., 6 respectively to 21, 2, 3, 43, 4, and 1. A less distorted curve results from the mappings to 141, 2321, 2, 3, 4323, and 414. The still less distorted curve shown at the right of Figure 4.2-J uses the mappings to 12321, 2343212, 3432, 34143, 4121434, and 1214.

4.2.4 Edge-covering curves on (3^6) from (3.6.3.6)-EC curves

By simply dropping edges with every second of the six directions in a curve on the tri-hexagonal grid a (3^6)-EC curve is obtained, see Figure 4.2-K. There always exist curves on the triangular grid with tiles of the same (6-fold symmetric) shape in the limit as the shape of Θ_+ on the tri-hexagonal grid, compare Figure 4.2-L and Figure 3.1-G on page 18.

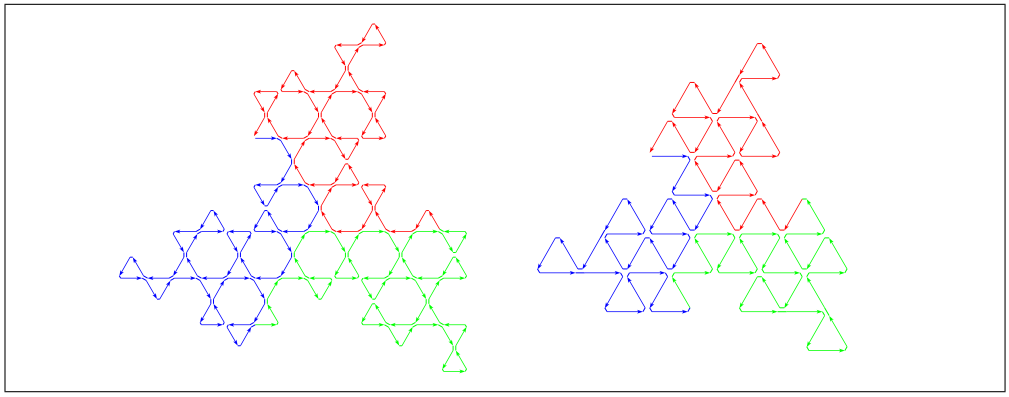

Figure 4.2-K: The tile Θ_{-2} of the curve `R7-1` on the tri-hexagonal grid (left). Simply dropping every second row of triangles gives a (3^6)-EC curve (right).

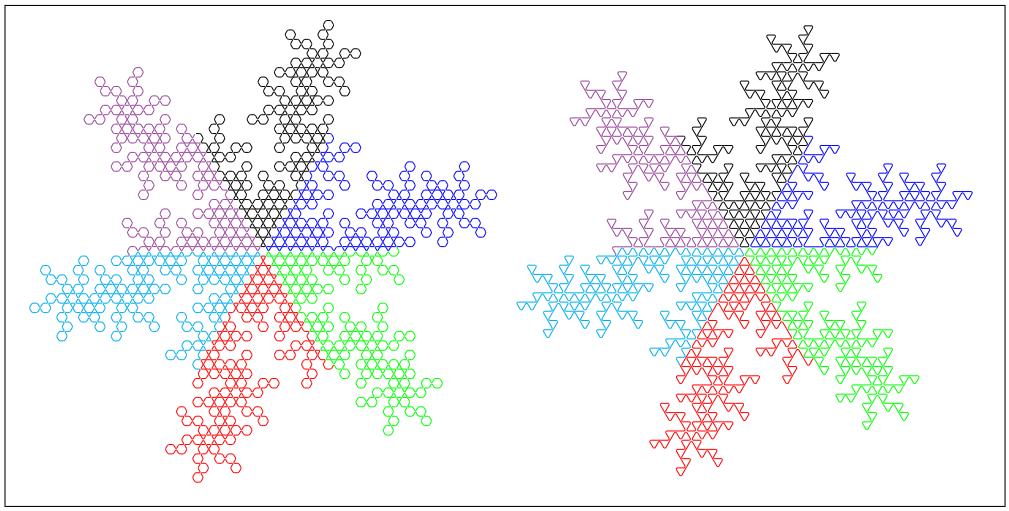

Figure 4.2-L: The tile Θ_{+2} of the curve `R25-11` on the tri-hexagonal grid (left) and the corresponding (3^6)-EC curve (right).

5 Multiplying and dividing curves

Two curves on the same grid of orders R_1 and R_2 can be combined to obtain curves whose order is $R_1\,R_2$ via a non-commutative product of the respective L-systems. This construction is equivalent to Dekking's *folding convolution* [10, Equation 1, pp. 21]. *folding convolution*

In the other direction, any curve can be divided into parts of equal lengths by a division procedure of the L-system.

5.1 Products of curves

For two simple L-systems L_1 and L_2 with maps `F` $\mapsto$ `f(F)` and `F` $\mapsto$ `g(F)` we define the *product* L_1L_2 as the L-system with map `F` $\mapsto$ `g(f(F))`. This product is obviously non-commutative in general. *product*

For an example we use the curves on the triangular grid with L-systems with maps `F` $\mapsto$ `F+F-F` (terdragon, L_1) and `F` $\mapsto$ `F+FOF-F` (crab, L_2). The products L_1L_2 and L_2L_1 respectively give the curves shown in Figure 5.1-A.

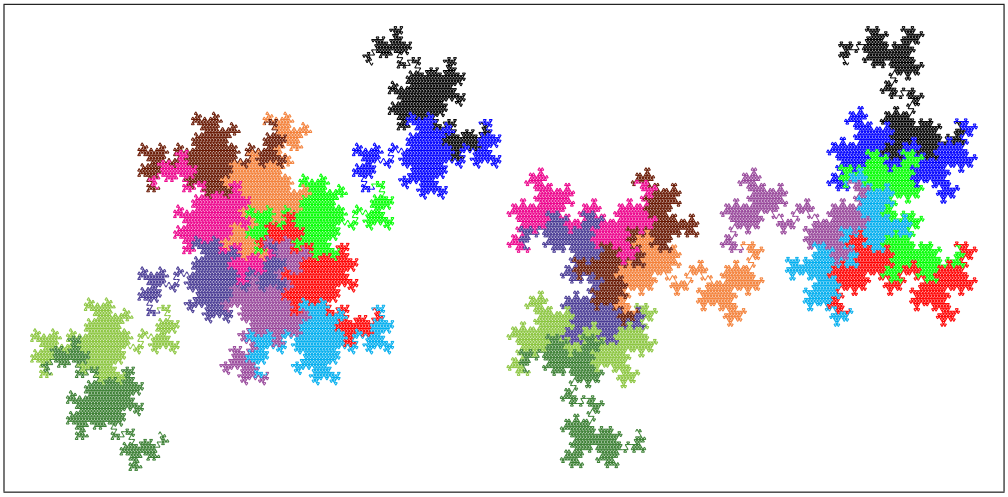

Figure 5.1-A: Products of the terdragon (`F` ↦ `F+F-F`) and the curve `F` ↦ `F+F0F-F`. The left curve is `R12-10`, the right one is `R12-17`.

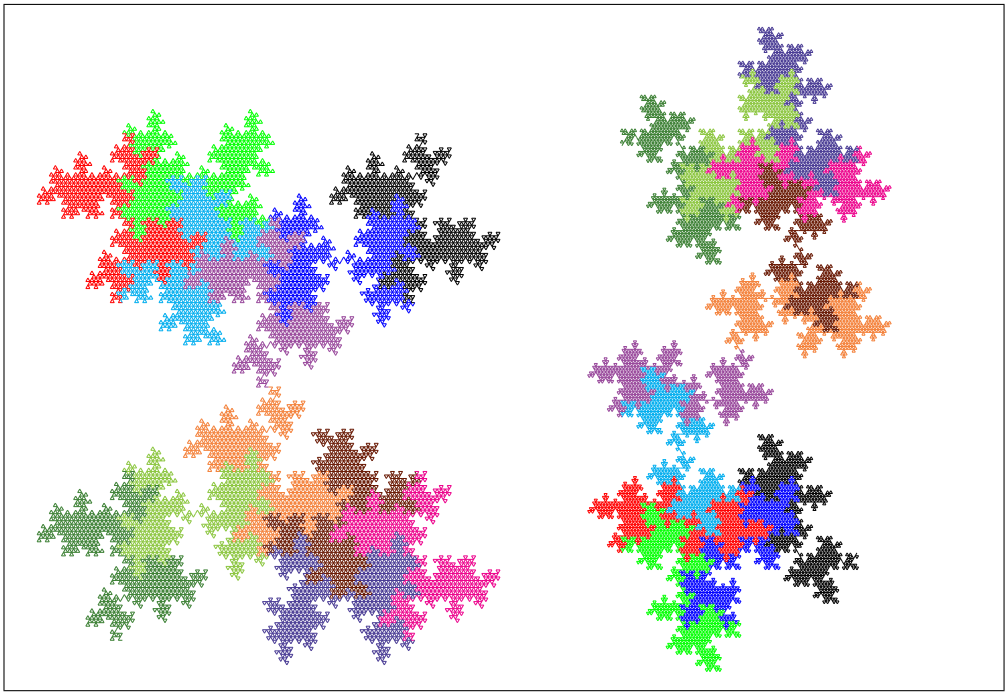

Figure 5.1-B: Products of the reversed terdragon (`F` ↦ `F-F+F`) and the curve `F` ↦ `F+F0F-F`. The left curve is `R12-13`, the right one is `R12-25`.

The map of the product L_1L_2 is computed as second iterate of `F` with maps `F` ↦ `G+G-G` and `G` ↦ `F+F0F-F`. For L_2L_1 use the L-system with maps `F` ↦ `G+G0G-G` and `G` ↦ `F+F-F`. The respective maps for the simple L-systems are `F` ↦ `(F+F0F-F) + (F+F0F-F) - (F+F0F-F)` and `F` ↦ `(F+F-F) + (F+F-F) 0 (F+F-F) - (F+F-F)` where spaces and parentheses emphasize the structure.

Fixing the axiom `F` and using k curves of pairwise different shapes we obtain $k!$ new curves as products, one for every permutation of the maps. The order of the product curve is the product of the orders of all curves used.

Still more curves can be obtained by reversing the map in one (or more) curves in the product. Reversing the map for the terdragon (to `F` ↦ `F-F+F`) in the example leads to respectively the maps `F` ↦ `(F+F0F-F) - (F+F0F-F) + (F+F0F-F)` and `F` ↦ `(F-F+F) + (F-F+F) 0 (F-F+F) - (F-F+F)` whose curves are shown in Figure 5.1-B.

In general the reversal of a map is not the same as the map with signs swapped, so even more products can be made (this does not work for the curves on the tri-hexagonal grid). In general a curve is distinct from the ones obtained by reversal, swapping signs, and doing both in the production of `F`, so even more products exist.

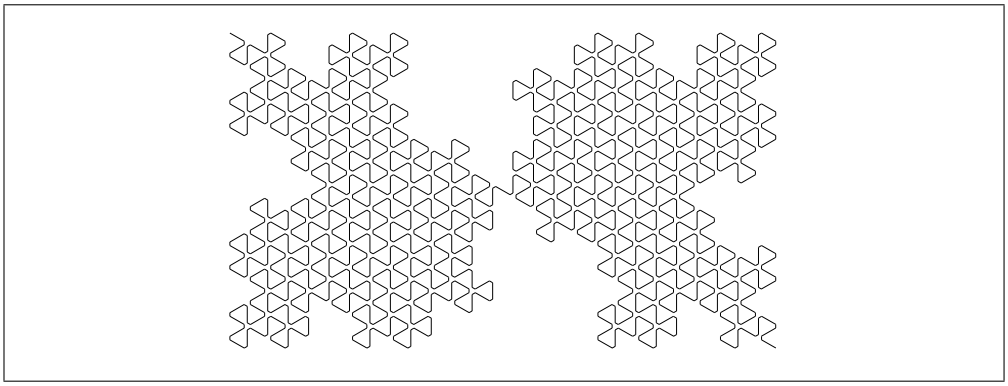

Figure 5.1-C: Curve (`R9-8`) for the product of the maps `F` ↦ `F+F-F` and `F` ↦ `F-F+F`.

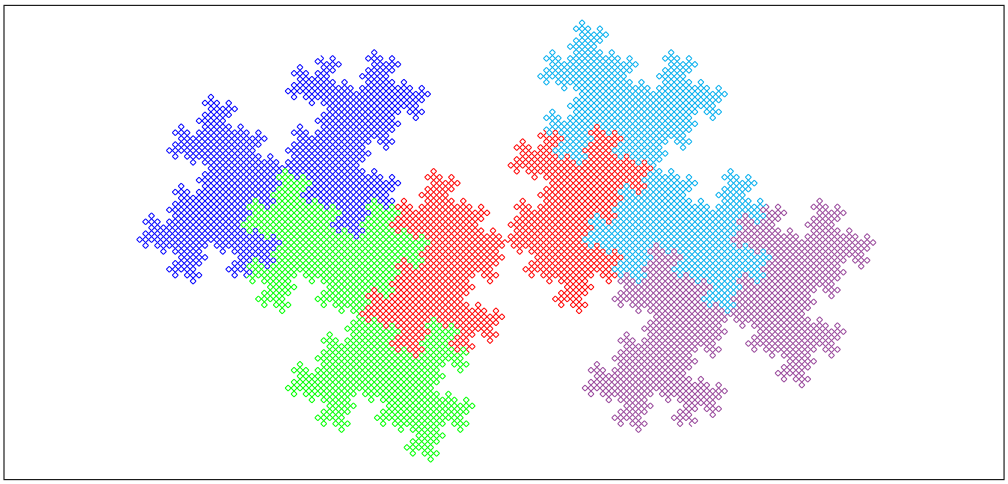

Figure 5.1-D: Curve (`R25-45`) for the product of the maps `F` ↦ `F+F+F-F-F` and `F` ↦ `F-F-F+F+F`, consisting of 5 smaller copies of itself that are flipped over.

A particularly simple way of multiplying an L-system with the L-system with swapped signs is to use the maps `+` ↦ `-` and `-` ↦ `+`. For example, the L-system (`F` ↦ `G+G-G`, `G` ↦ `F-F+F`, `+` ↦ `+`, `-` ↦ `-`) is (for the even iterates) the same as (`F` ↦ `F-F+F`, `+` ↦ `-`, `-` ↦ `+`). The resulting curve is shown in Figure 5.1-C, it is called the *alternate terdragon* in [7, Figure 21, p. 600], it also appears on page 59 in [25]. *alternate terdragon*

The curve shown at the bottom of page 84 in [25] can be obtained in this way from the one at the top of the same page (the R5-dragon, also shown in Figure 1.1-B). It is a curve of order 25 (not 5) according to our conventions. Figure 5.1-D shows the curve decomposed into 5 smaller copies of itself that are flipped over. All product curves of this kind have this sort of self-similarity, Figure 5.1-E shows this for the

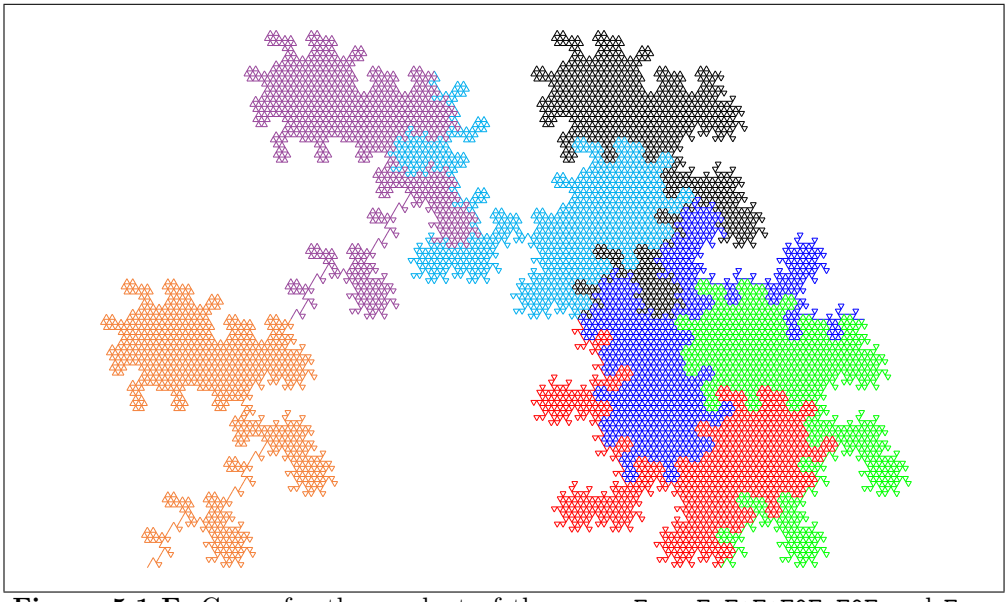

Figure 5.1-E: Curve for the product of the maps `F` ↦ `F-F+F+FOF-FOF` and `F` ↦ `F+F-F-FOF+FOF`, consisting of 7 smaller copies of itself that are flipped over.

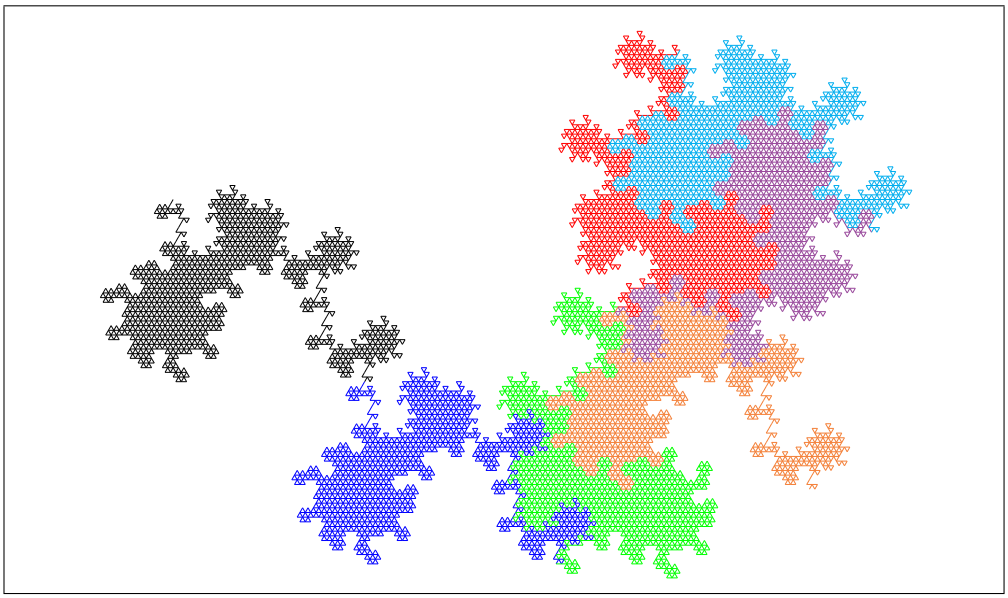

Figure 5.1-F: Curve for the product of the maps `F` ↦ `FOF-FOF+F+F-F` and `F` ↦ `F+F-F-FOF+FOF`.

curve `R7-1` on the triangular grid, shown in Figure 1.2-D on page 5. A similar (but different) curve appears on the bottom of page 107 in [25], it is shown in Figure 5.1-F.

5.2 Divisions of a curve

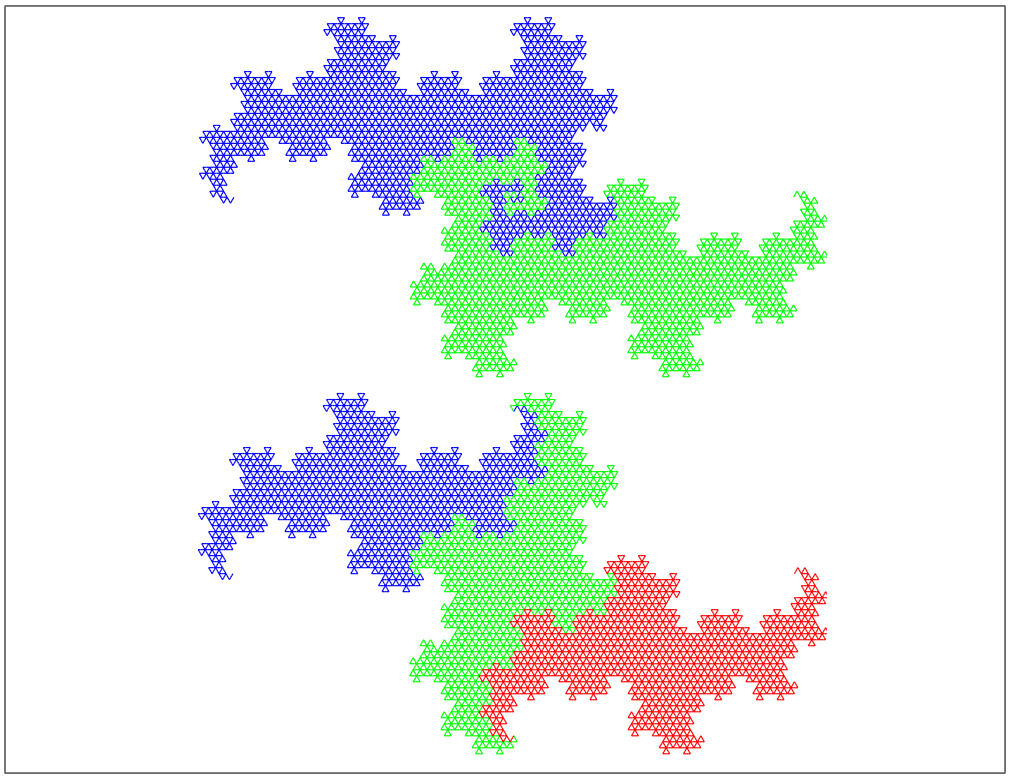

Figure 5.2-A: Dividing the terdragon (`R3-1`) by 2 and 3.

The curve for a simple L-system of order R can be divided into parts with equal numbers d of edges as follows. Replace each `F` in the production of `F` by $s_1 s_2 \dots s_d$ (where the s_j are any pairwise different symbols), then split the word obtained into d parts each containing R letters s_j, these are the productions of the letters s_j for $1 \le j \le d$.

The jth part ($1 \le j \le d$) of the curve corresponds to the axiom s_j.

For example, to divide the terdragon ($R = 3$) into $d = 5$ parts, first replace each `F` in the production `F+F-F` by `ABCDE` to obtain the word `ABCDE+ABCDE-ABCDE`, split it into 5 parts of length 3, used as productions, giving the maps `A` $\mapsto$ `ABC`, `B` $\mapsto$ `DE+A`, `C` $\mapsto$ `BCD`, `D` $\mapsto$ `E-AB`, and `E` $\mapsto$ `CDE`.

The corresponding curves, concatenated to give the terdragon by using the axiom `ABCDE`, is shown in Figure 5.2-B (bottom). The division of a curve of order R by $d = R$ (here 3, see bottom of Figure 5.2-A) is just another way to observe the self-similarity of the curve. The division by two (see top of Figure 5.2-A) appears on page 59 in [25].

The shape of the curve obtained by division is dependent on the curve, not only on its shape. The division by 2 of the curves `R7-2` and `R7-5` whose motifs are shown in Figure 2.2-A on page 11 appear on pages 101 and 103 in [25], see Figures 5.2-C and 5.2-D.

Indeed a bit more is possible. After substitution of the letter `F` in its own production by a sequence of d symbols, we can partition the word into maps of length $\neq d$. As long as no sequence of maps is just a cyclic permutation of the letters such as `B` $\mapsto$ `C`, `C` $\mapsto$ `E`, `E` $\mapsto$ `B` (a fixed point such as `B` $\mapsto$ `B` is a special case of this), we obtain any

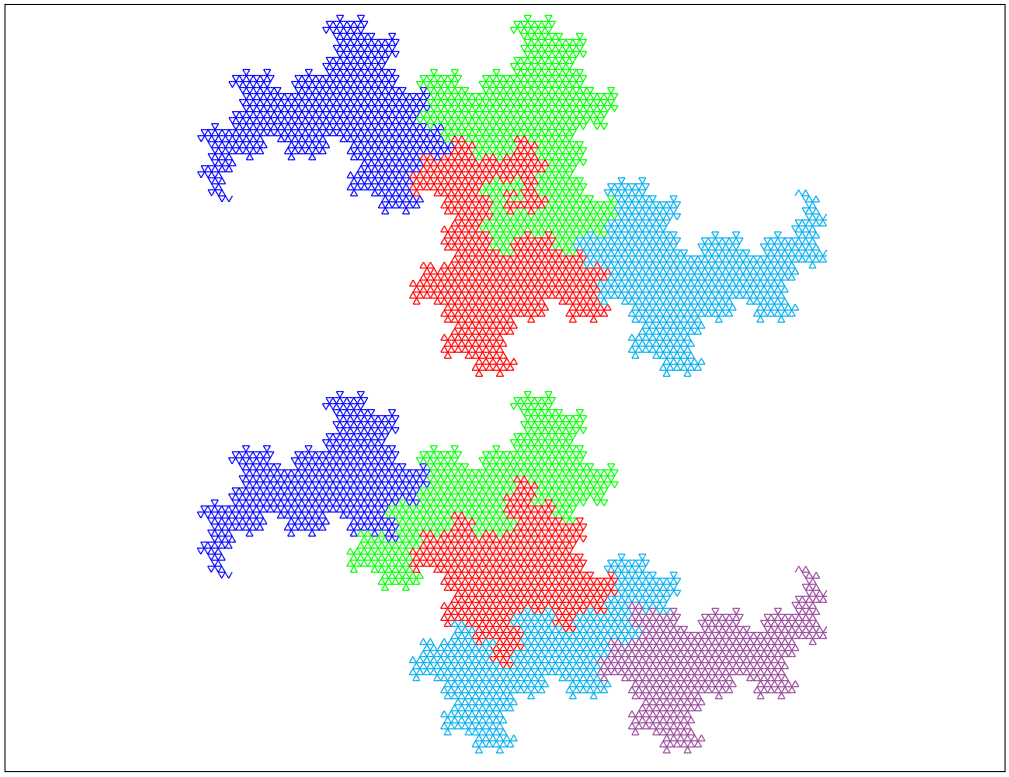

Figure 5.2-B: Dividing the terdragon (`R3-1`) by 4 and 5.

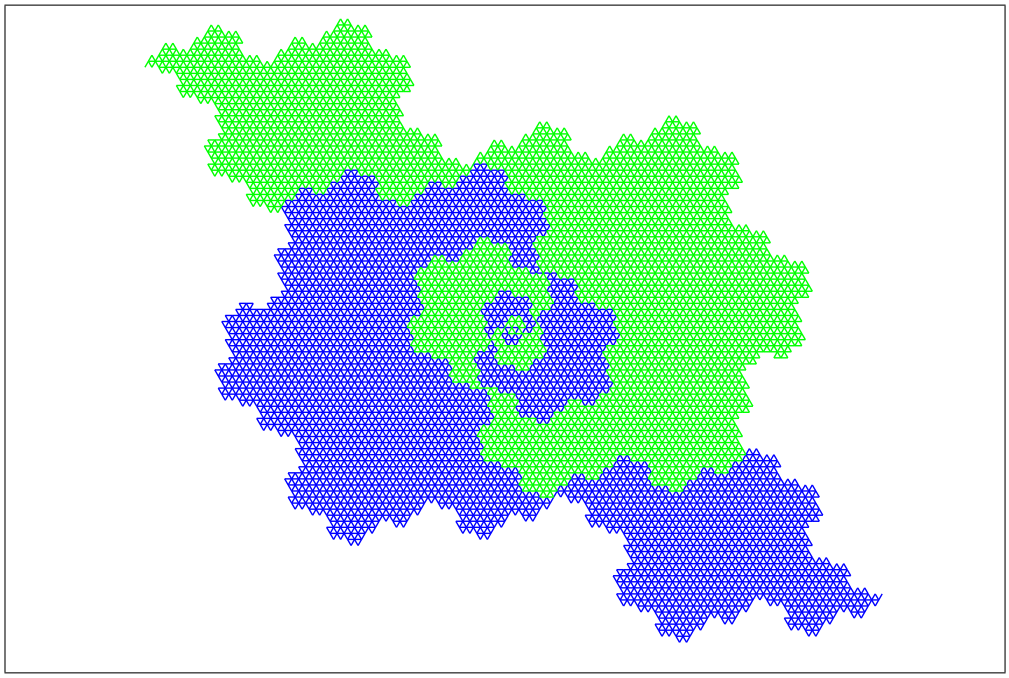

Figure 5.2-C: Division of the curve `R7-2` on the triangular grid by 2.

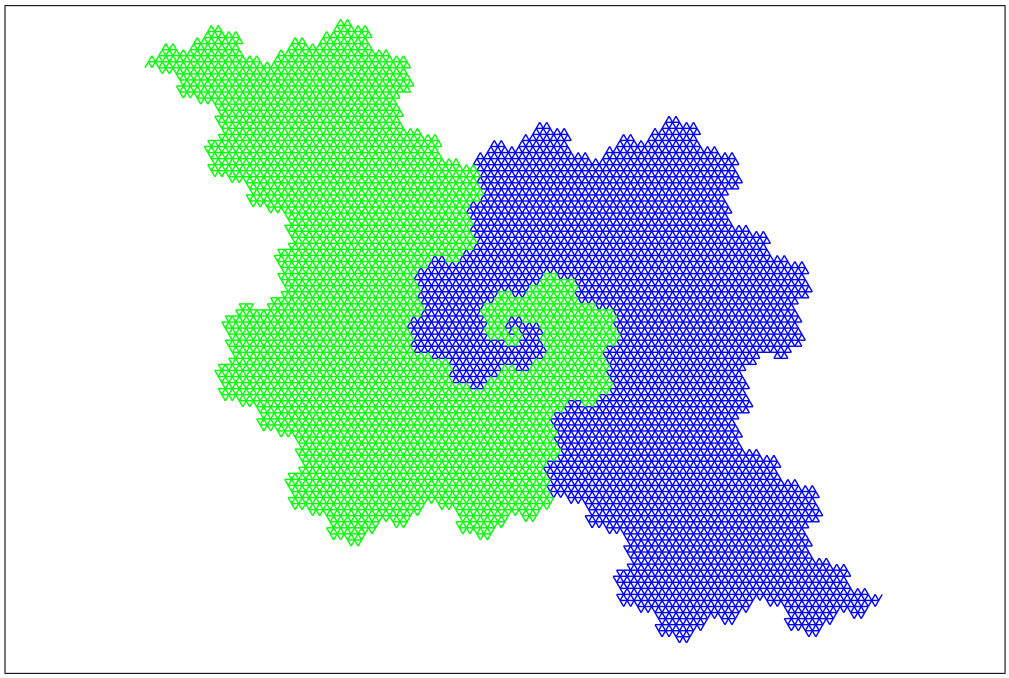

Figure 5.2-D: Division of the curve `R7-5` on the triangular grid by 2.

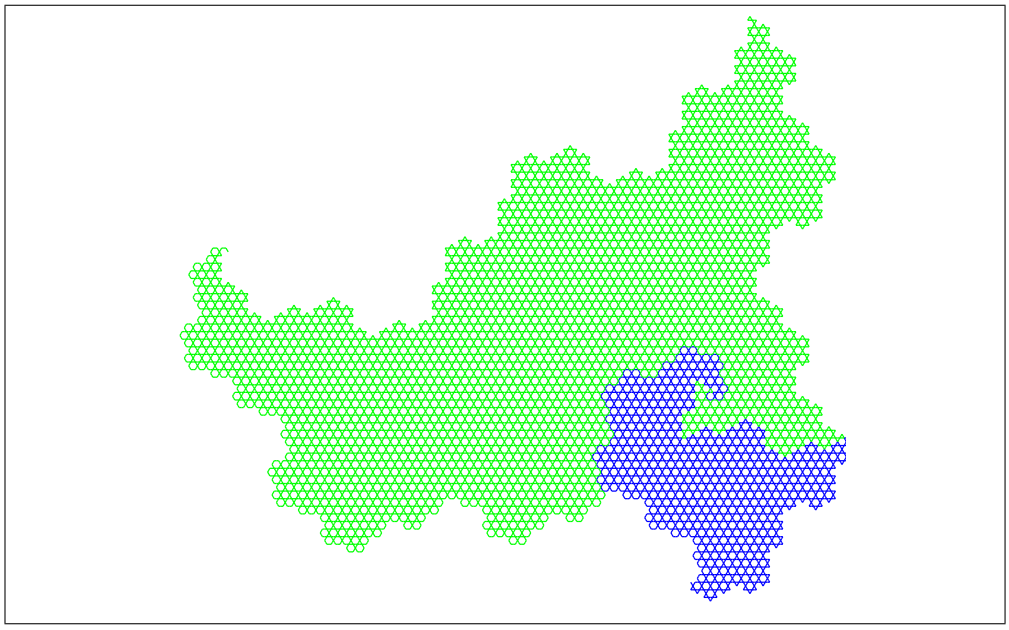

Figure 5.2-E: Ventrella's curve split into parts of 3/14 and 11/14 of the total length.

partition of $R\,d$ into d parts, not just the partition into equal parts R. For example, for Ventrella's curve with map `F` ↦ `F+F+F+F--F--F+F` (`R7-1` on the tri-hexagonal grid see Figure 1.1-D on page 3), we can split the word `AB+AB--AB--AB+AB+AB+AB` into the productions for the maps `A` ↦ `AB+A` and `B` ↦ `B--AB--AB+AB+AB+AB` to partition the curve into parts 3 and $14 - 3 = 11$, this is shown in Figure 5.2-E.

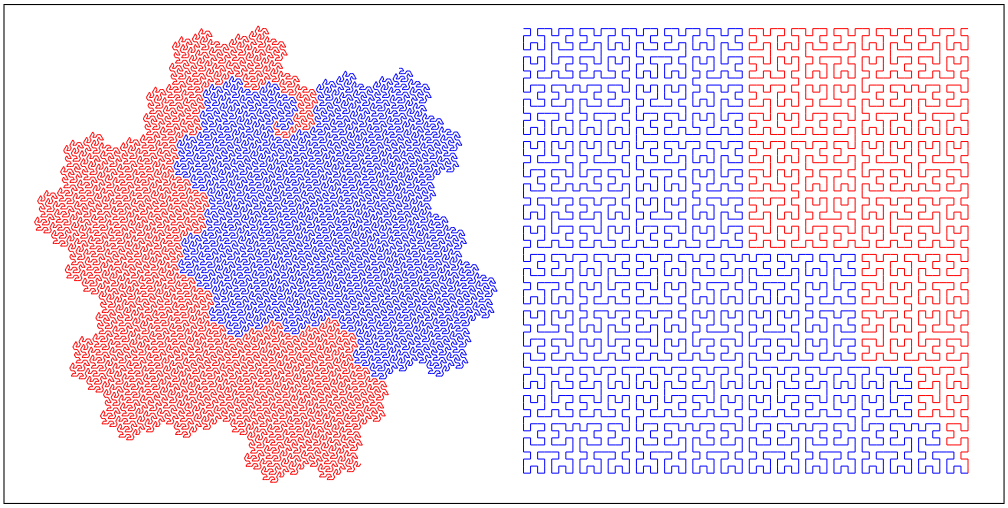

Figure 5.2-F: Gosper's flowsnake split into halves (left) and the Hilbert curve split into parts of 2/3 and 1/3 of the total length (right).

Splitting curves with non-simple L-systems can similarly be achieved. For Gosper's curve with maps `L` ↦ `L+R++R-L--LL-R+` and `R` ↦ `-L+RR++R+L--L-R`, replace all `L` by `AB` and all `R` by `CD` to obtain the two words `AB+CD++CD-AB--ABAB-CD+` and `-AB+CDCD++CD+AB--AB-CD`. Divide both evenly, giving the maps `A` ↦ `AB+CD++CD-A`, `B` ↦ `B--ABAB-CD+`, `C` ↦ `-AB+CDCD++C`, and `D` ↦ `D+AB--AB-CD`. The split into halves using the axiom `AB` is shown at the left in Figure 5.2-F.

For the Hilbert curve with the maps `L` ↦ `+Rt-LtL-tR+` and `R` ↦ `-Lt+RtR+tL-` (where finally only the letters `t` are used for drawing edges), we obtain the two words `+CDt-ABtAB-tCD+` and `-ABt+CDtCD+tAB-`. Note that the letters `t` are kept unmodified. Here we divide unevenly to make the picture more interesting, using the maps `A` ↦ `+CDt-ABtA`, `B` ↦ `B-tCD+`, `C` ↦ `-ABt+CDtC`, and `D` ↦ `D+tAB-`. The split resulting from the axiom `AB` is shown at the right in Figure 5.2-F.

6 Eye of the beholder

We show the fourth iterates for all shapes of curves for the triangular, square, and tri-hexagonal grid of respectively orders $R = 12$, $R = 17$, and $R = 19$, using coloration to make the self-similarity apparent.

```
01:  F F0F+F0F-F+F-F-F0F+F+F-F  R12-1  #
02:  F F0F+F+F-F0F-F-F+F0F+F-F  R12-3  #
03:  F F0F+F+F-F-F0F+F+F-F-F0F  R12-4  # # symm-dr
04:  F F0F+F-F+F0F+F-F-F0F-F+F  R12-6  #
05:  F F+F0F+F-F+F0F-F+F-F0F-F  R12-9  # # symm-dr
06:  F F+F0F+F-F-F0F+F+F-F0F-F  R12-10  # # symm-dr
07:  F F+F0F-F+F+F-F+F-F-F+F-F  R12-12  #
08:  F F+F0F-F-F+F0F-F+F+F0F-F  R12-13  # # symm-dr
09:  F F+F-F+F+F-F0F+F-F-F+F-F  R12-17  # # symm-dr
10:  F F+F-F-F+F-F0F+F-F+F+F-F  R12-25  # # symm-dr
```

Figure 6.0-A: The curves of order 12 on triangular grid without those duplicating a prior shape.

For the triangular grid we skip four curves already seen, `R12-10` and `R12-17` appear in Figure 5.1-A, `R12-13` and `R12-25` in Figure 5.1-B.

```
01:  F F+F+F-F+F-F-F-F+F-F+F+F+F-F+F-F-F  R17-1  # # symm-dr
02:  F F+F+F-F-F+F+F+F-F-F+F-F-F-F+F+F-F-F  R17-2  # # symm-dr
03:  F F+F+F-F-F+F+F+F-F-F+F+F-F-F-F+F+F-F  R17-3  #
04:  F F+F-F+F+F+F-F-F+F-F+F+F-F-F-F+F+F-F  R17-6  # # symm-dr
05:  F F+F-F+F+F-F+F+F-F-F-F+F+F-F-F-F+F  R17-10  #
06:  F F+F-F+F+F-F+F-F+F+F-F-F-F+F-F-F+F  R17-11  #
```

Figure 6.0-B: The curves of order 17 on the square grid without those duplicating a prior shape.

```
01:  F F+F+F+F+F--F+F+F+F--F+F+F--F--F+F+F--F+F--F  R19-1  #
02:  F F+F+F+F+F--F--F+F+F--F--F+F+F--F+F+F+F+F--F  R19-2  #
03:  F F+F+F+F--F+F--F--F+F+F+F--F--F+F+F+F+F+F--F  R19-3  #
04:  F F+F+F+F--F--F+F+F--F--F+F+F--F+F+F+F+F--F+F  R19-5  #
05:  F F+F+F--F+F+F+F--F--F+F--F+F--F+F+F+F+F+F--F  R19-6  #
06:  F F+F+F--F+F--F--F+F+F+F--F+F--F+F+F+F+F+F--F  R19-7  #
07:  F F+F--F+F+F+F--F--F+F--F+F+F--F+F+F+F+F+F--F  R19-14  #
```

Figure 6.0-C: The curves of order 19 on the tri-hexagonal grid without those duplicating a prior shape.

For the tri-hexagonal the third iterate of curve `R19-1` already appeared in Figure 3.4-B.

6.1 Triangular grid

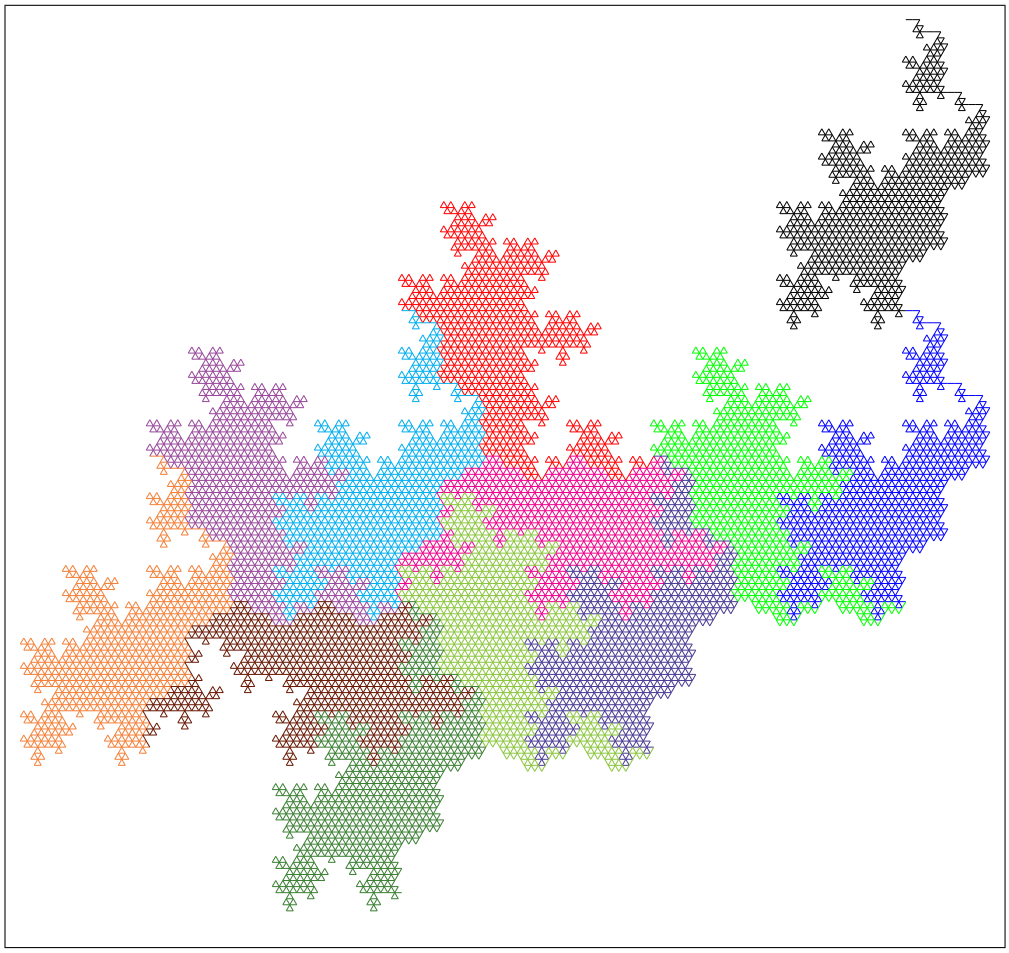

Figure 6.1-A: `R12-1` on the triangular grid.

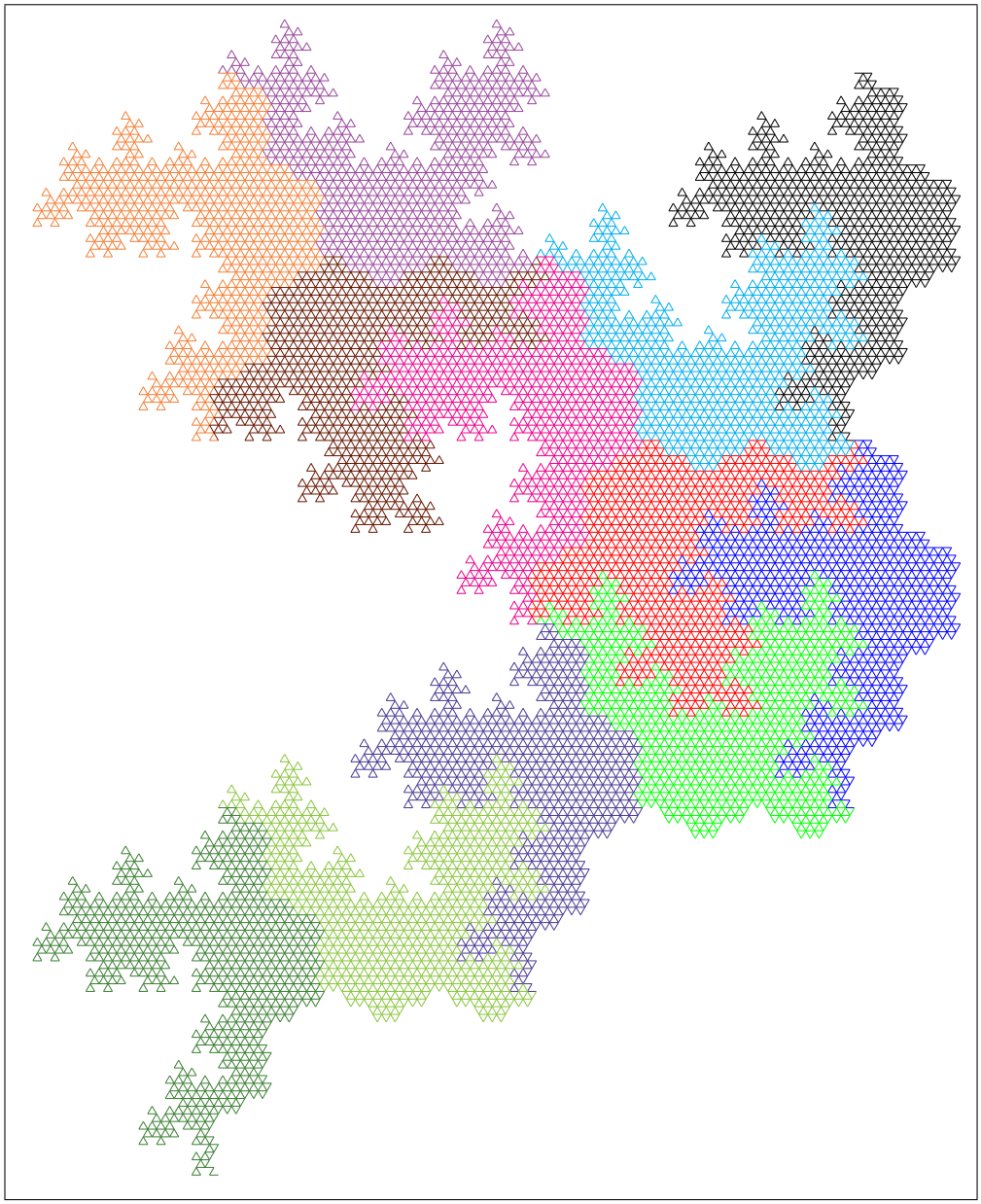

Figure 6.1-B: `R12-3` on the triangular grid.

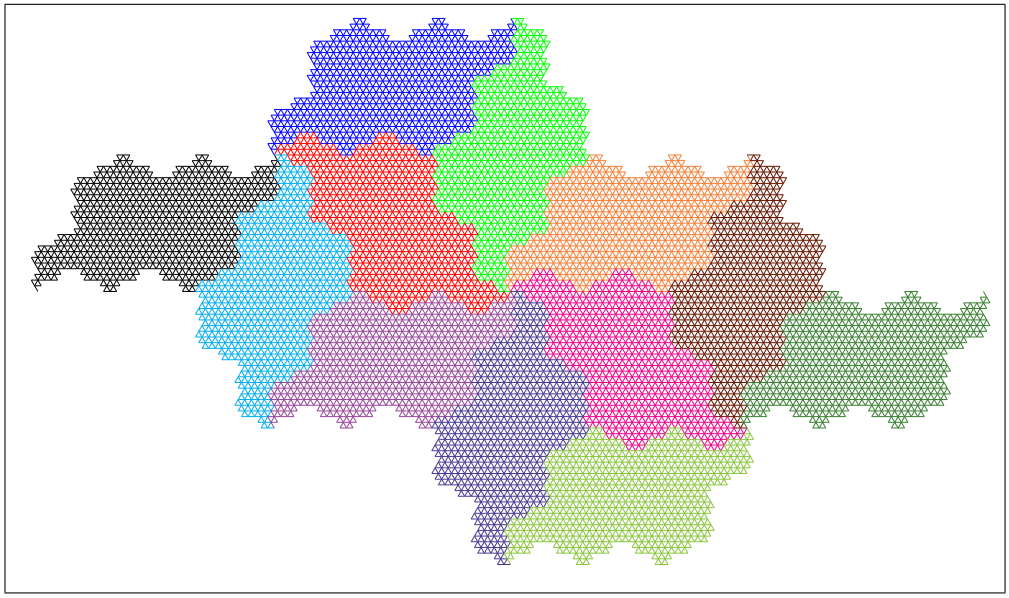

Figure 6.1-C: `R12-4` on the triangular grid.

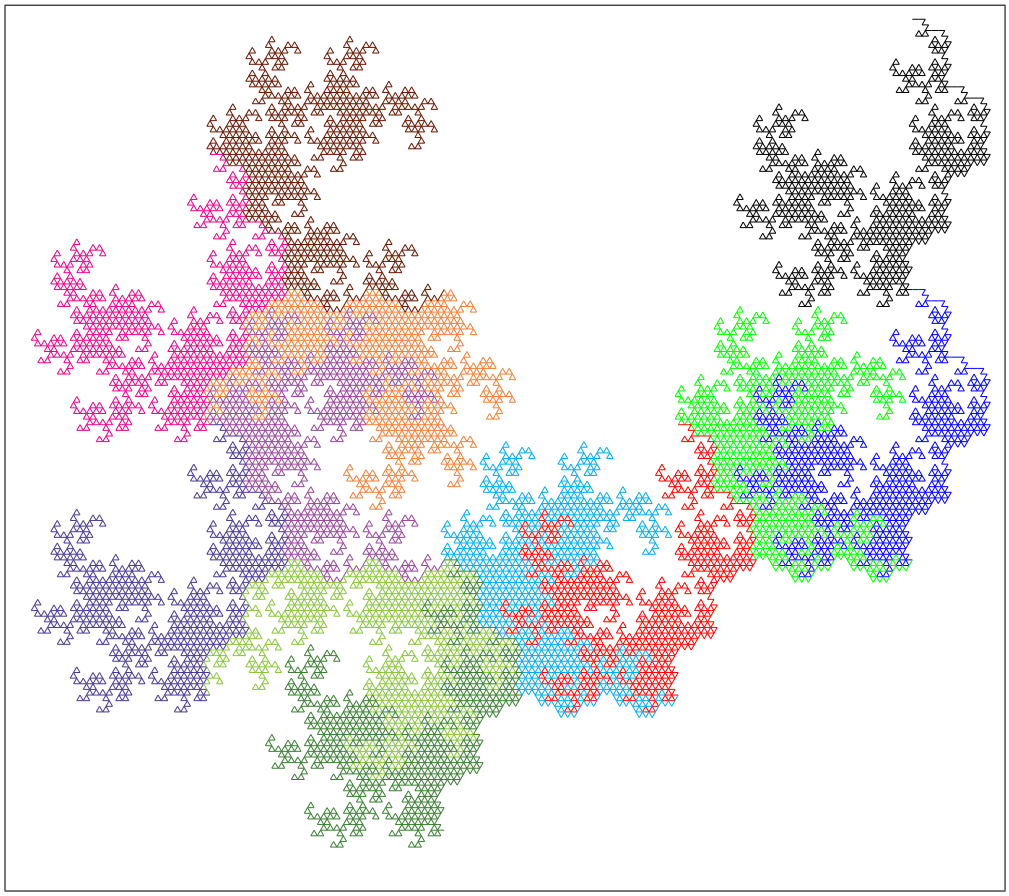

Figure 6.1-D: `R12-6` on the triangular grid.

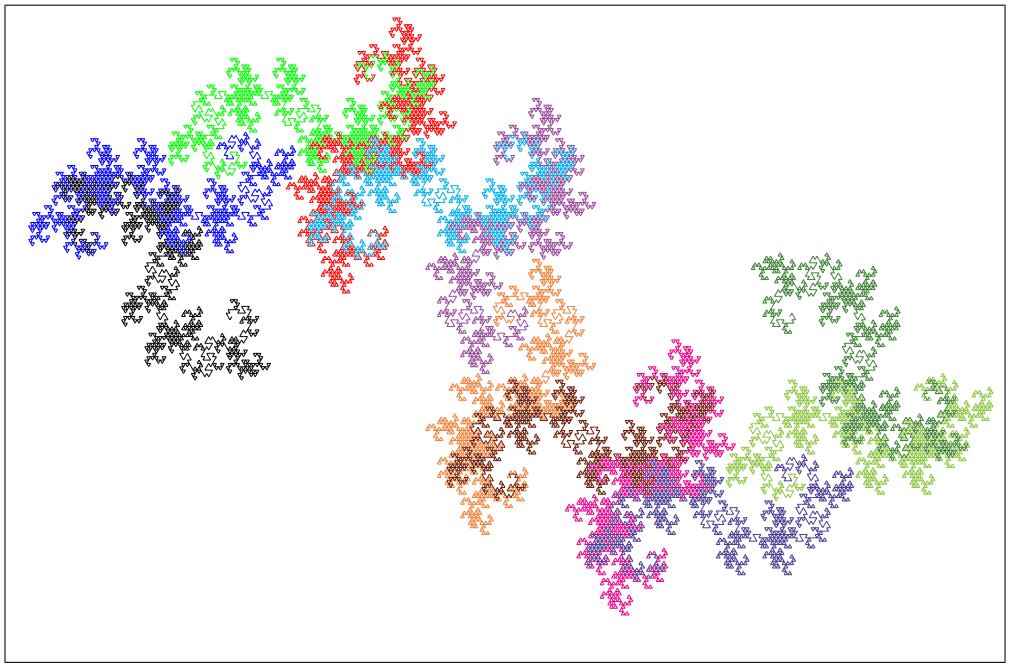

Figure 6.1-E: `R12-9` on the triangular grid.

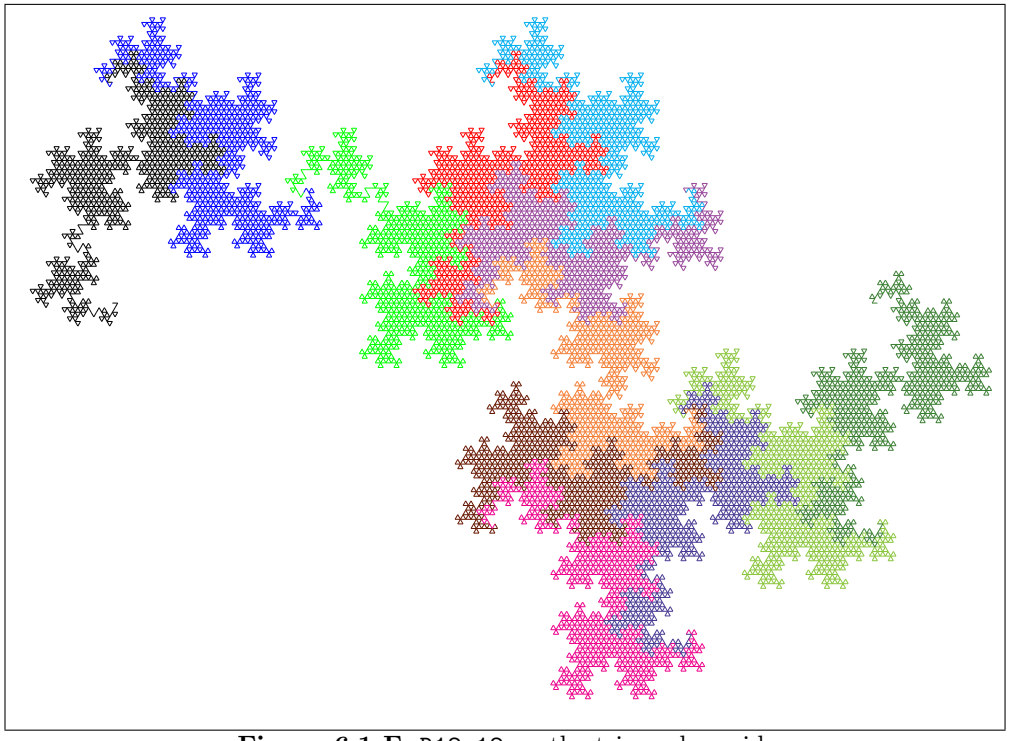

Figure 6.1-F: `R12-12` on the triangular grid.

6.2 Square grid

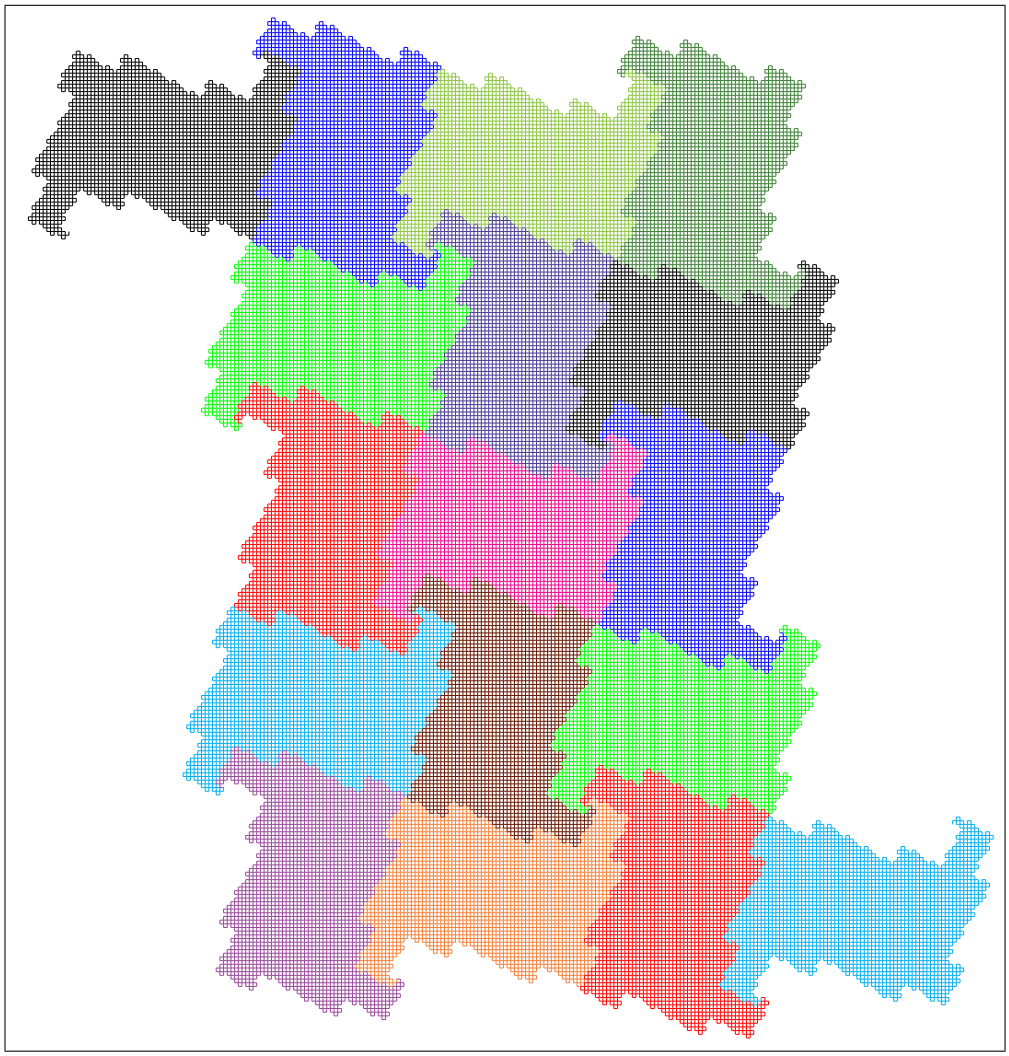

Figure 6.2-A: `R17-1` on the square grid.

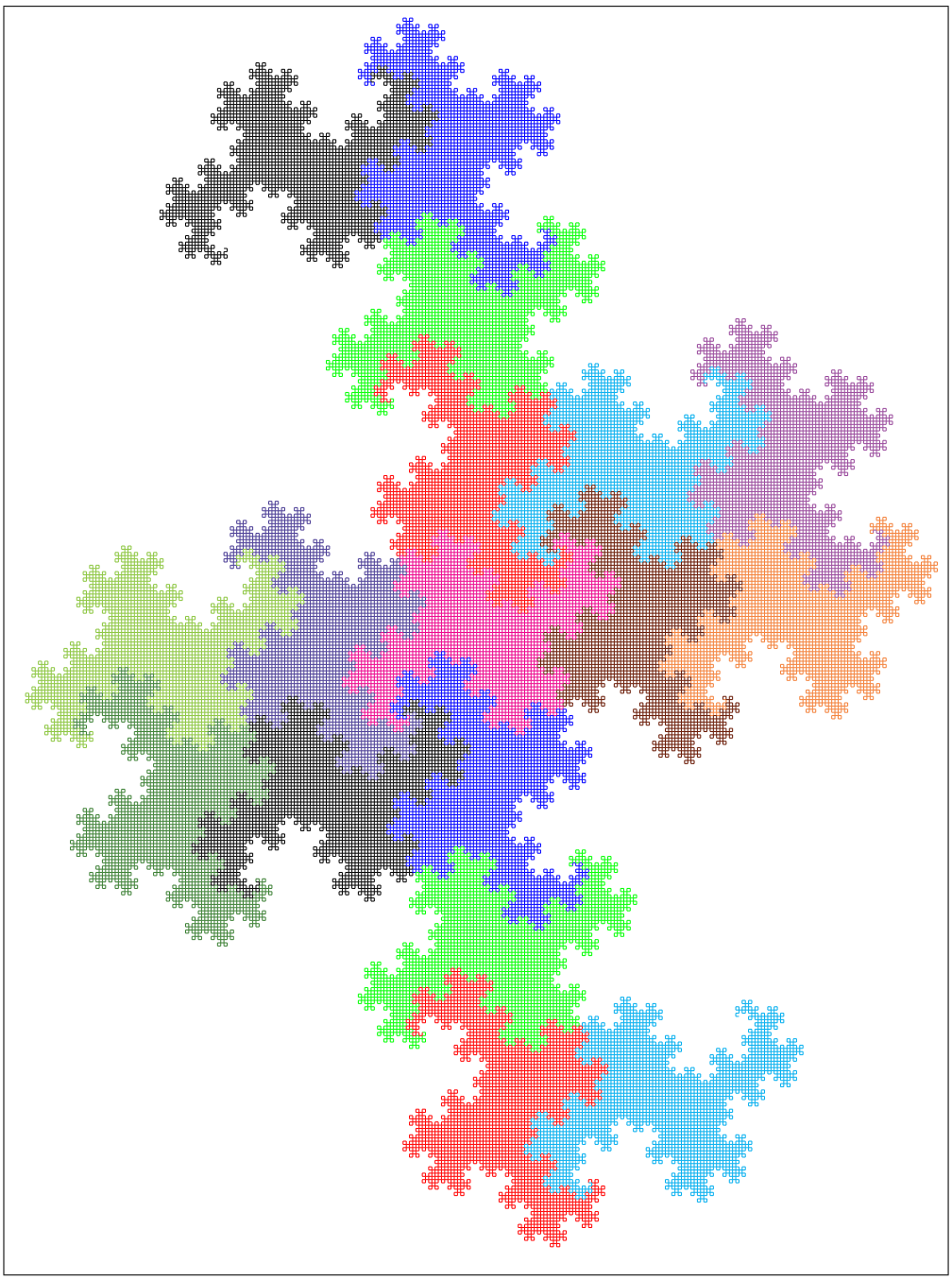

Figure 6.2-B: `R17-2` on the square grid.

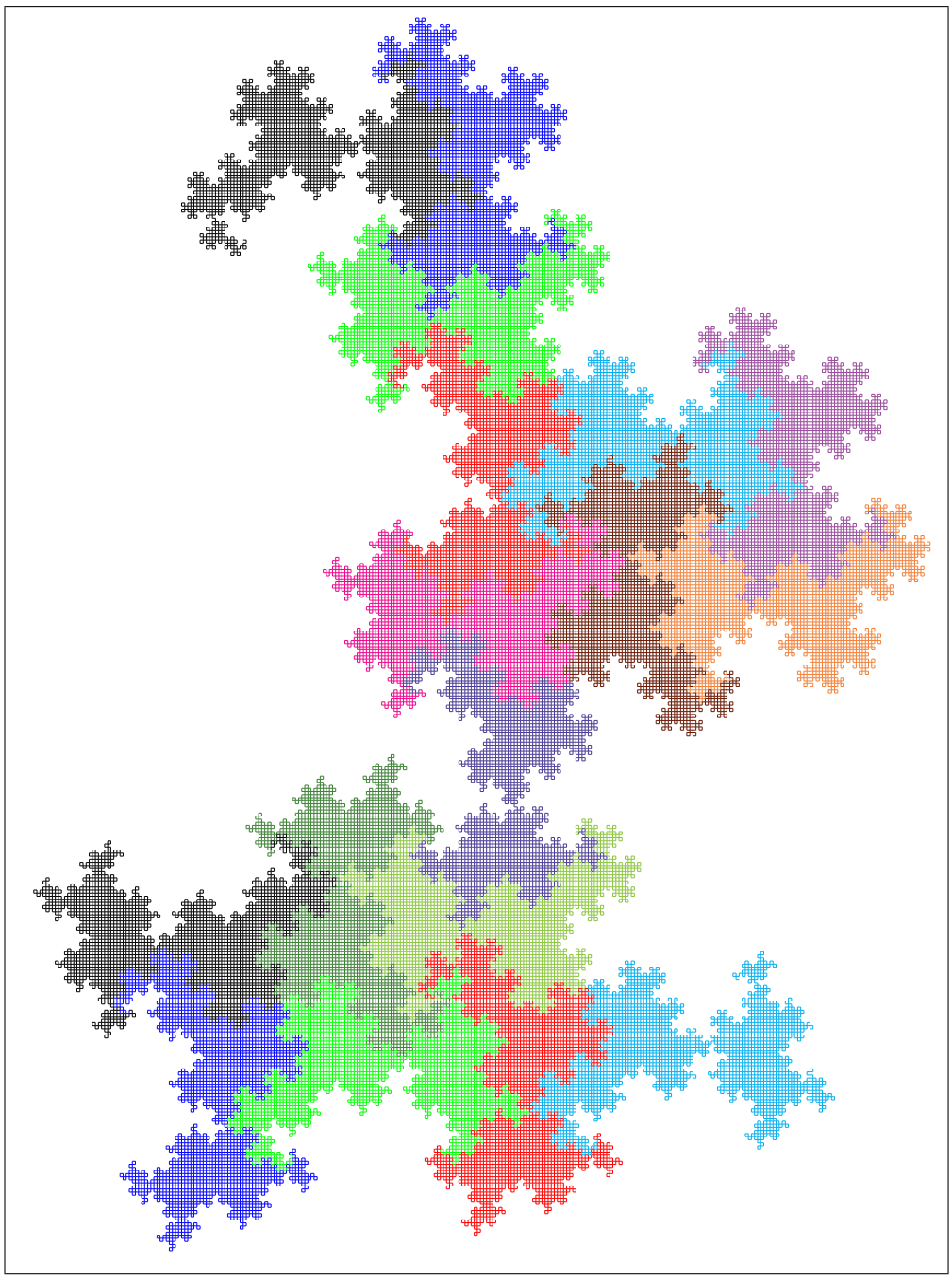

Figure 6.2-C: `R17-3` on the square grid.

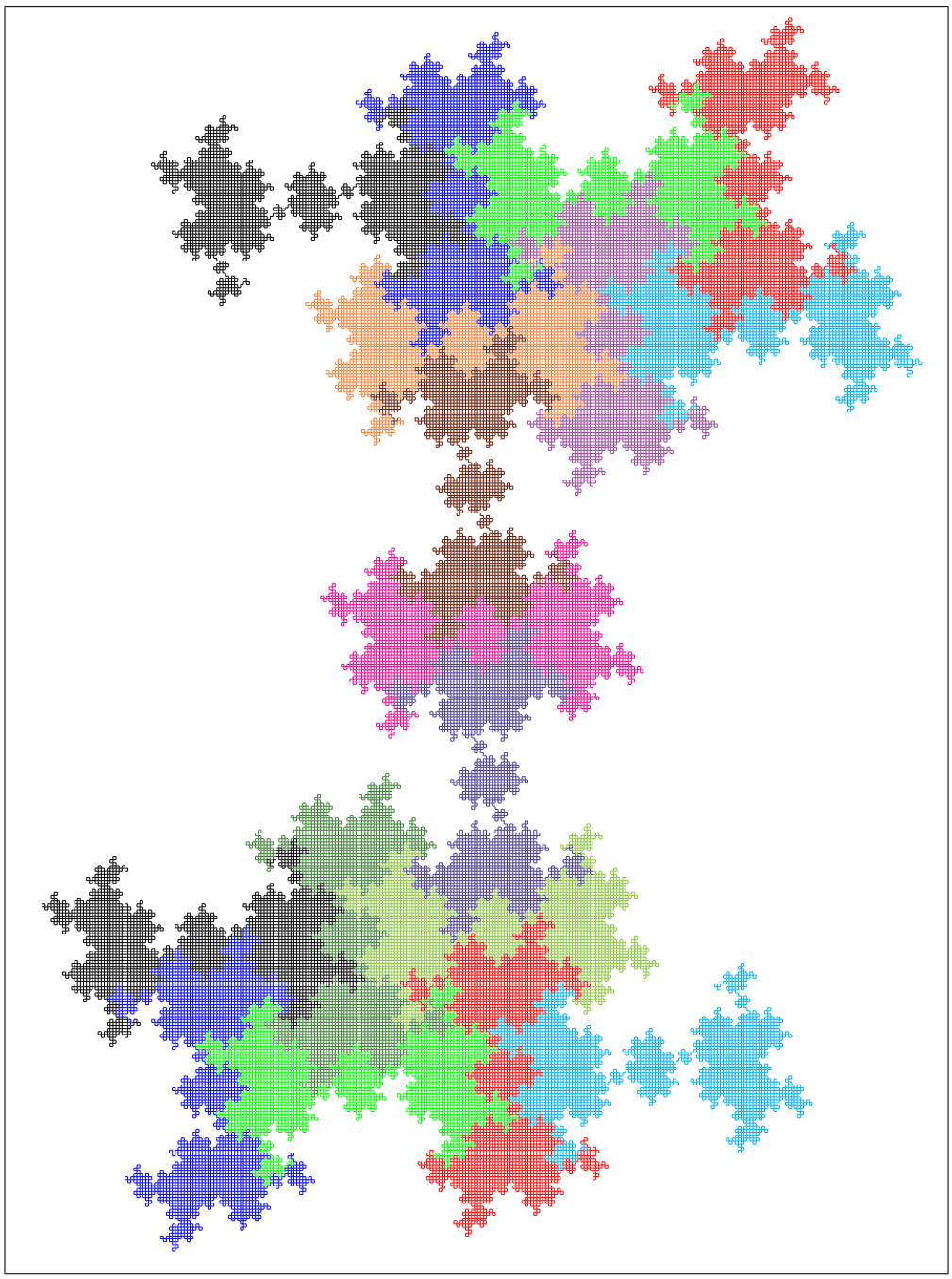

Figure 6.2-D: `R17-6` on the square grid.

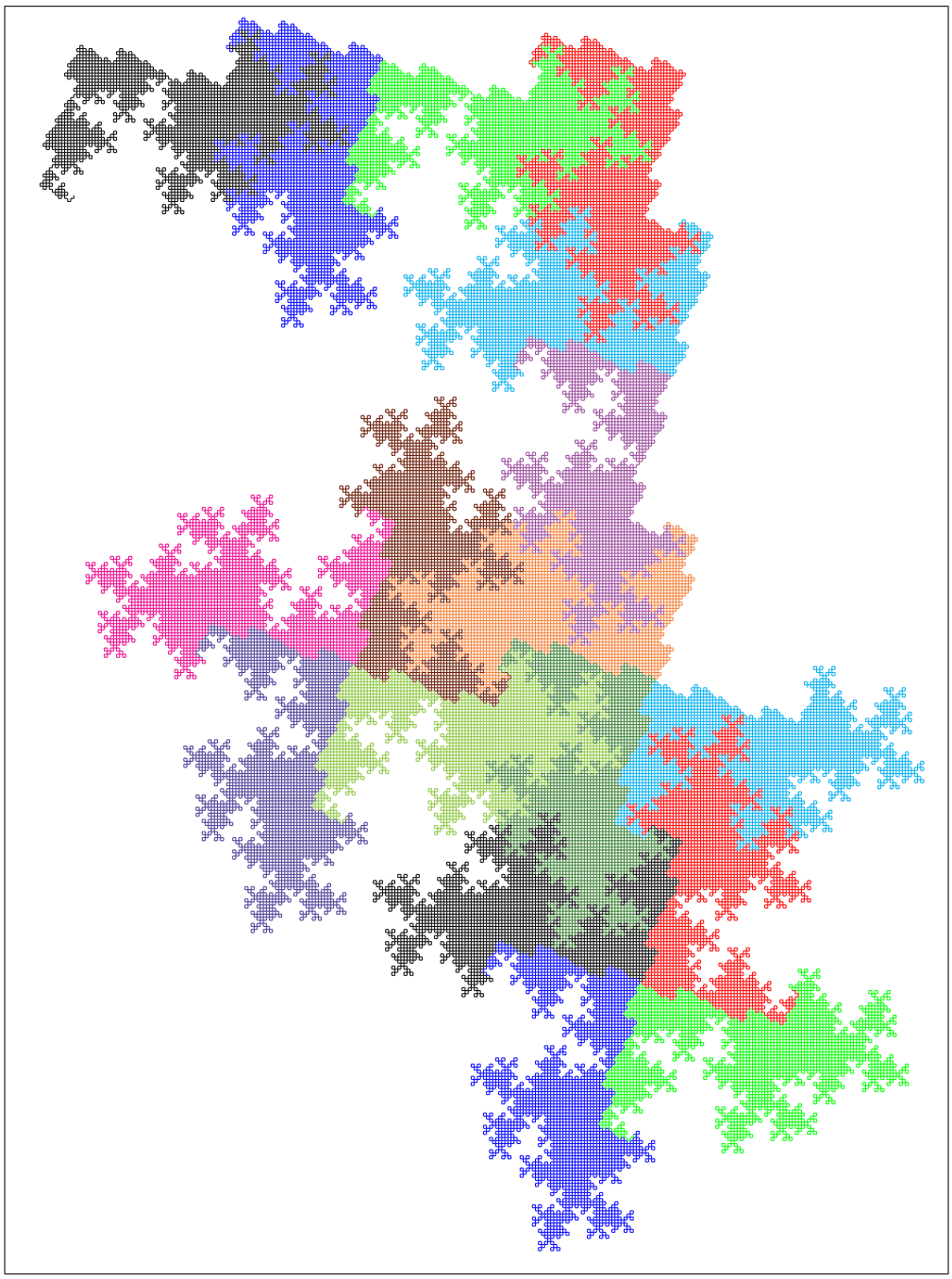

Figure 6.2-E: `R17-10` on the square grid.

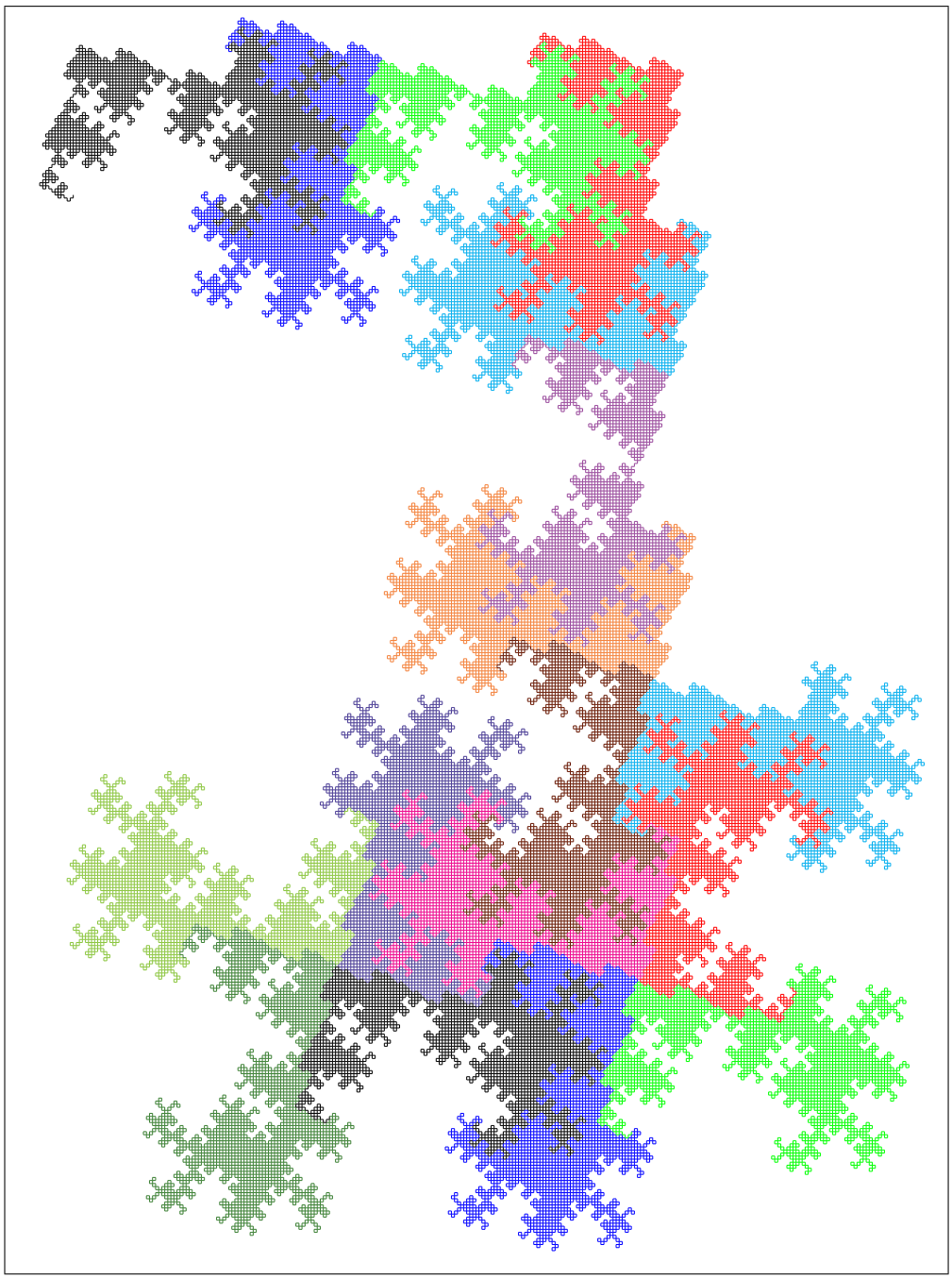

Figure 6.2-F: `R17-11` on the square grid.

6.3 Tri-Hexagonal grid

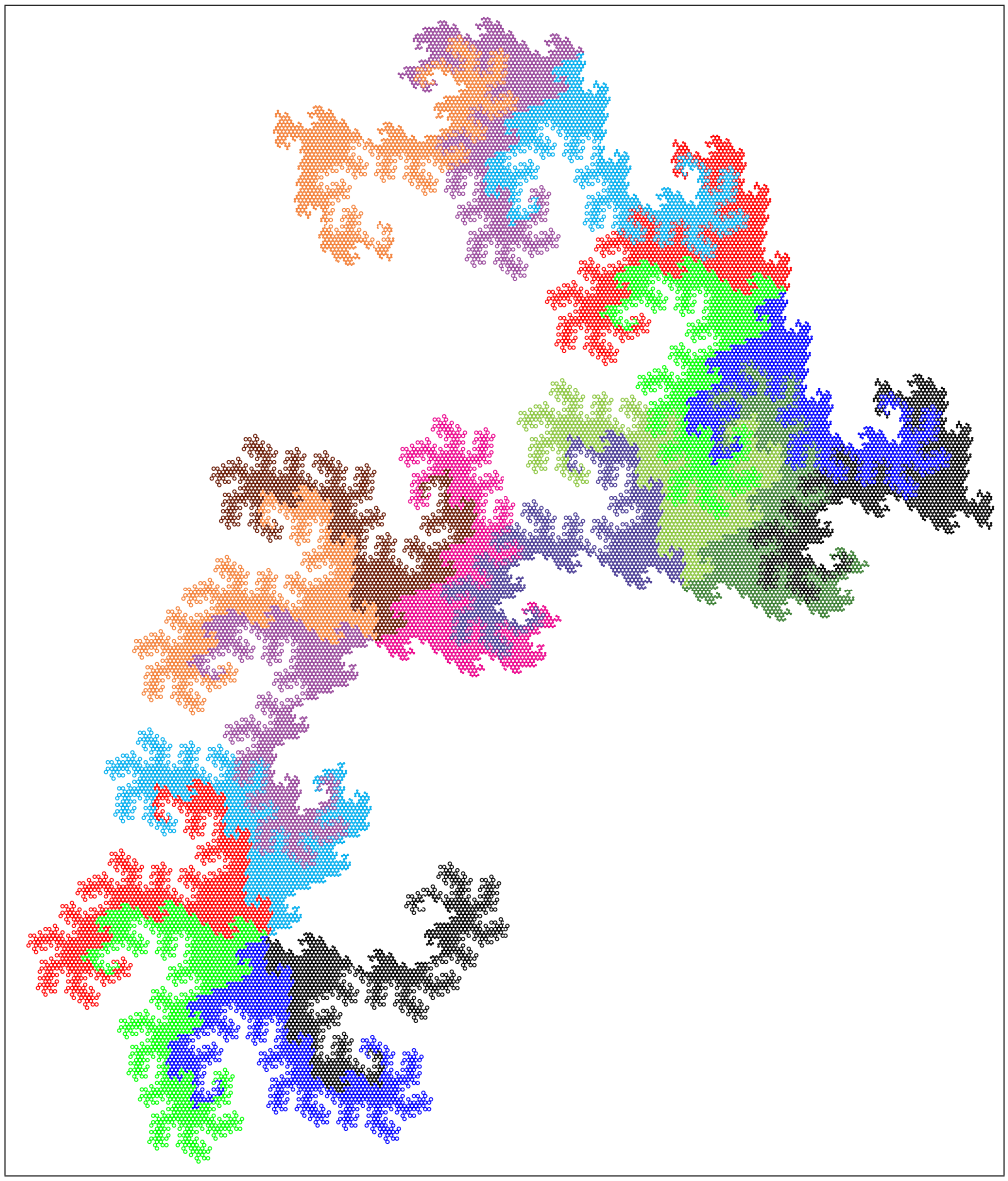

Figure 6.3-A: `R19-1` on the tri-hexagonal grid.

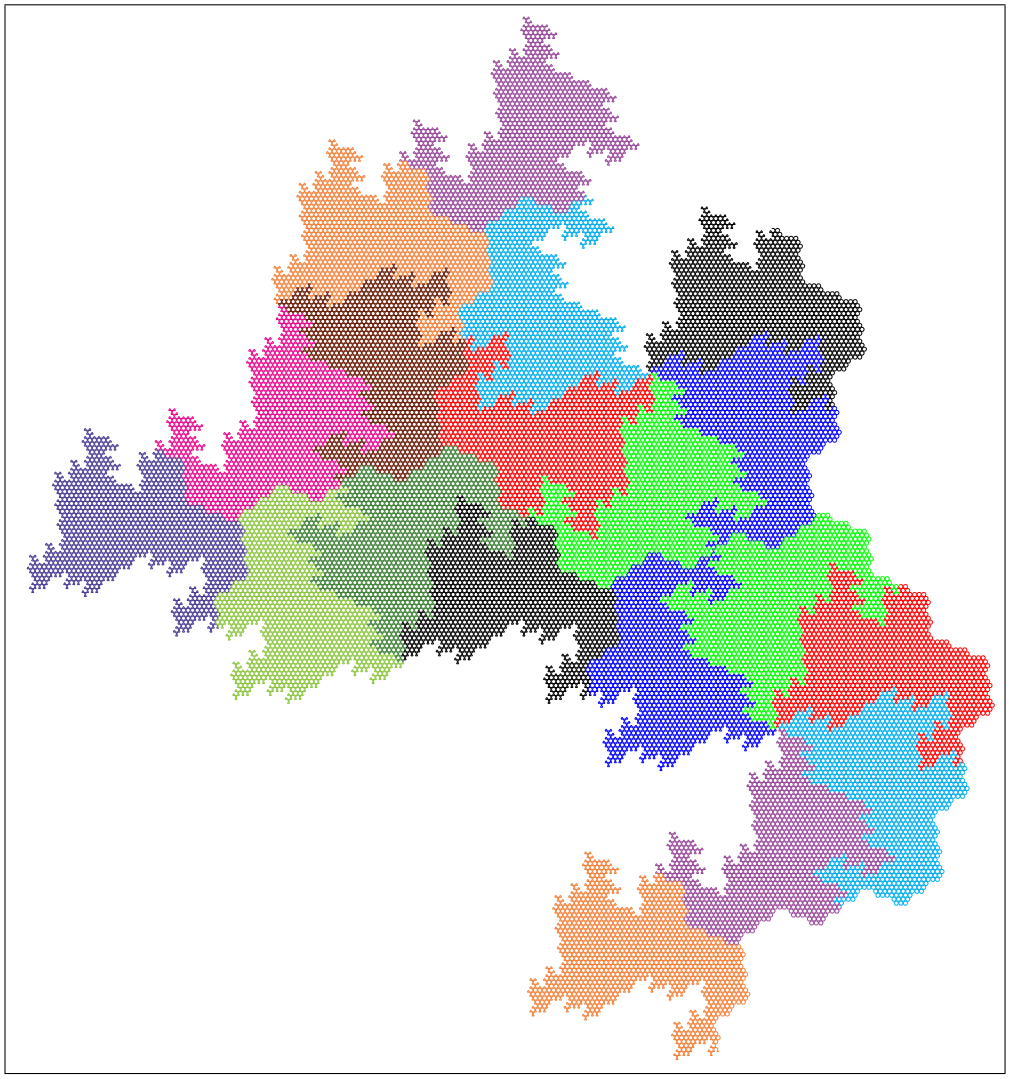

Figure 6.3-B: `R19-2` on the tri-hexagonal grid.

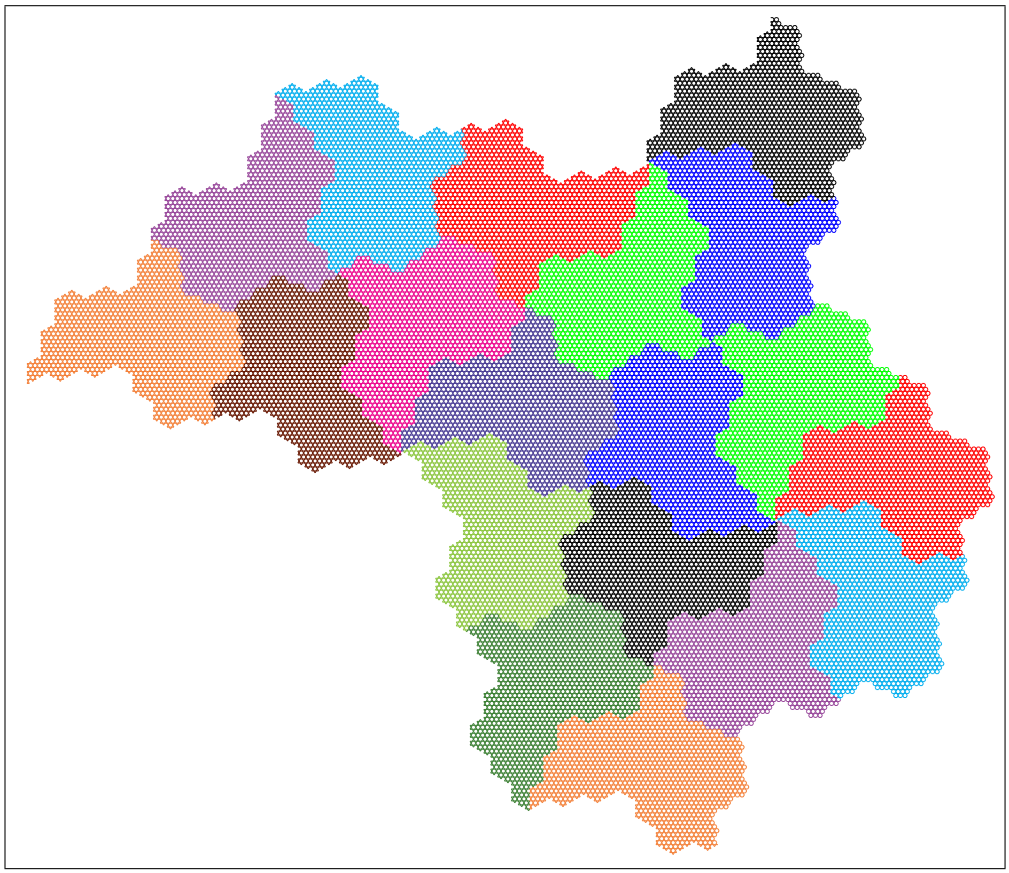

Figure 6.3-C: `R19-3` on the tri-hexagonal grid.

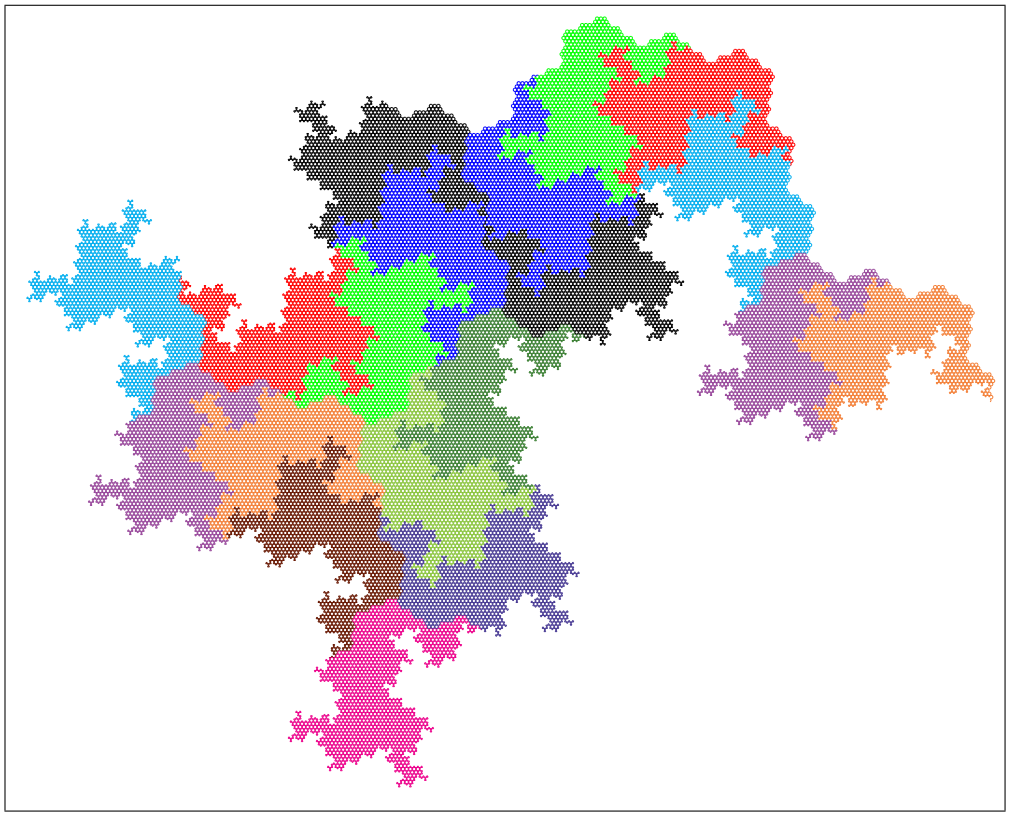

Figure 6.3-D: `R19-5` on the tri-hexagonal grid.

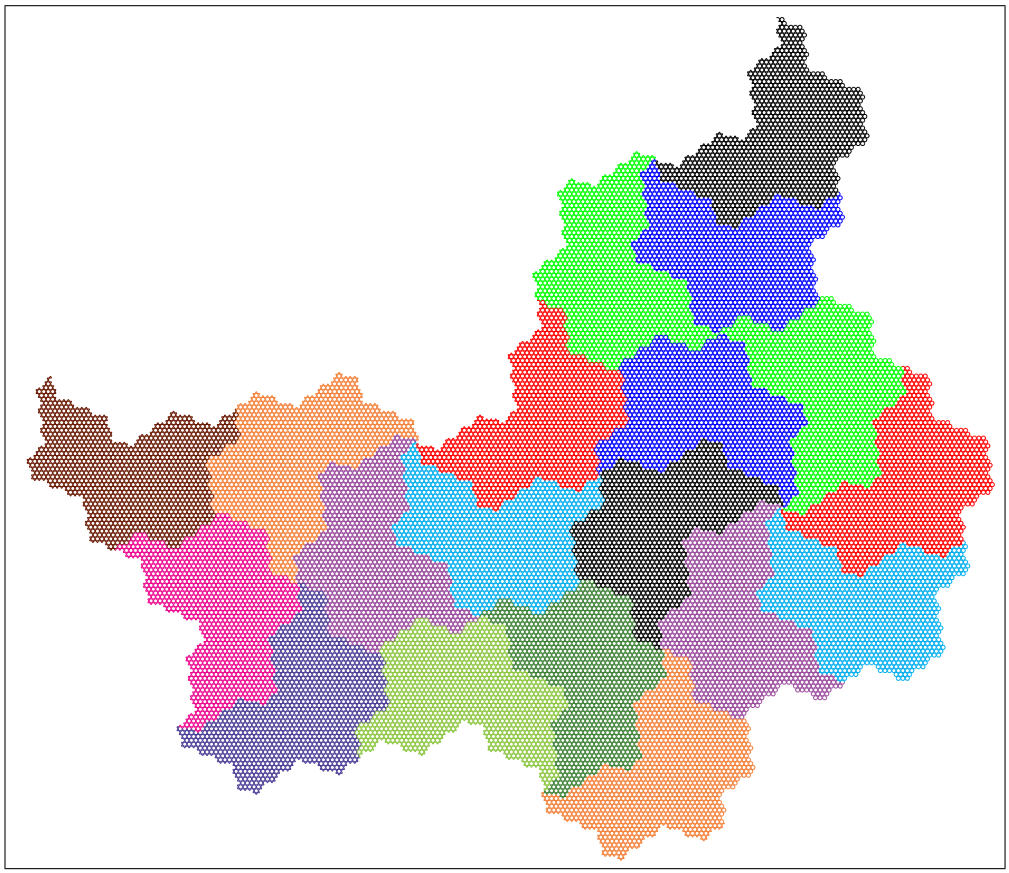

Figure 6.3-E: `R19-6` on the tri-hexagonal grid.

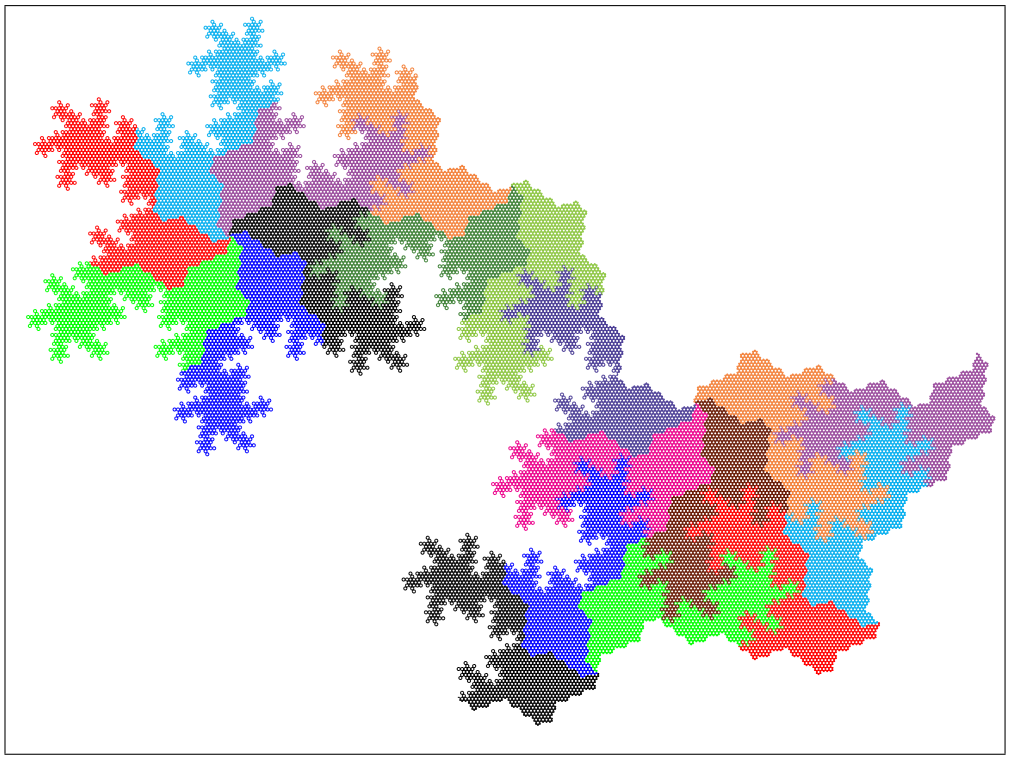

Figure 6.3-F: `R19-7` on the tri-hexagonal grid.

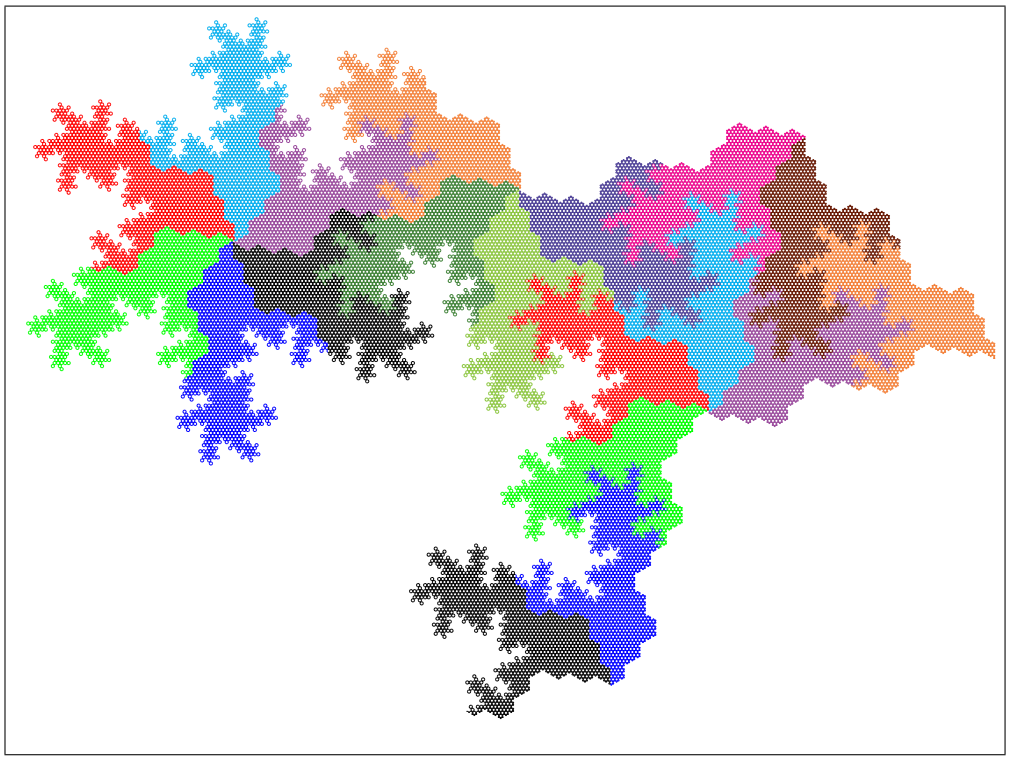

Figure 6.3-G: `R19-14` on the tri-hexagonal grid.

Acknowledgments

It is my pleasure to thank the following people for their support, criticism, corrections, and improvements.

Michael Beeler, Michel Dekking, Brian Galebach, Bill Gosper, Julia Handl, Michael Lontke, Tarkan Nein, Edith Parzefall, Andreas Pazureck, James Propp, Kevin Ryde, Jeffrey Shallit, Marcus Veitenthal, and Allan Wechsler.

References

[1] Jin Akiyama, Hiroshi Fukuda, Hiro Ito, Gisaku Nakamura: **Infinite Series of Generalized Gosper Space Filling Curves**, Lecture Notes in Computer Science, vol. 4381, pp. 1-9, (2007). 7

[2] Jean-Paul Allouche, Jeffrey Shallit: **Automatic Sequences**, Cambridge University Press, (2003). Errata and addenda online at URL: http://www.cs.uwaterloo.ca/~shallit/asas.html. 4

[3] Jörg Arndt: **Matters computational – Ideas, Algorithms, Source Code**, Springer-Verlag, (2011). URL: http://www.jjj.de/fxt/#fxtbook. 8, 9, 12

[4] Michael Bader: **Space-Filling Curves**, Springer-Verlag, (2013). 9

[5] A. J. Cole: **A Note on Peano Polygons and Gray Codes**, International Journal of Computer Mathematics, vol. 18, no. 1, pp. 3-13, (1985). 7

[6] Chandler Davis, Donald E. Knuth: **Number representations and dragon curves, I and II**, Journal for Recreational Mathematics, vol. 3, pp. 61-81 and pp. 133-149, (1970). 4

[7] Chandler Davis, Donald E. Knuth: **Number representations and dragon curves**, in: Selected Papers on Fun and Games, CSLI Publications, pp. 571-614, (2011). 4, 6, 19, 49

[8] F. M. Dekking: **Iterated paperfolding and planefilling curves**, Report 8126 of the Mathematical Institute, Katholieke Universiteit, Nijmegen, The Netherlands, pp. 1-16, (August-1981). 8

[9] Michel Dekking, M. Mendes France, A. J. van der Poorten: **Folds!**, The Mathematical Intelligencer, vol. 4, no. 4, pp. 190-195, (December-1982). 8

[10] Michel Dekking: **Paperfolding morphisms, planefilling curves, and fractal tiles**, Theoretical Computer Science, vol. 414, no. 1, pp. 20-37, (January-2012). URL: http://arxiv.org/abs/1011.5788 (arXiv preprint). 9, 10, 11, 19, 47

[11] Martin Gardner: **In which "monster" curves force redefinition of the word "curve"**, Scientific American, Mathematical Games, (December-1976). 7

[12] Branko Grünbaum, Geoffrey C. Shephard: **Tilings and Patterns**, Freemann, NY, (1986). 29

[13] Gilbert Helmberg: **Getting Acquainted with Fractals**, Walter de Gruyter, (2007). 8

[14] Gilbert Helmberg: **The Crab: A Connected Fractile of Infinite Connectivity**, Fractals, vol. 19, no. 03, pp. 367-377, (2011). 24

[15] David Hilbert: **Ueber die stetige Abbildung einer Linie auf ein Flächenstück**, Mathematische Annalen, vol. 38, pp. 459-460, (1891). 7

[16] Donald E. Knuth: **The Art of Computer Programming**, third edition, Volume 2: Seminumerical Algorithms, Addison-Wesley, (1997). Online errata list at http://www-cs-staff.stanford.edu/~knuth/. 23

[17] M. Lothaire: **Algebraic combinatorics on words**, Cambridge University Press, (2002). URL: http://www-igm.univ-mlv.fr/~berstel/Lothaire/AlgCWContents.html. 22

[18] Benoit B. Mandelbrot: **The Fractal Geometry of Nature**, W. H. Freeman, (1982). 7

[19] Douglas M. McKenna: **SquaRecurves, E-Tours, Eddies, and Frenzies: Basic Families of Peano Curves on the Square Grid**, in Richard K. Guy, Robert E. Woodrow, The Lighter Side of Mathematics: Proceedings of the Eugene Strens Memorial Conference on Recreational Mathematics and its History, Mathematical Association of America, pp. 49-73, (1994). 7

[20] Giuseppe Peano: **Sur une courbe, qui remplit toute une aire plane**, Mathematische Annalen, vol. 36, no. 1, (March-1890). 7

[21] Heinz-Otto Peitgen, Dietmar Saupe (eds.): **The Science of Fractal Images**, Springer-Verlag, (1988). 11

[22] P. Prusinkiewicz, A. Lindenmayer: **The Algorithmic Beauty of Plants**, Springer-Verlag, (1990). URL: http://algorithmicbotany.org/papers/#abop. 3

[23] Hans Sagan: **Space-Filling Curves**, Springer-Verlag, (1994). 7, 37

[24] N. J. A. Sloane: **The On-Line Encyclopedia of Integer Sequences**, (1964-2015). URL: http://oeis.org/?blank=1. 11, 14, 27, 31

[25] Jeffrey J. Ventrella: **Brain-Filling Curves: A Fractal Bestiary**, LuLu.com, (2012). URL: http://archive.org/download/BrainfillingCurves-AFractalBestiary/. 8, 49, 50, 51

[26] Wikipedia: **List of convex uniform tilings**, Wikipedia article, (September-2015). URL: https://en.wikipedia.org/wiki/List_of_convex_uniform_tilings. 29

[27] Wikipedia: **Euclidean tilings by convex regular polygons**, Wikipedia article, (September-2015). URL: https://en.wikipedia.org/wiki/Euclidean_tilings_by_convex_regular_polygons. 29