A New Method of Constructing Fractals and Other Graphics

Janet Helmstedt

In this article we demonstrate a method of constructing various types of graphics from polygonal arcs using replacement rules for vertices. The article is based on a presentation entitled "Constructing Graphics from Polygonal Arcs Using Replacement Rules for Vertices" given by the author at the Wolfram Technology Conference in Champaign, Illinois in October 2006.

■ Introduction

The simplest type of Lindenmeyer or *L*-system can be used to construct graphics as follows. Two polygonal arcs `A1` and `A2` are chosen, such that the length of each line segment is an integral multiple of a fixed positive number, λ, and if a line segment has length $n\,\lambda$, then it is treated as a polygonal arc consisting of *n* line segments of equal length. Also, the angle between each pair of adjacent line segments is an integral multiple of a fixed angle, δ. `A1` is usually chosen as a single line segment or as the boundary of a regular polygon. Each line segment of `A1` is replaced by a copy of `A2`, and then each line segment of the resulting polygonal arc is replaced by a copy of `A2`, and so on. The constructions are achieved by interpreting certain replacement rules for sequences as replacement rules for line segments [1].

We present a different method of constructing graphics of the above type that has the advantage that there are no restrictions on the lengths of the line segments of `A1` and `A2`, nor on the angles between adjacent line segments. Also, different replacement rules may be applied to the different line segments of `A1`, and the code used is straightforward.

More complicated forms of *L*-systems can be used to construct space-filling curves and plant-like forms [1]. The method described in this article does not lend itself to such constructions. Applications of even more complicated forms of *L*-systems called parametric *L*-systems, which place no restrictions on lengths or angles of polygonal arcs, can be found in [2].

We make simple changes to the given programs in order to generate other interesting graphics.

■ A Method of Constructing a Polygonal Arc

A *polygonal arc* is a segmented line connecting a finite sequence of points in the plane. The line may intersect itself and segments of the line may be repeated. The programs described here are based on a method of constructing a polygonal arc with initial point the origin.

A point P in the plane at a distance r from the origin can be represented by a complex number of the form $r \exp(i\,\theta) = \exp(i(\theta - i \log(r)))$, θ real, $r > 0$.

A second point Q at a distance ρ from P can be represented by a complex number of the form $r \exp(i\,\theta) + \rho \exp(i\,(\theta + \phi))$, and so on.

To construct a polygonal arc, A, with vertices represented by the sequence S of complex numbers $\{0, r \exp(i\,\theta), \; r \exp(i\,\theta) + \rho \exp(i(\theta + \phi)), \ldots\}$, we apply operations on lists to the sequence $E = \{\theta - i \log(r), \; \theta + \phi - i \log(\rho), \ldots\}$.

We then calculate the coordinates of the points and join them in order with `Line` or `ListPlot`. With this method of construction, the polygonal arc is determined by the sequence E.

We can then apply replacement rules to the sequence E, and hence to A itself.

For example, we construct the polygonal arc with initial point the origin and other vertices determined by the sequence $\{\theta - i \log(1.5), \theta - 2\,\phi, \theta - 3.5\,\phi, \theta, \theta + 3.5\,\phi, \theta + 2\,\phi, \theta - i \log(1.5)\}$, where θ and ϕ are arbitrary.

```
ListLinePlot[
  {Re[#], Im[#]} & /@
   (FoldList[Plus, 0,
     Exp[
       I ({θ - I Log[1.5], θ - 2 φ, θ - 3.5 φ, θ, θ + 3.5 φ,
          θ + 2 φ, θ - I Log[1.5]} /. {θ → 0, φ → π / 5.4})]]),
  PlotStyle → {Black, Thick}]
```
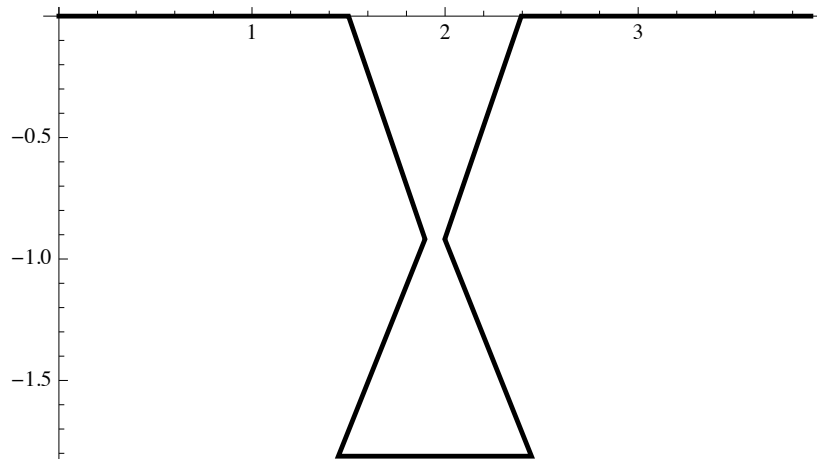
## ■ Applying Replacement Rules to a Line Segment

Consider the function `rA`.

```
rA[θ0_, ϕ0_, n_] := Module[{θ, ϕ},
  Nest[Function[θ, Flatten[
      {θ - I Log[1.5], θ - 2 ϕ, θ - 3.5 ϕ, θ, θ + 3.5 ϕ, θ + 2 ϕ,
       θ - I Log[1.5]}
    ]], N[θ0], n] /. ϕ → N[ϕ0]
]
```

The sequence $\{\theta\}$ determines a polygonal arc consisting of a single line segment, L, starting at the origin. The list inside `rA` determines a replacement rule for L.

The resulting arc, t, is called the initial arc. We apply a nesting process in `rA`, choose values for θ and ϕ, and interpret the resulting sequence as a polygonal arc. The nesting process replaces L by a (possibly scaled) copy of t, and then each segment of t by a copy of itself, and so on. The function `arcRep` can be used to apply the replacement rule determined by `r` repeatedly to a line segment. Also `opts` can be used to determine a graphics option if needed.

```
arcRep[r_, opts___] :=
 ListLinePlot[{Re[#], Im[#]} & /@ (FoldList[Plus, 0, E^(I r)]),
   opts, PlotStyle → Black]
```

```
arcRep[rA[0, π / 5.4, 2]]
```
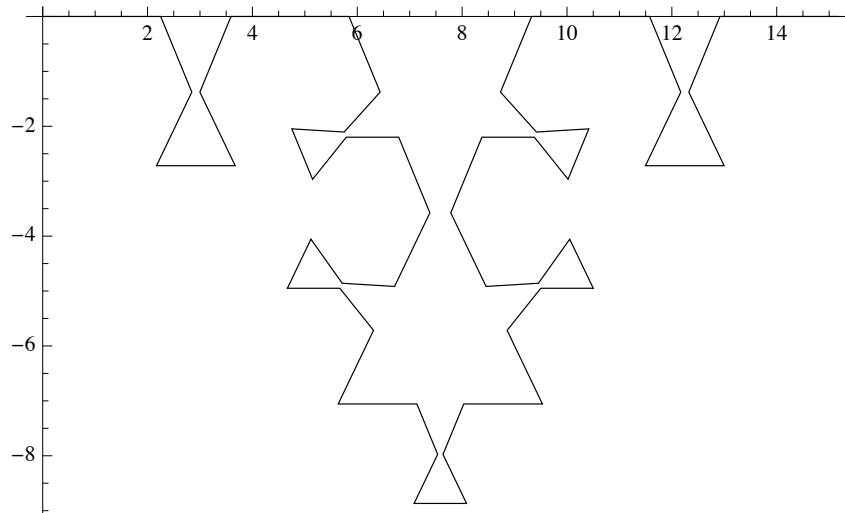

If the zero in the definition (the second argument of `FoldList`) is replaced by `a + b I`, the starting point of the arc becomes (a, b).
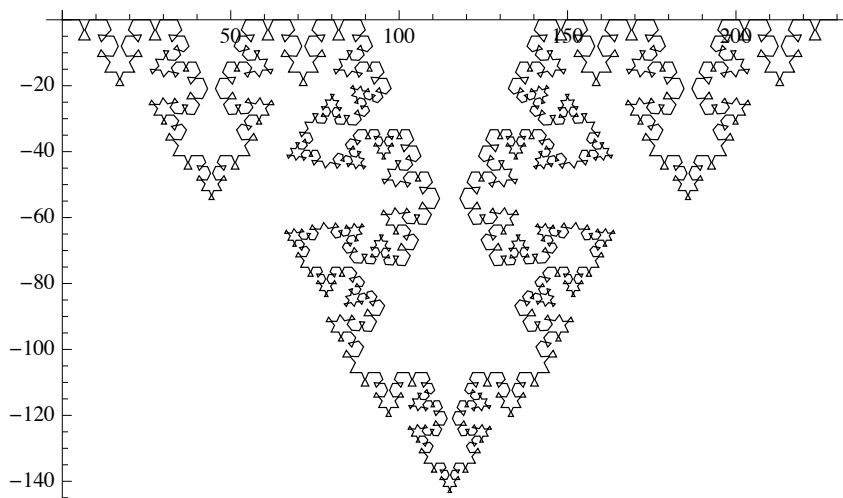
```
arcRep0[r_, opts___] :=
 ListLinePlot[{Re[#], Im[#]} & /@
   (FoldList[Plus, 2 + 3 I, E^(I r)]), opts, PlotStyle → Black]
```

```
arcRep0[rA[0, π / 5.4, 2]]
```
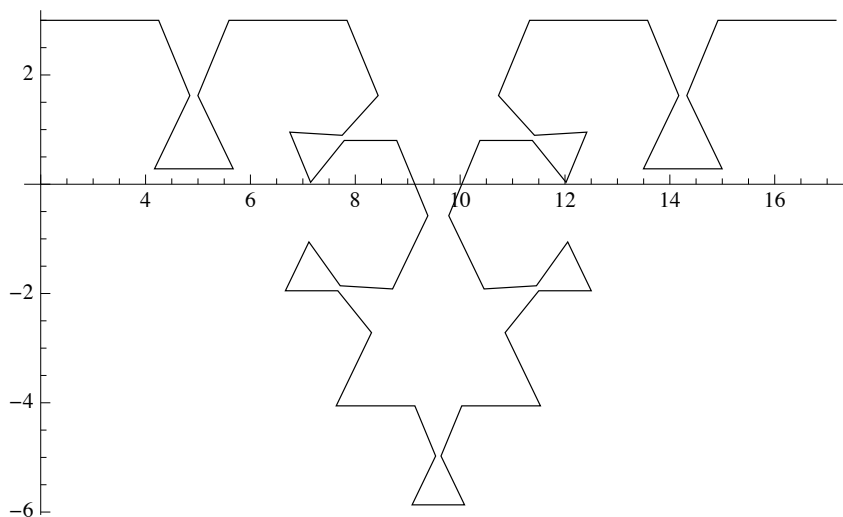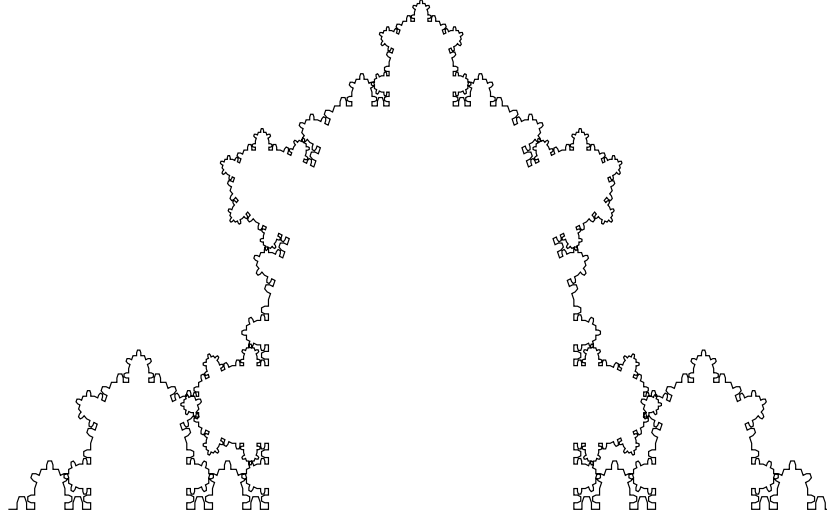

Here rA is nested four times.

```
arcRep[rA[0, π / 5.4, 4]]
```


Notice that the above graphic is covered by seven (approximately) similar copies of itself. In the first three graphics, as can be seen from the axes, the larger the number of iterations, the larger the graphic in real terms. From now on, except in one case, we omit axes.

Here $\phi = \pi / 1.06$.

```
arcRep[rA[0, -5 π / 4, 4], Axes → False]
```


In the rest of this article, we will construct a dozen or so variations of the function `rA` by changing the list involving θ and ϕ. To automate this, we define the function `defineRule`. Its first argument is the function name (like `rA`) and its second argument is a list.

```
defineRule =
  ToExpression[
    StringReplace[
      "Clear[rX];rX[θ0_,φ0_,n_]:=Module[{θ,φ},Nest[Function[
        θ,Flatten[
        θφlist
      ]],N[θ0],n]/.φ→N[φ0]]",
      {"rX" → ToString[#1],
       "θφlist" → ToString[FullForm[#2]]}] ] &;
```

 In the following example, we use a nonsymmetric rewriting rule.

```
defineRule[rB, {2 θ, θ + φ, θ, θ - φ, θ, θ}]
```

As before with `rA`, this recovers the list.

```
rB[θ, φ, 1]
```

{2 θ, θ + φ, θ, θ - φ, θ, θ}

We remark that the entry 2θ in the replacement rule `rB` has the effect of doubling angles.
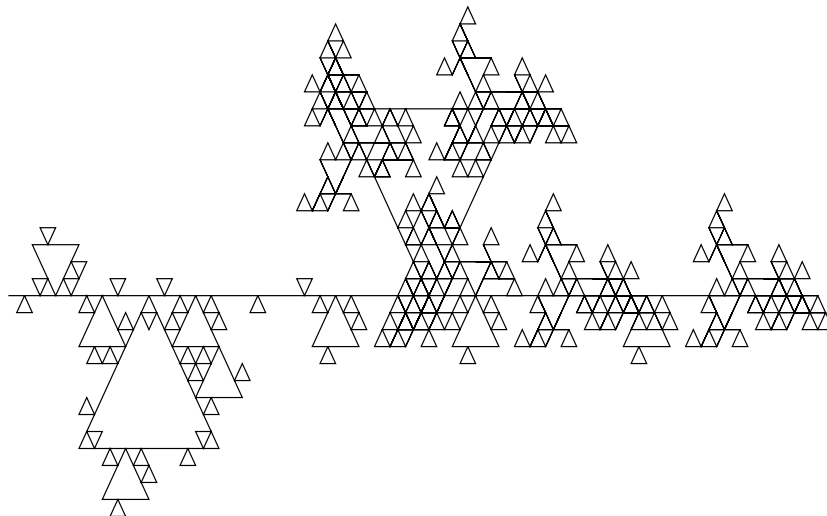
Here $\phi = 2\pi/3$.

```
arcRep[rB[0, 2 π / 3, 4], Axes → False]
```
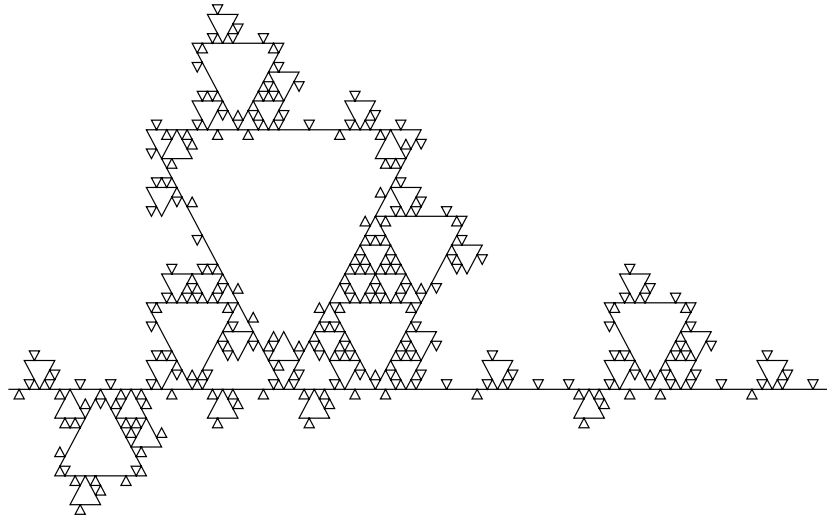

In the case $\theta = \pi$, there is no rotation.

```
arcRep[rB[π, 2 π / 3, 4], Axes → False]
```


A polygonal arc constructed by the above method may be rotated, scaled, and in some cases, reflected, using variations of its defining replacement rule.

```
defineRule[rC, {θ, θ + 2.5 ϕ, θ - ϕ, θ}]

p = Style["P", Italic];
Grid[{
  {
   arcRep[rC[0, 4 π / 15, 3],
    PlotLabel → Text@Row[{"polygonal arc ", p}]],
   arcRep[rC[0, 4 π / 15, 3] - π / 6,
    PlotLabel → Text@Row[{p, " rotated by ", -π / 6}]]
  },
  {
   arcRep[-rC[0, 4 π / 15, 3],
    PlotLabel →
     Text@Row[{p, " reflected in the real axis"}]],
   arcRep[rC[0, 4 π / 15, 3] - I Log[2],
    PlotLabel → Text@Row[{p, " scaled by 2"}]]
  },
  {
   arcRep[Reverse[rC[0, 4 π / 15, 3]],
    PlotLabel →
     Text@Row[{"order of segments of ", p, " reversed"}]],
   arcRep[-Reverse[rC[0, 4 π / 15, 3]],
    PlotLabel →
     Text@
      Row[{"     ", p,
        " reversed and then \n reflected in the real
         axis"}]]
  }
}]
```

polygonal arc *P*

P rotated by $-\frac{\pi}{6}$

P reflected in the real axis

P scaled by 2

order of segments of *P* reversed

P reversed and then
reflected in the real axis
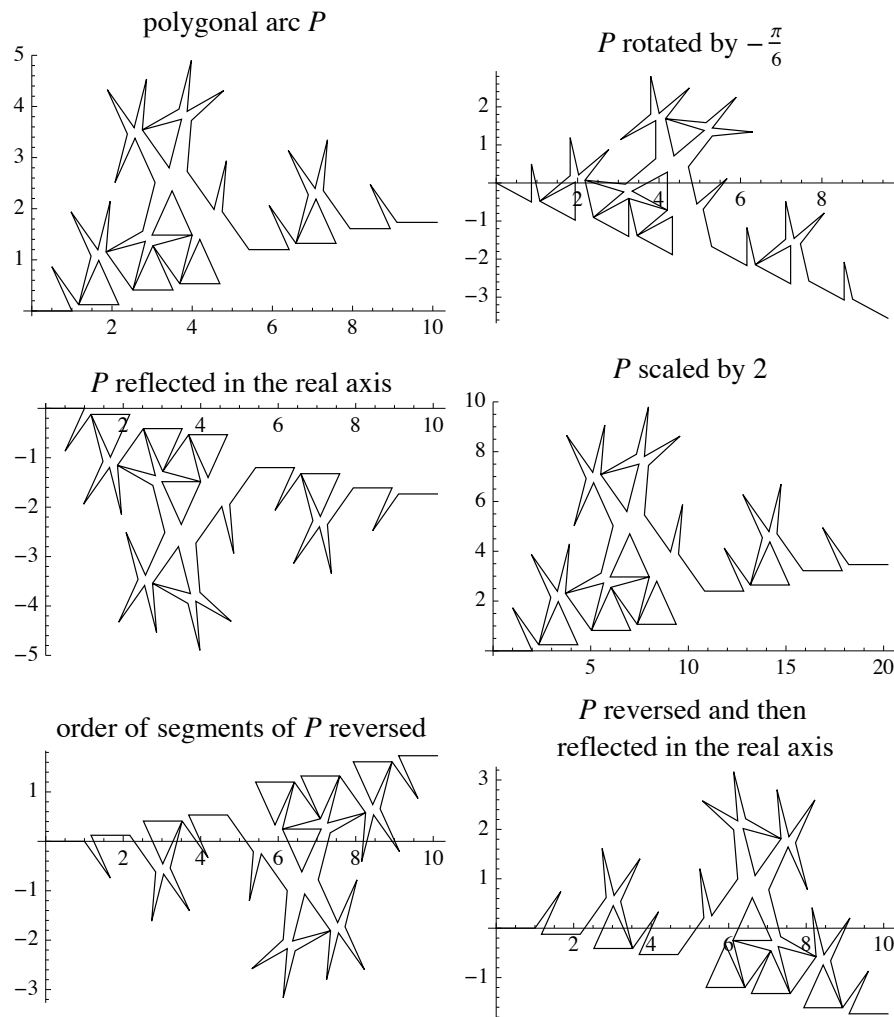
A reflection will not be obtained unless the entries in the replacement rule are all real. However, in such a case, a different replacement rule may be constructed to effect a reflection.

■ Applying Replacement Rules to the Segments of a Polygonal Arc

We use `Join` to concatenate a list of replacement rules and so concatenate the resulting polygonal arcs. The function `arcJoin` defined here can be used to construct a sequence of concatenated polygonal arcs when `x` is replaced by a sequence of replacement rules.

```
arcJoin[x__] :=
 ListLinePlot[{Re[#], Im[#]} & /@
    (FoldList[Plus, 0, E^(I Join[x])]),
   AspectRatio → Automatic, Axes → False, PlotStyle -> Black]
```

In the following example, we apply a replacement rule to the edges of a right isosceles triangle.

$$\texttt{defineRule}\Big[\texttt{rD}, \ \Big\{\theta, \ \theta + 2\,\phi, \ \theta + \sqrt{3}\,\phi, \ \theta, \ \theta - \sqrt{3}\,\phi, \ \theta - 2\,\phi, \ \theta\Big\}\Big]$$

```
arcJoin[
 rD[0, -19 π / 5, 4] - I Log[2] / 2,
 rD[0, -19 π / 5, 4] - 3 π / 4,
 rD[0, -19 π / 5, 4] - 3 π / 4 - π / 2
]
```
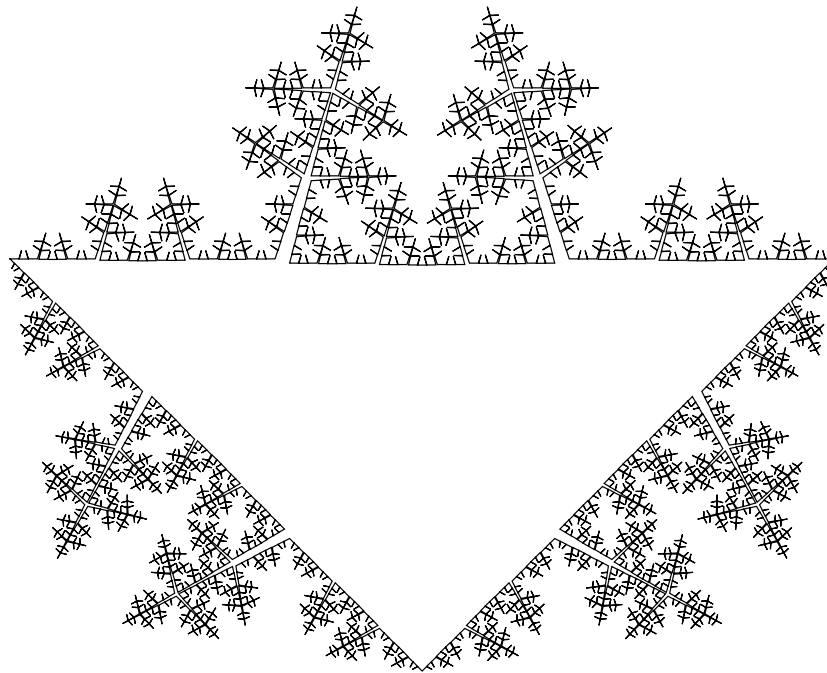
Here $\phi = 80\,\pi\,/\,13$.

```
arcJoin[
 rD[0, 80 π / 13, 4] - I Log[2] / 2,
 rD[0, 80 π / 13, 4] - 3 π / 4,
 rD[0, 80 π / 13, 4] - 3 π / 4 - π / 2
]
```
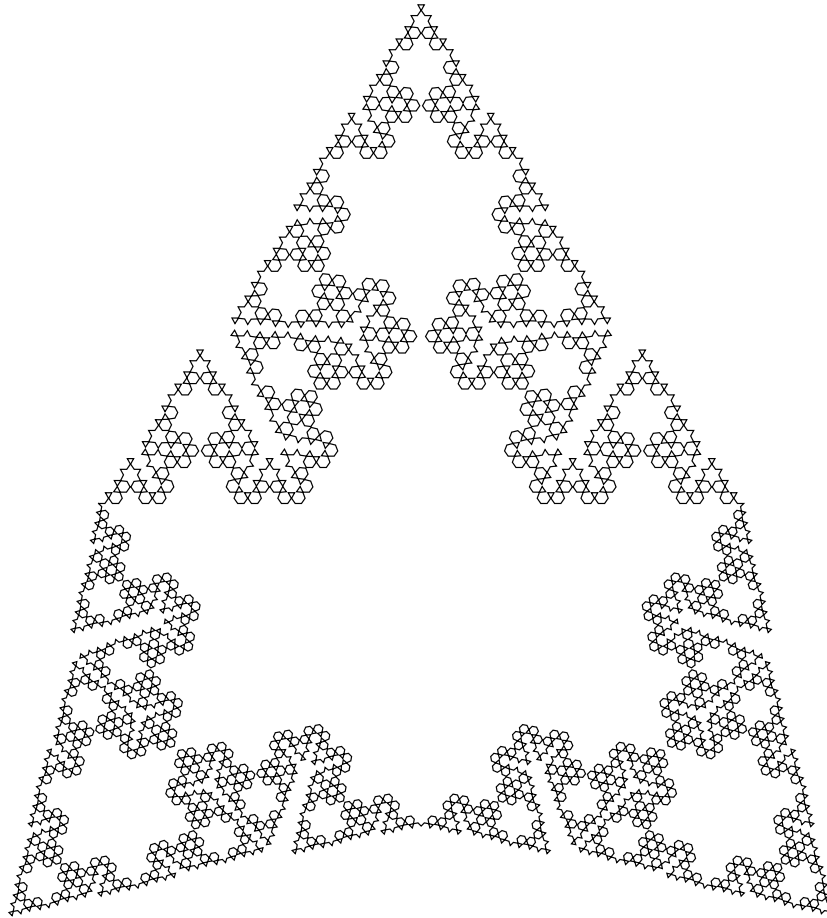

In the following example, we apply a replacement rule to each side of a line segment.

```
defineRule[rE, {θ + φ, θ + 3 φ, θ - 3 φ, θ}]
```
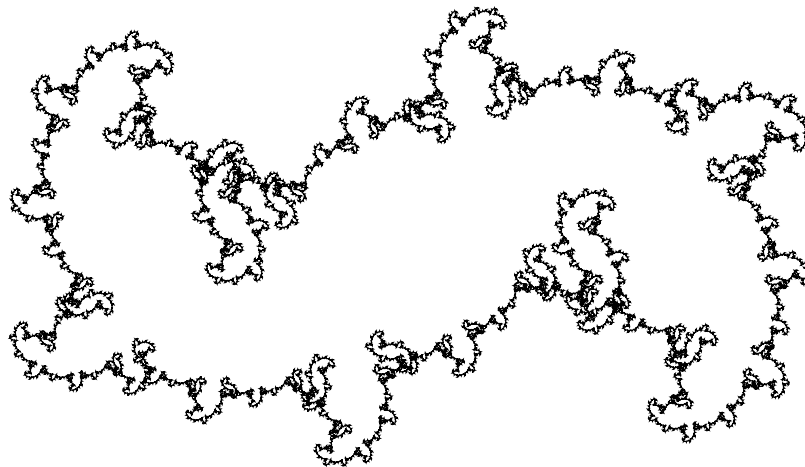
```
arcJoin[
 rE[π / 4, 3 π / 5, 7],
 rE[π / 4, 3 π / 5, 7] + π
]
```


Here $\phi = 5 \pi / 7$.

```
arcJoin[
 rE[π / 4, 5 π / 7, 7],
 rE[π / 4, 5 π / 7, 7] + π
]
```
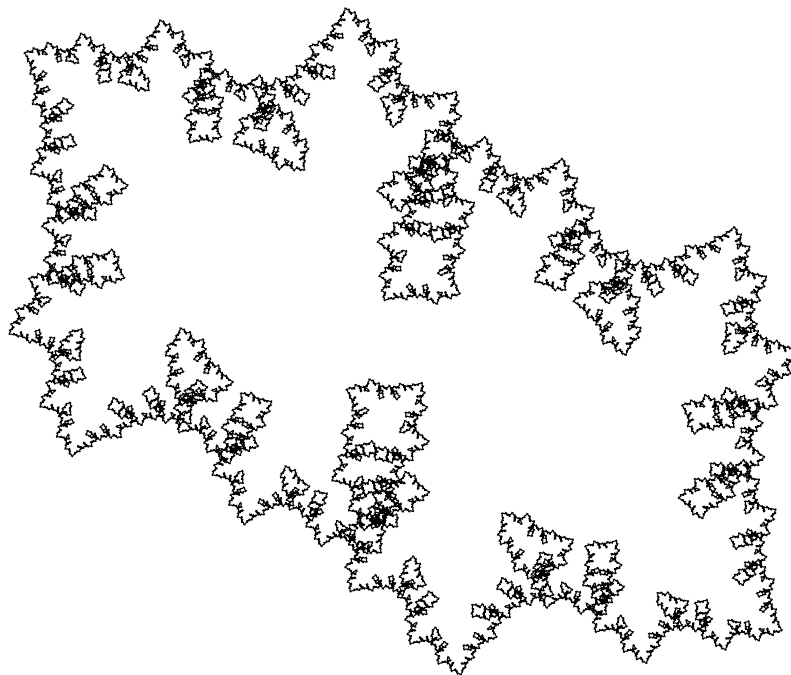
In the following example, we apply a replacement rule to each side of the edges of a square.

```
defineRule[rF, {θ, θ + φ, θ + 2 φ, θ - 3 φ, θ}]
```

```
arcJoin[
 -rF[0, 2 π / 3, 5],
 -rF[0, 2 π / 3, 5] + π / 2,
 -rF[0, 2 π / 3, 5] + π,
 -rF[0, 2 π / 3, 5] + 3 π / 2,
 rF[0, 2 π / 3, 5],
 rF[0, 2 π / 3, 5] + π / 2,
 rF[0, 2 π / 3, 5] + π,
 rF[0, 2 π / 3, 5] + 3 π / 2
]
```
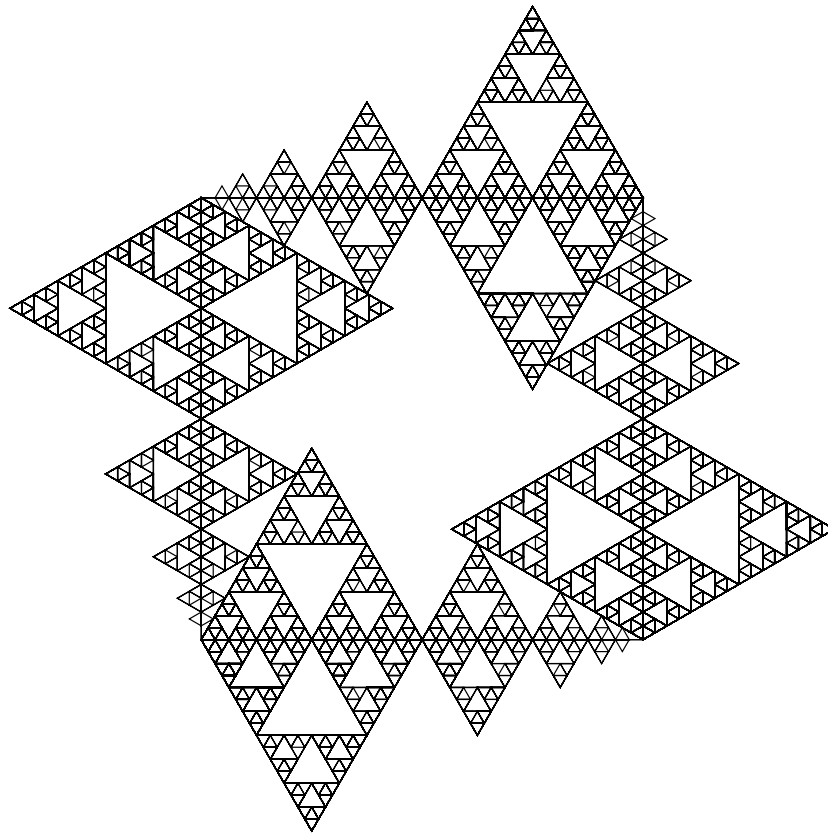

We apply the replacement rules rG and −rG alternately to the edges of a square.

```
defineRule[rG, {θ, θ - 4 φ, θ - φ, θ + φ, θ + 4 φ, θ}]
```

```
arcJoin[
  rG[0, -5 π / 8.9, 4],
  -rG[0, -5 π / 8.9, 4] + π/2,
  rG[0, -5 π / 8.9, 4] + π,
  -rG[0, -5 π / 8.9, 4] + 3 π / 2
]
```
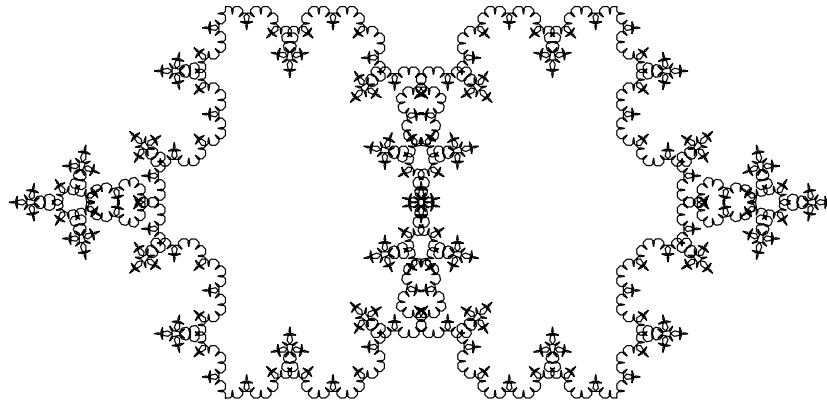

Here $\phi = 2\pi/13$.

```
arcJoin[
  -rG[0, 2 π / 13, 4],
  rG[0, 2 π / 13, 4] + π/2,
  -rG[0, 2 π / 13, 4] + π,
  rG[0, 2 π / 13, 4] + 3 π / 2
]
```
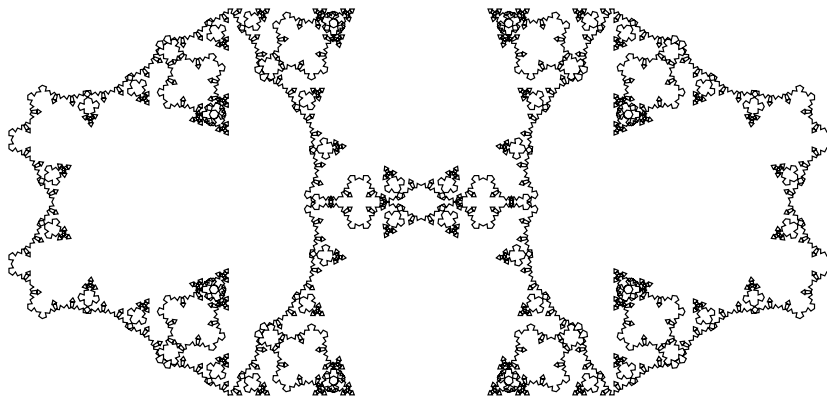
In the following example, we alternate `rH` with four and `- rH` with three iterations.

```
defineRule[rH, {θ, θ + ϕ, θ - 3 ϕ, θ + 3 ϕ, θ - ϕ, θ}]
```
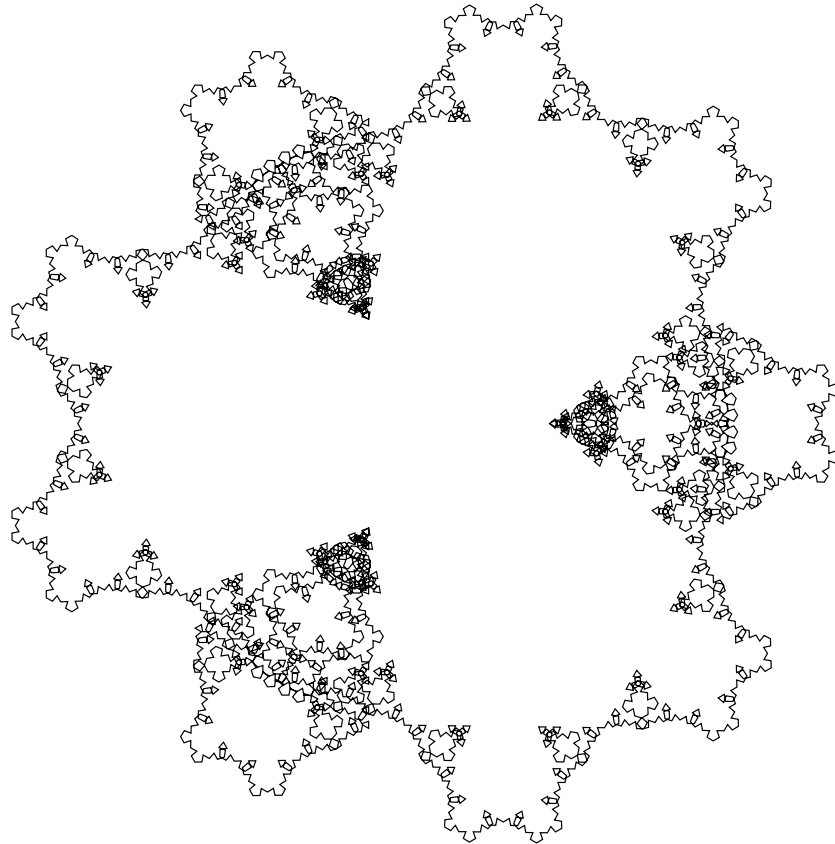
```
arcJoin[
 rH[π, 18 π / 25, 4],
 -rH[π, 18 π / 25, 3] + π / 3,
 rH[π, 18 π / 25, 4] + 2 π / 3,
 -rH[π, 18 π / 25, 3] + π,
 rH[π, 18 π / 25, 4] + 4 π / 3,
 -rH[π, 18 π / 25, 3] + 5 π / 3
]
```
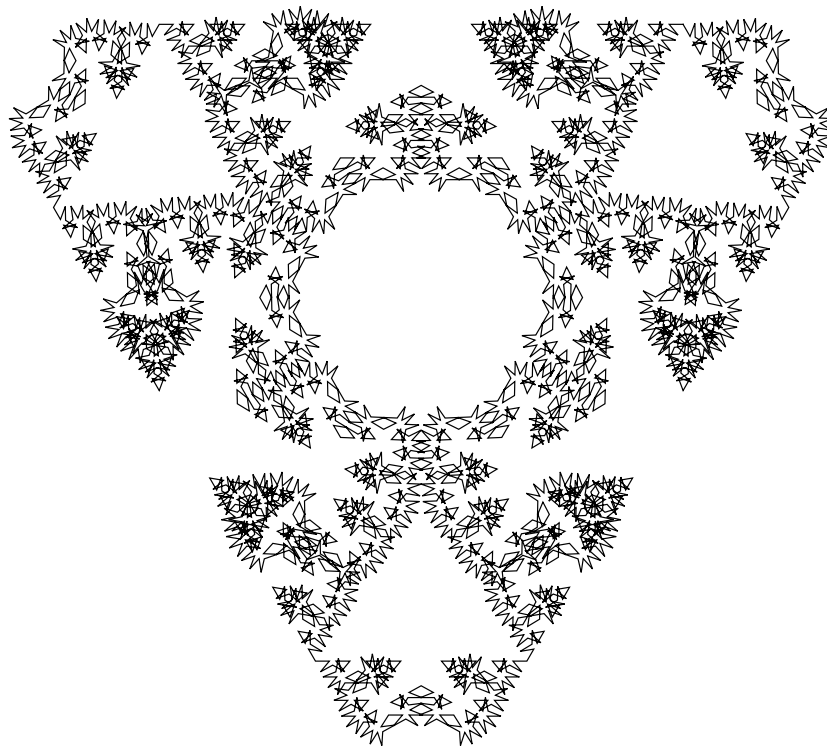
In the following example, $\theta = \pi/2$ and we alternate -rH with four and rH with three iterations.

```
arcJoin[
 -rH[π / 2, 5 π / 8.2, 4],
 rH[π / 2, 5 π / 8.2, 3] + π / 3,
 -rH[π / 2, 5 π / 8.2, 4] + 2 π / 3,
 rH[π / 2, 5 π / 8.2, 3] + π,
 -rH[π / 2, 5 π / 8.2, 4] + 4 π / 3,
 rH[π / 2, 5 π / 8.2, 3] + 5 π / 3
]
```
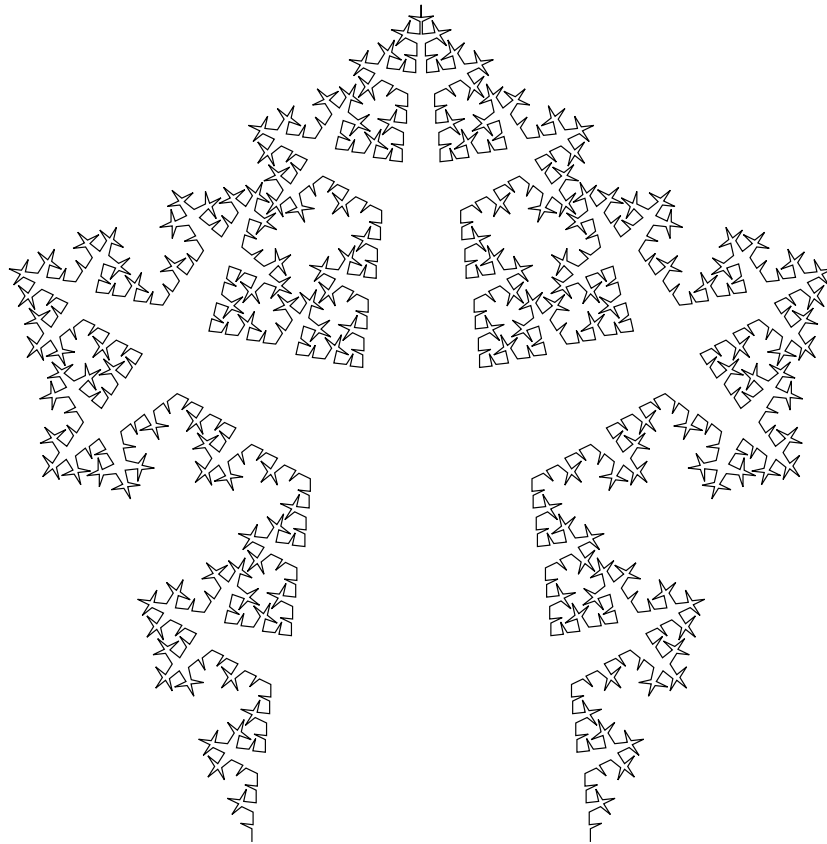
The following example illustrates the use of `Reverse` to obtain a symmetric image from a nonsymmetric replacement rule.

```
defineRule[rI, {θ, θ - 2 ϕ, θ + 3 ϕ, θ}]
```

```
arcJoin[
 -rI[-π / 2, π / 5.6, 5],
 Reverse[rI[-π / 2, π / 5.6, 5]]
]
```
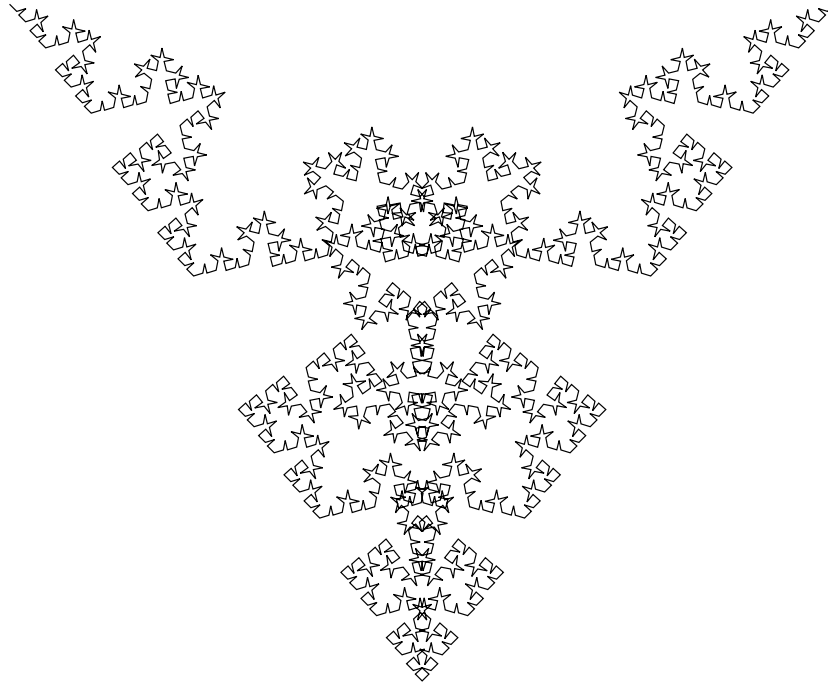
Here is a variation.

```
arcJoin7[θ_, ϕ_] := arcJoin[
  -rI[θ, ϕ, 5],
  Reverse[rI[θ, ϕ, 5]]
 ]
```

```
arcJoin7[π / 4, π / 5.8]
```


Interesting graphics may sometimes be constructed using replacement rules of the following type.

```
{θ, θ + π / 3} /. θ → {θ, θ + ϕ, θ - ϕ, θ};
```

The function `defineRule` only works with lists, so for `rJ` we go back to the style of definition used for `rA`.

```
rJ[θ0_, ϕ0_, n_] := Module[{θ, ϕ},
  Nest[Function[θ, Flatten[
      {θ, θ + π / 3} /. θ → {θ, θ + ϕ, θ - ϕ, θ}
    ]], N[θ0], n] /. ϕ → N[ϕ0]
 ]
```

arcJoin[rJ[π / 2, π / 2, 4]]

rJ[Θ, φ, 1]

$$\left\{\Theta, \ \Theta + \phi, \ \Theta - \phi, \ \Theta, \ \frac{\pi}{3} + \Theta, \ \frac{\pi}{3} + \Theta + \phi, \ \frac{\pi}{3} + \Theta - \phi, \ \frac{\pi}{3} + \Theta\right\}$$
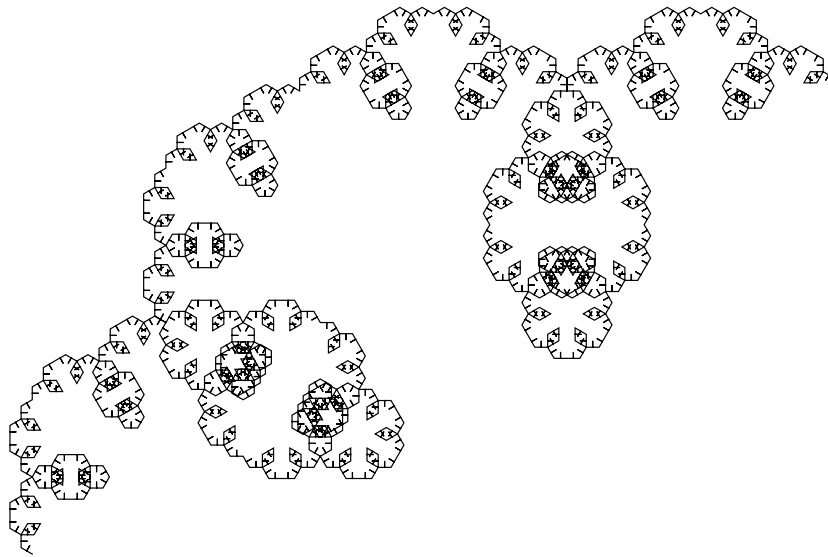
In spite of the result for rJ[Θ, φ, 1], the replacement rule rJ1 does not generate the above graphic.

defineRule[rJ1, {Θ, Θ + π / 3} /. Θ → {Θ, Θ + φ, Θ - φ, Θ}]
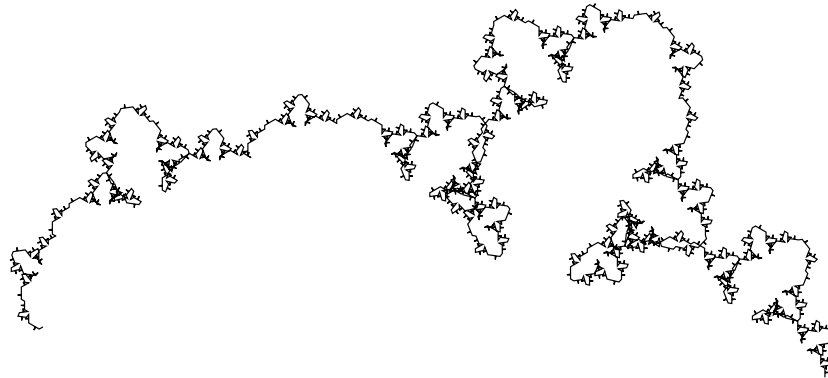
rJ1[Θ, φ, 1]

$$\left\{\Theta, \ \Theta + \phi, \ \Theta - \phi, \ \Theta, \ \frac{\pi}{3} + \Theta, \ \frac{\pi}{3} + \Theta + \phi, \ \frac{\pi}{3} + \Theta - \phi, \ \frac{\pi}{3} + \Theta\right\}$$

```
arcJoin[rJ1[π / 2, π / 2, 4]]
```

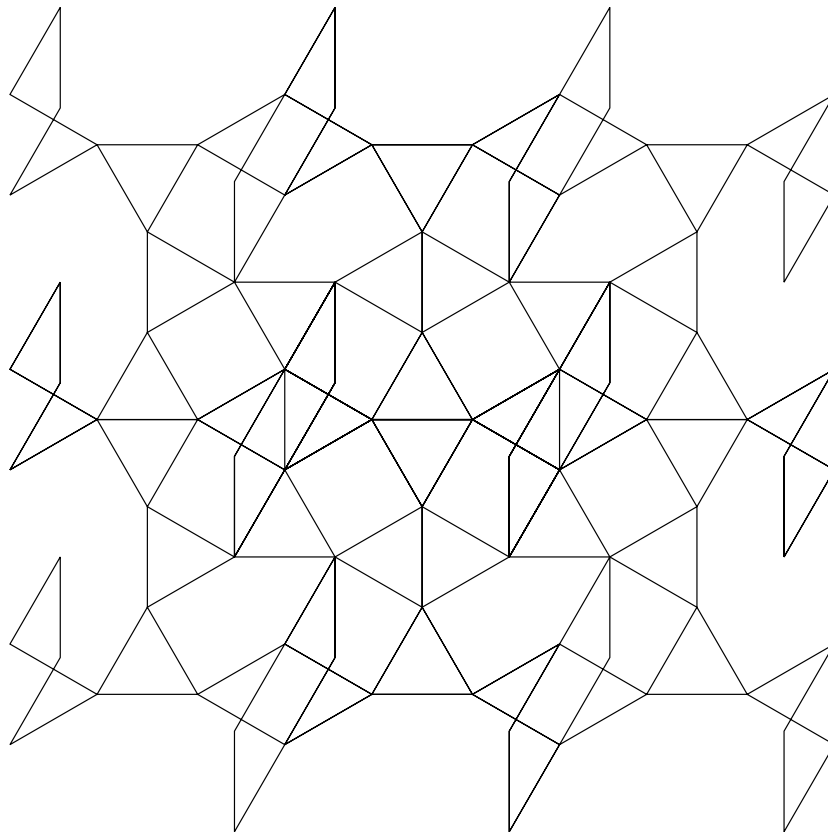A variation.

```
arcJoin[rJ[π / 2, -π / 2.2, 5]]
```

☐ Replacement Rules with Constant Entries

A rewriting system may contain constant entries, which are not affected by iterations. This means that one or more of the line segments of the initial arc remain unchanged after iteration.
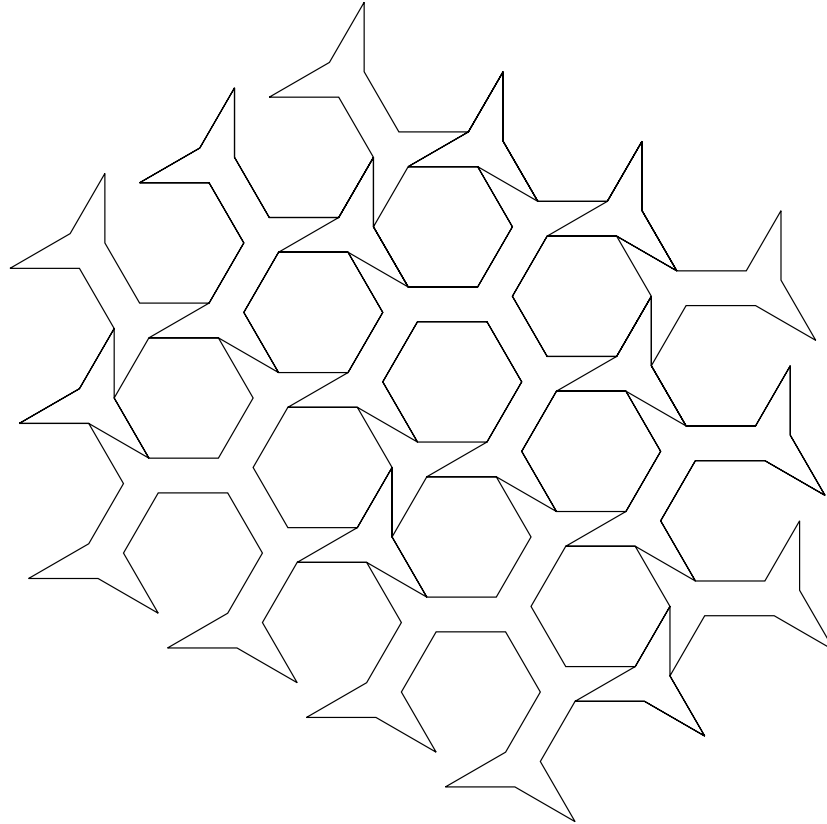
This sometimes gives a tiling of the plane.

```
defineRule[rK, {π / 3, θ - ϕ, π, θ + ϕ, -π / 3}]
```

```
arcJoin[rK[π / 3, π / 2, 6]]
```

Here is a variation.

```
arcJoin[rK[π / 2, π / 3, 6]]
```


■ Plot Variations

◻ Applying Polygon to a List of Vertices

The polygonal arcs constructed by the above method do not always form the boundary of a polygon. However, interesting graphics may sometimes be obtained by applying `Polygon` to the list of vertices. The function `poly` defined below can be used to construct such a graphic.

```
poly[r__] :=
 Graphics[
  Polygon[{Re[#], Im[#]} & /@
    (FoldList[Plus, 0, E^(I (Join[r]))])],
   AspectRatio → Automatic, Axes → False, PlotRange → All]
```

In this example, `Polygon` is applied to the vertices of an arc that is neither closed nor symmetrical.
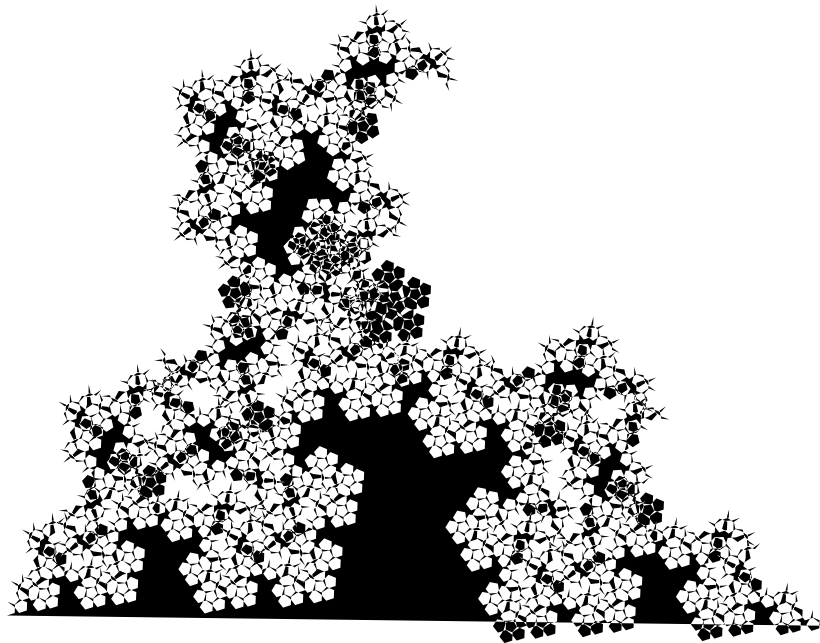
```
defineRule[rL, {θ, θ + ϕ, θ + 2 ϕ, θ - 1.3 ϕ, θ - ϕ, θ}]
```

```
poly[rL[π / 10, 32 π / 5.01, 5]]
```

Here is a variation.

```
poly[rL[-π / 10, -24 π / 5, 4]]
```


The entries in our very first replacement rule `rA` are not all real.

```
rA[θ, ϕ, 1]
```

```
{(0. - 0.405465 i) + θ, θ - 2 ϕ, θ - 3.5 ϕ,
 θ, θ + 3.5 ϕ, θ + 2 ϕ, (0. - 0.405465 i) + θ}
```

Define `rA1` by replacing `I` by `- I`.

```
defineRule[rA1, {θ + I Log[1.5], θ - 2 ϕ, θ - 3.5 ϕ, θ,
 θ + 3.5 ϕ, θ + 2 ϕ, θ + I Log[1.5]}]
```

The replacement rule `- rA1` may be used to generate polygonal arcs that are reflections of those generated by `rA` and similarly for `rA1` and `- rA`.

We illustrate by means of the following example, in which we apply the replacement rule `- rA1` to two sides and `rA` to the third side of an equilateral triangle.

```
poly[
 rA[0, 4 π / 5, 4],
 -rA1[0, 4 π / 5, 4] + 2 π / 3,
 -rA1[0, 4 π / 5, 4] + 4 π / 3
]
```
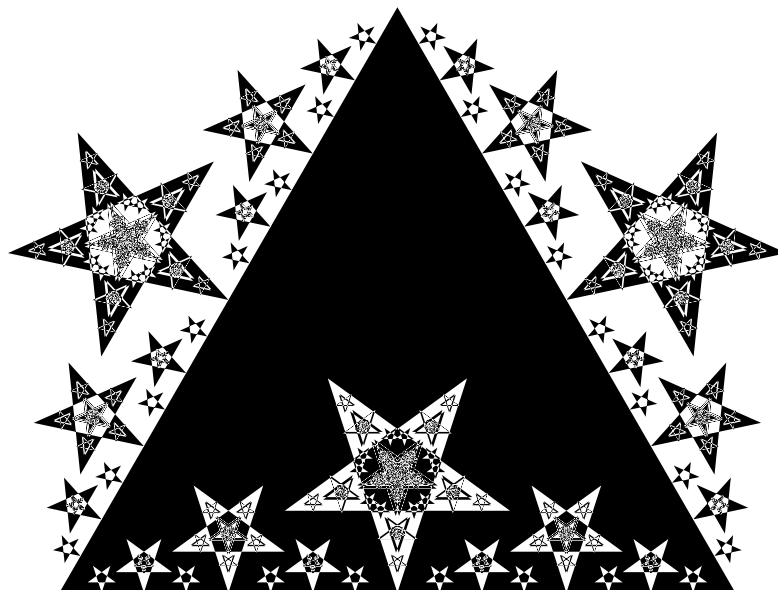
Here is a variation.

```
poly2[θ_, φ_] := poly[rA[θ, φ, 4], -rA1[θ, φ, 4] + 2 π / 3,
  -rA1[θ, φ, 4] + 4 π / 3]
```
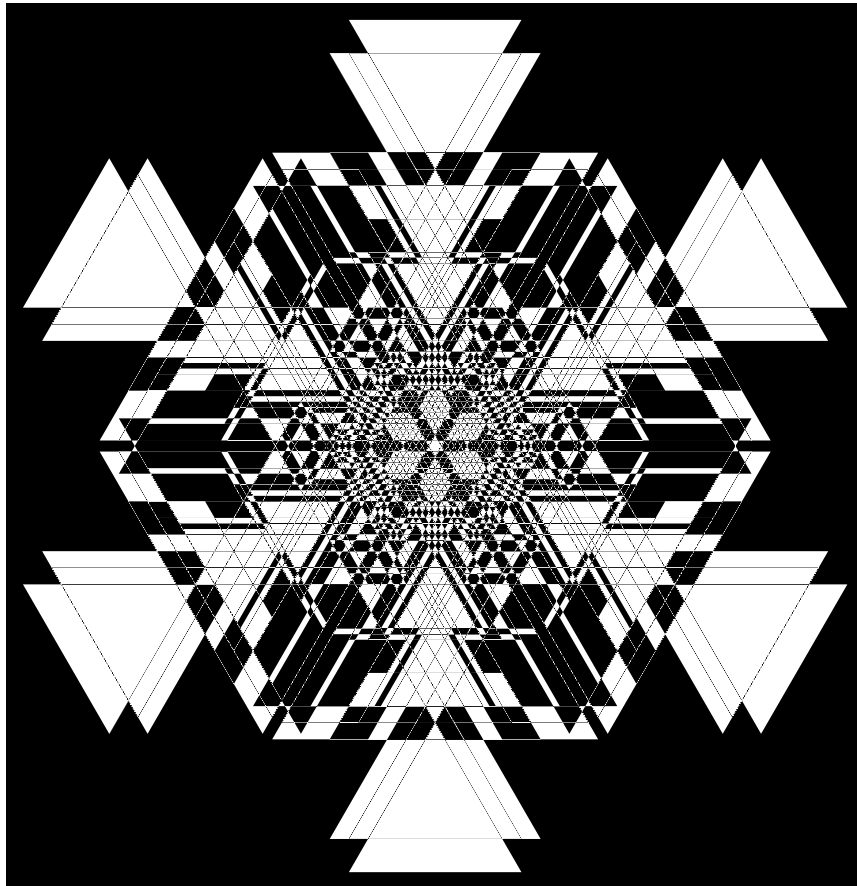
```
poly2[0, -5 π / 3.9]
```

In the following example, we apply the replacement rule `rA1` to the sides of a regular hexagon.

```
hex[r__] :=
 Graphics[
  {White,
   Polygon[{Re[#], Im[#]} & /@
      (FoldList[Plus, 0, E^(I (Join[r]))])]},
   AspectRatio → Automatic, Axes → False, PlotRange → All,
   Background → Black]

hex2[θ_, ϕ_] := hex[rA1[θ, ϕ, 3], rA1[θ, ϕ, 3] + π / 3,
   rA1[θ, ϕ, 3] + 2 π / 3, rA1[θ, ϕ, 3] + π, rA1[θ, ϕ, 3] + 4 π / 3,
   rA1[θ, ϕ, 3] + 5 π / 3]

hex2[0, -3 π / 2.25]
```
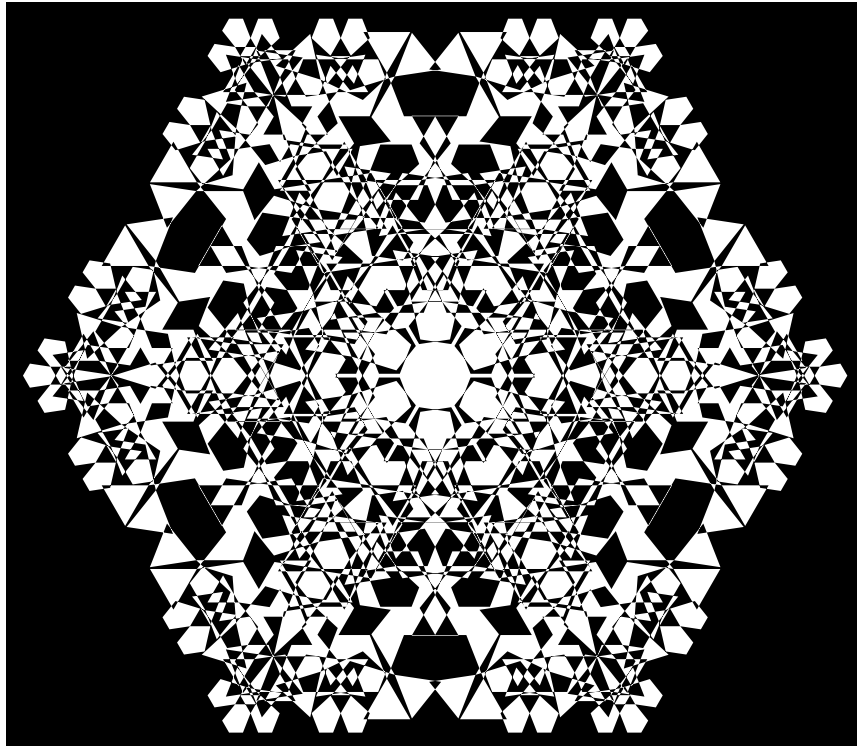
Here is a variation.

```
hex2[0, -π / 5.25]
```


■ References

[1] M. Trott, *The Mathematica GuideBook for Graphics*, New York: Springer-Verlag, 2004.

[2] P. Prusinkiewicz and A. Lindenmayer, *The Algorithmic Beauty of Plants*, New York: Springer-Verlag, 1990.

About the Author

Janet Helmstedt was a lecturer in the Department of Mathematics of the University of the Witwatersrand, Johannesburg, South Africa. She is coauthor with Chonat Getz of the book *Graphics with Mathematica: Fractals, Julia Sets, Patterns and Natural Forms*.

Janet Helmstedt
30 Meyer Street
Oaklands
Johannesburg, 2192, South Africa
janetfractal@hotmail.com