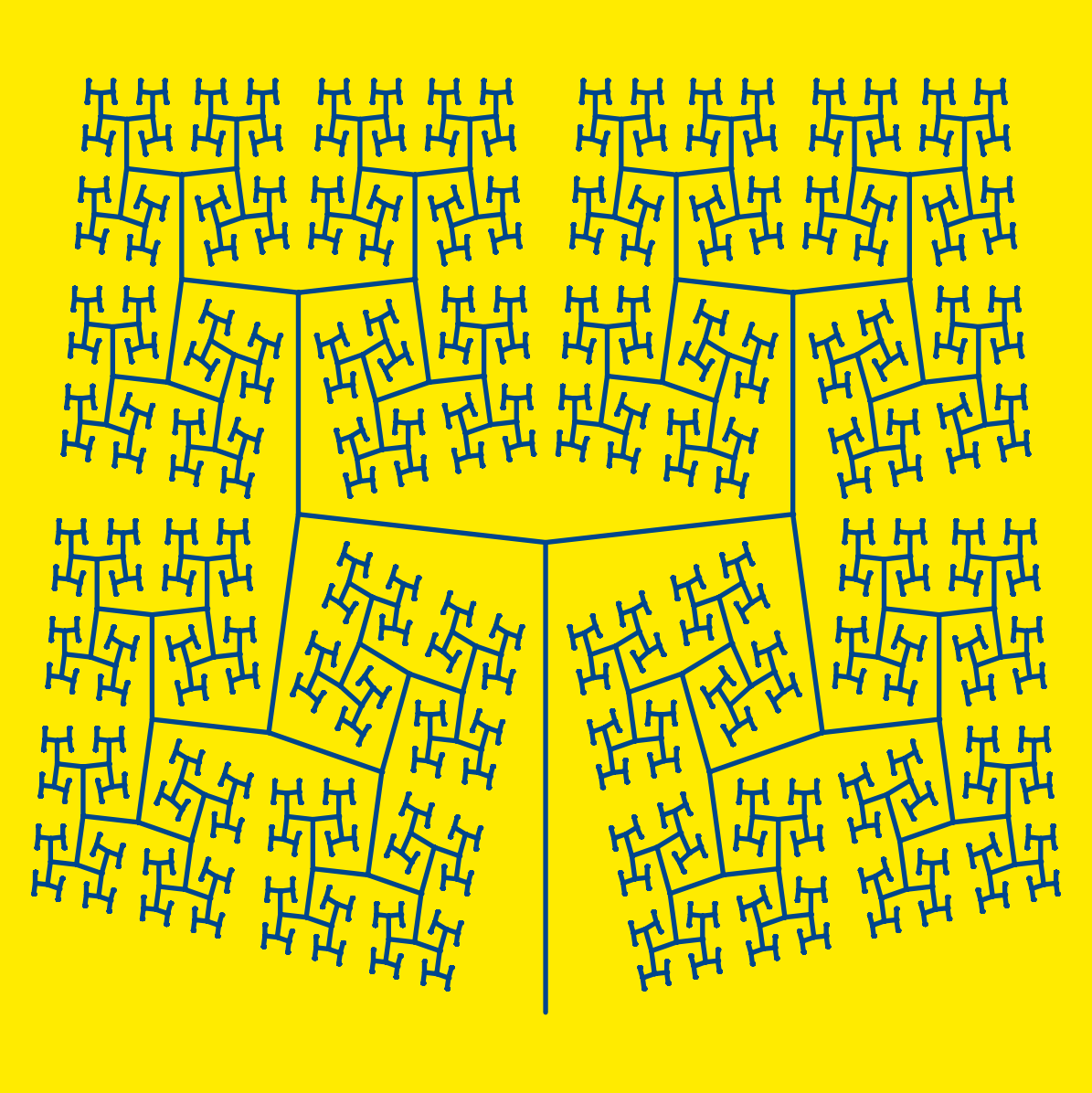
# MetaPost NEWS

**Hans Hagen**

**Mikael Sundqvist**

BachoTEX 2026



## METAPOST NEWS

Playing with MetaPost/MetaFun is ... fun.

We will here present some new additions (and some ongoing work).

You can read more in the articles or in the documentation

```
./tex/texmf-context/doc/context/documents/
 /general/manuals/luametafun.pdf
```

that is updated as work goes on.

## RELATIVE POINTS

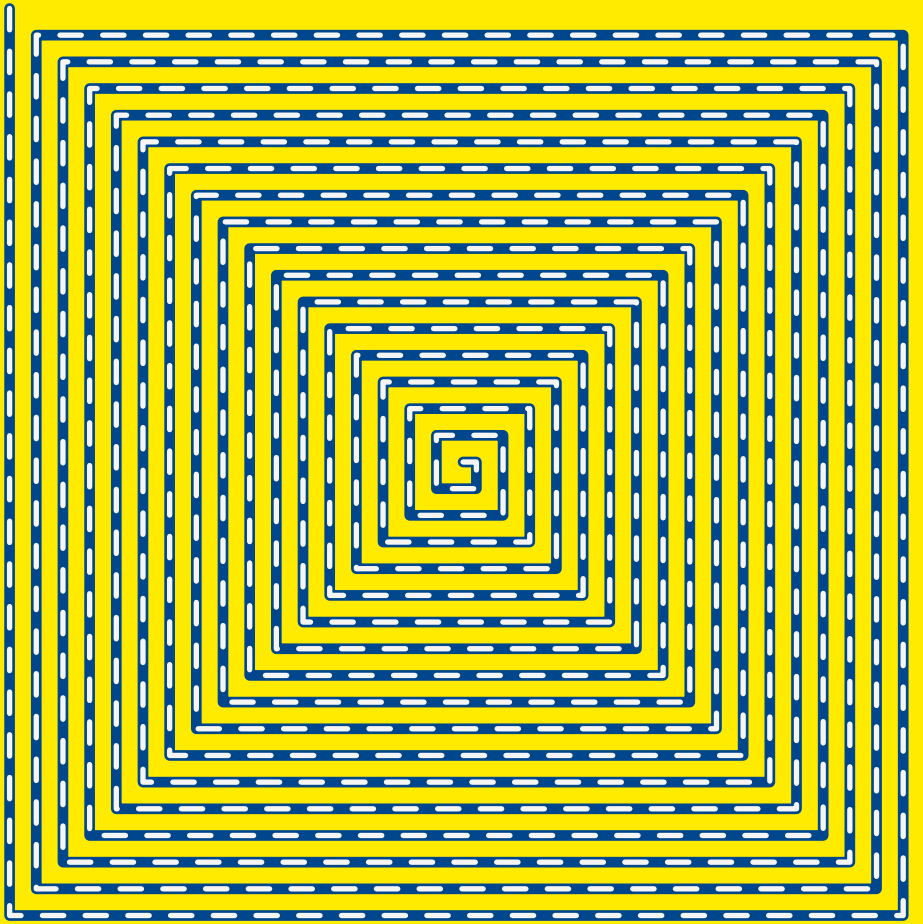
```
fill unitsquare
     scaled PaperHeight
     withcolor "C:2" ;

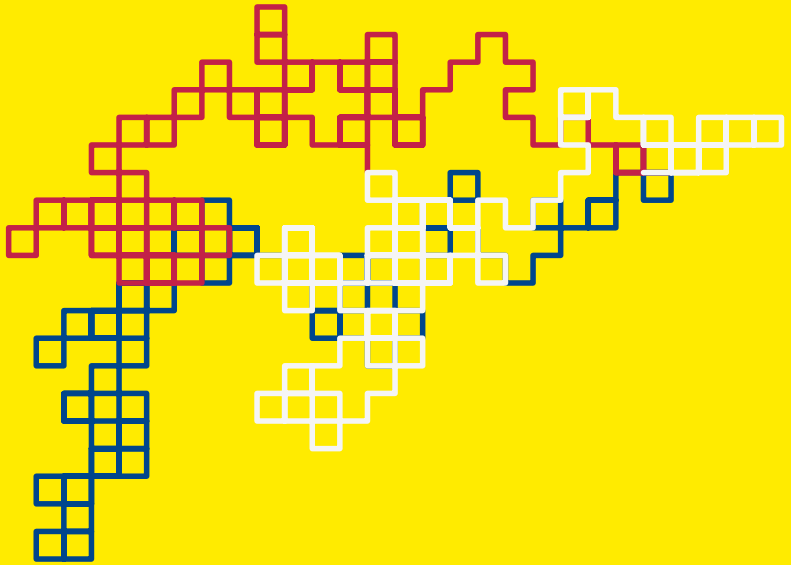
path p ;

p := origin
     for i = 0 upto 16 :
       -- xrelative   20i +  5
       -- yrelative -(20i + 10)
       -- xrelative -(20i + 15)
       -- yrelative   20i + 20
     endfor ;

draw p withpen pencircle scaled 4
       withcolor "C:1" ;

draw p withdashes (8,8)
       withpen pencircle scaled 2
       withcolor "C:4" ;
```





## RELATIVE AND ABSOLUTE MOVEMENTS

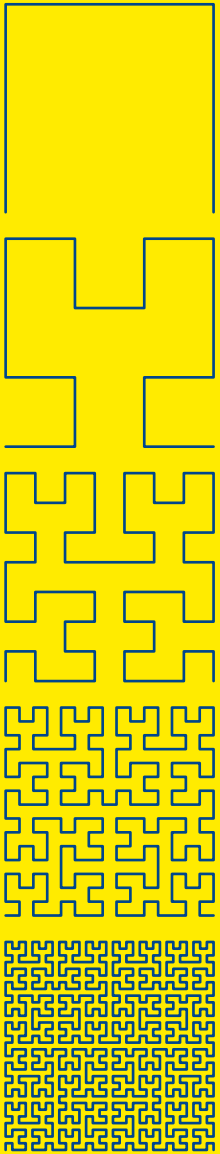
`xrelative dx` Create a point that is shifted `dx` in horizontal direction relative to the previous point.

`yrelative dy` Create a point that is shifted `dy` in vertical direction relative to the previous point.

`xyrelative (dx,dy)` Create a point that is shifted `(dx,dy)` relative to the previous point. Shift `(dx,dx)` if the numerical value `dx` is given.

`xabsolute x` Create a point, shifted horizontally compared to the previous point, so that the first coordinate becomes `x`.

`yabsolute y` Create a point, shifted vertically compared to the previous point, so that the second coordinate becomes `y`.



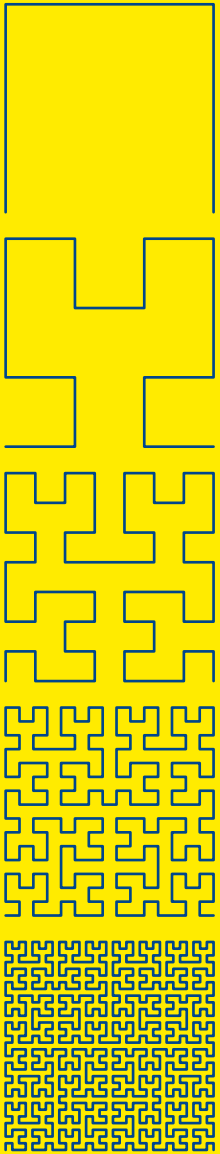
## A HILBERT CURVE

```
numeric cang ;

def polarrelative(expr r,phi) =
  xyrelative (r*cosd(phi),-r*sind(phi))
enddef ;

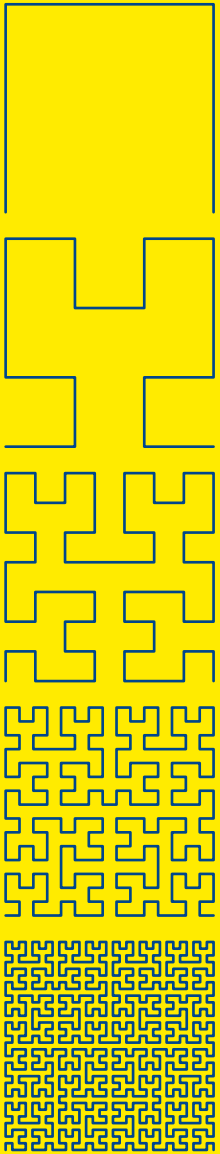
def turn(expr ang) =
  hide(cang := cang - ang)
enddef ;

def walk(expr dis) =
  -- polarrelative(dis,cang)
enddef ;
```



## A HILBERT CURVE

```
def hilbert(expr iteration, size, ang) =
  if iteration == 0 :
  else :
    turn(ang)
    hilbert(iteration - 1, size, -ang)
    walk(size)
    turn(-ang)
    hilbert(iteration - 1, size,  ang)
    walk(size)
    hilbert(iteration - 1, size,  ang)
    turn(-ang)
    walk(size)
    hilbert(iteration - 1, size, -ang)
    turn(ang)
  fi
enddef ;
```



## A HILBERT CURVE

```
path p ;

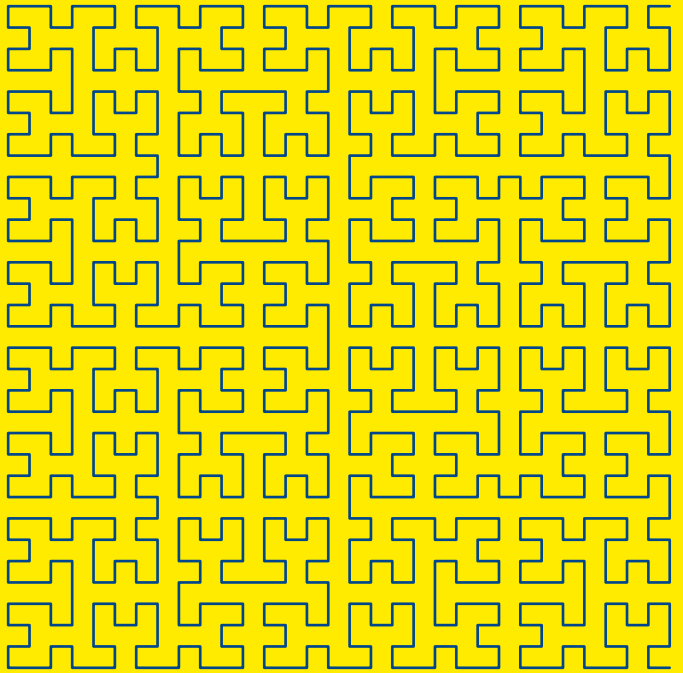
for i = 1 upto 5 :
  cang := 0 ;
  p := (origin hilbert(i, 1, 90))
    xsized 2.75cm ;
  draw p shifted (-1.5cm,4.85cm-(i - 1)*3.1cm)
    withpen pencircle scaled 1
    withcolor "C:1" ;
endfor ;
```

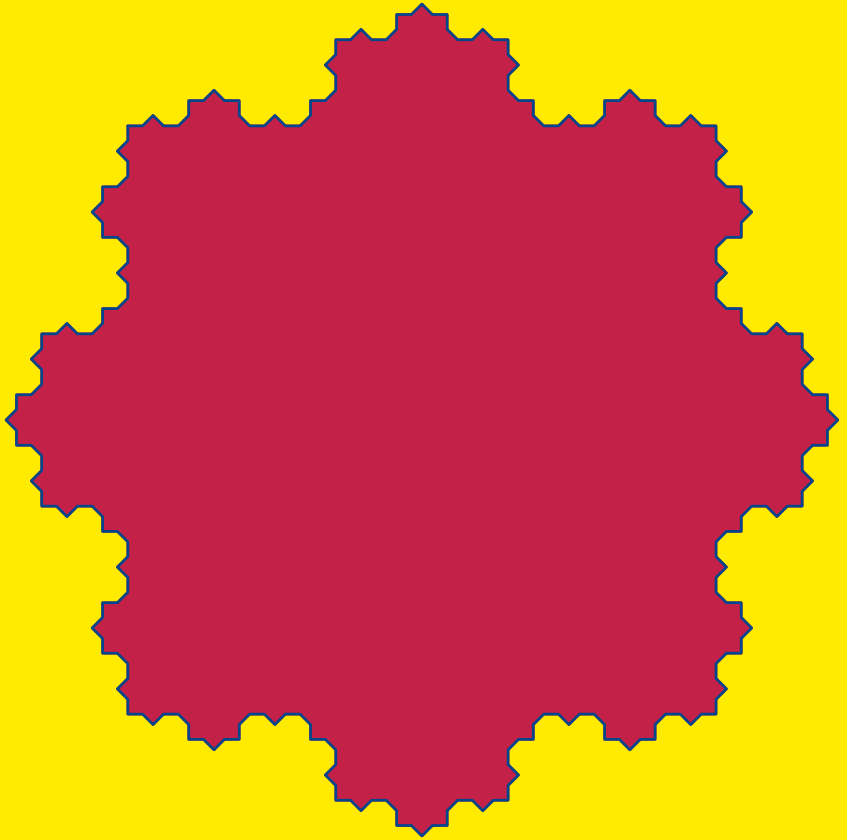
## HILBERT CURVE AS A LINDENMAYER SYSTEM

```
path p ; p := lmt_lsystem [
  name      = "hilbert",
  angle     = 90,
  scale     = 8,
  iteration = 6,
  syntax    = [
    start   = "A",
      X     = "",
      A     = "+BX-AXA-XB+",
      B     = "-AX+BXB+XA-",
  ]
] ;

p := p shifted - center p ;

draw p withcolor "C:1"
       withpen pencircle scaled 1 ;
```





## KOCH SNOWFLAKE

```
vardef Koch(expr n) =
  lmt_lsystem [
    angle     = 45,
    rotation  = 0,
    iteration = n,
    scale     = 0.5cm,
    syntax    = [
      start = "X",
        X = "X+X--X+X",
    ]
  ]
enddef ;
```

## KOCH SNOWFLAKE

```
drawoptions(withpen pencircle scaled 1
            withcolor "C:1") ;

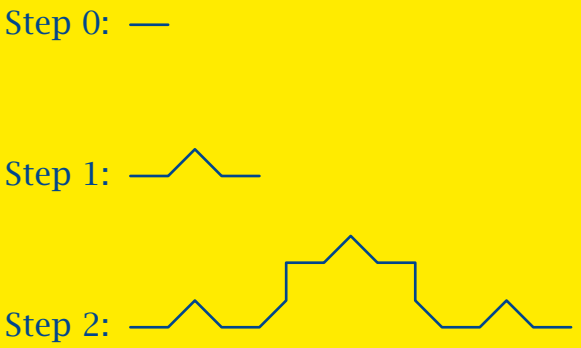
picture p ;

p := image(
  draw (0,0) -- (0.5cm,0) ;
  draw Koch(1) shifted (0,-2cm) ;
  draw Koch(2) shifted (0,-4cm) ;
  for i = 0 upto 2 :
    label.lft("Step " & decimal i & ": ", (0,-i*2cm)) ;
  endfor );

p := p shifted - center p ;

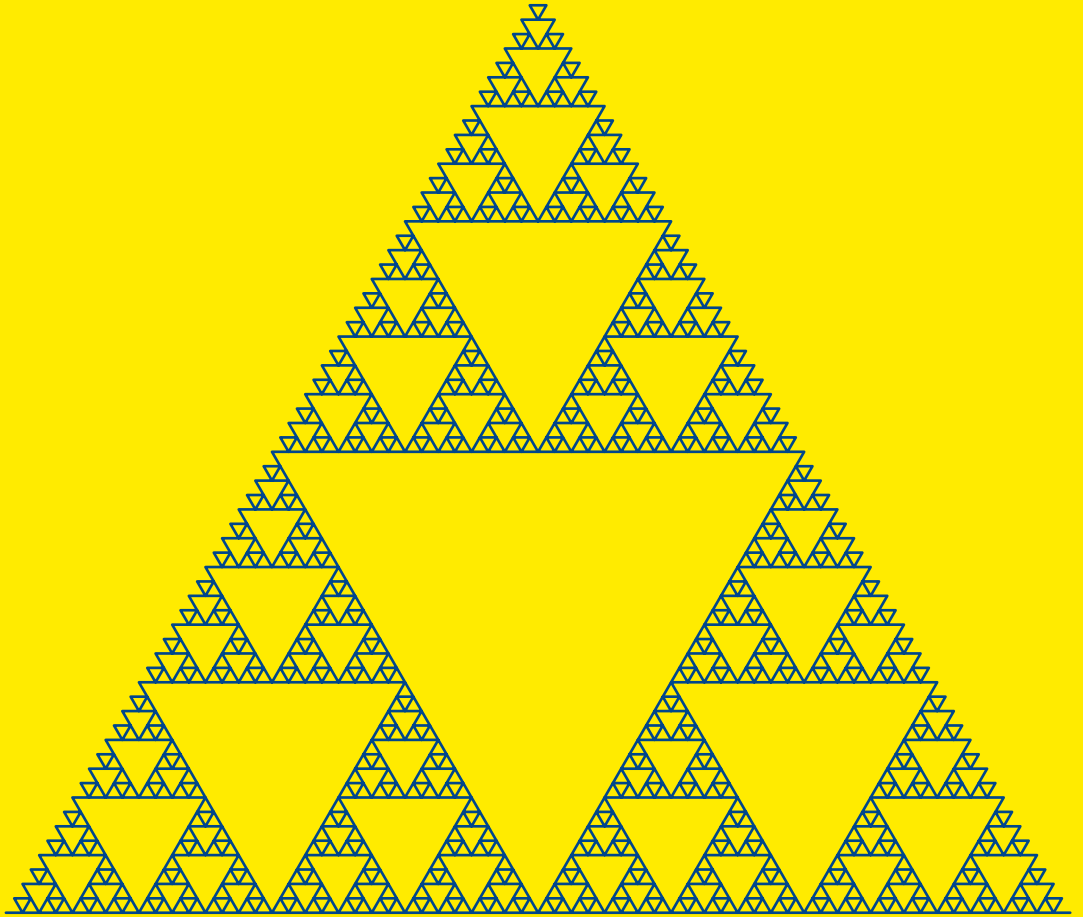
draw p ;

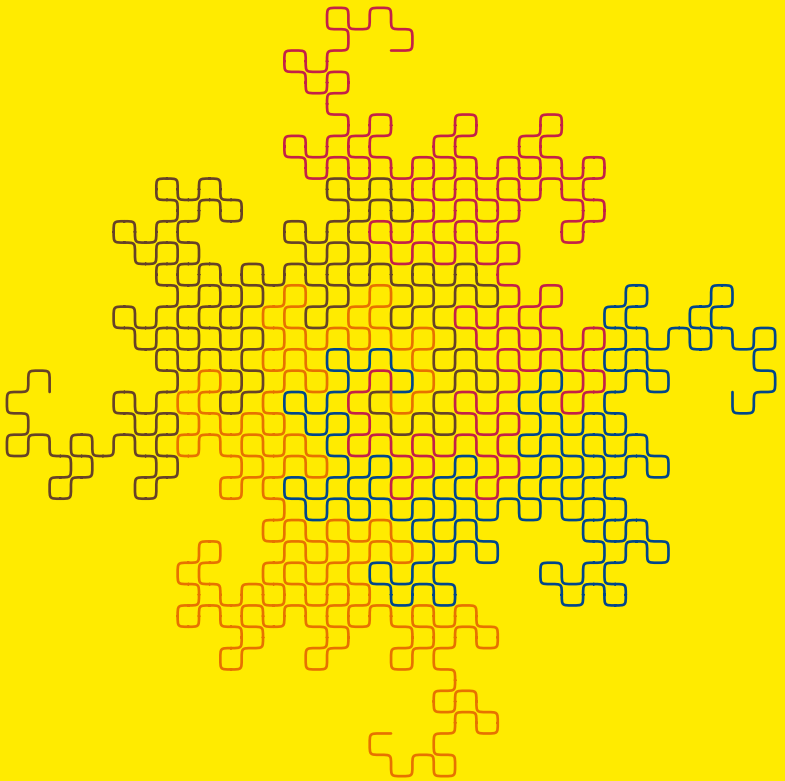
% Remember the rule: X = "X+X--X+X"
```



## A SIERPIŃSKI TRIANGLE

```
path p ; p := lmt_lsystem [
    name      = "sierpinski",
    angle     = 120,
    scale     = 1,
    iteration = 6,
    syntax    = [
        start = "L",
          L   = "L-R+L+R-L",
          R   = "RR",
    ]
] xsized (12cm)
  rotated 90 ;
p := p shifted - center p ;
draw p withcolor "C:1"
       withpen pencircle scaled 1 ;
```





## DRAGON CURVES

```
path p ; p := lmt_lsystem [
  name      = "four dragon",
  angle     = 90,
  rotation  = 0,
  scale     = 8,
  iteration = 8,
  syntax    = [
    start   = "L",
      L     = "L+R",
      R     = "L-R",
  ]
] ;

p := p smoothcornered 7 ;
pickup pencircle scaled 1 ;

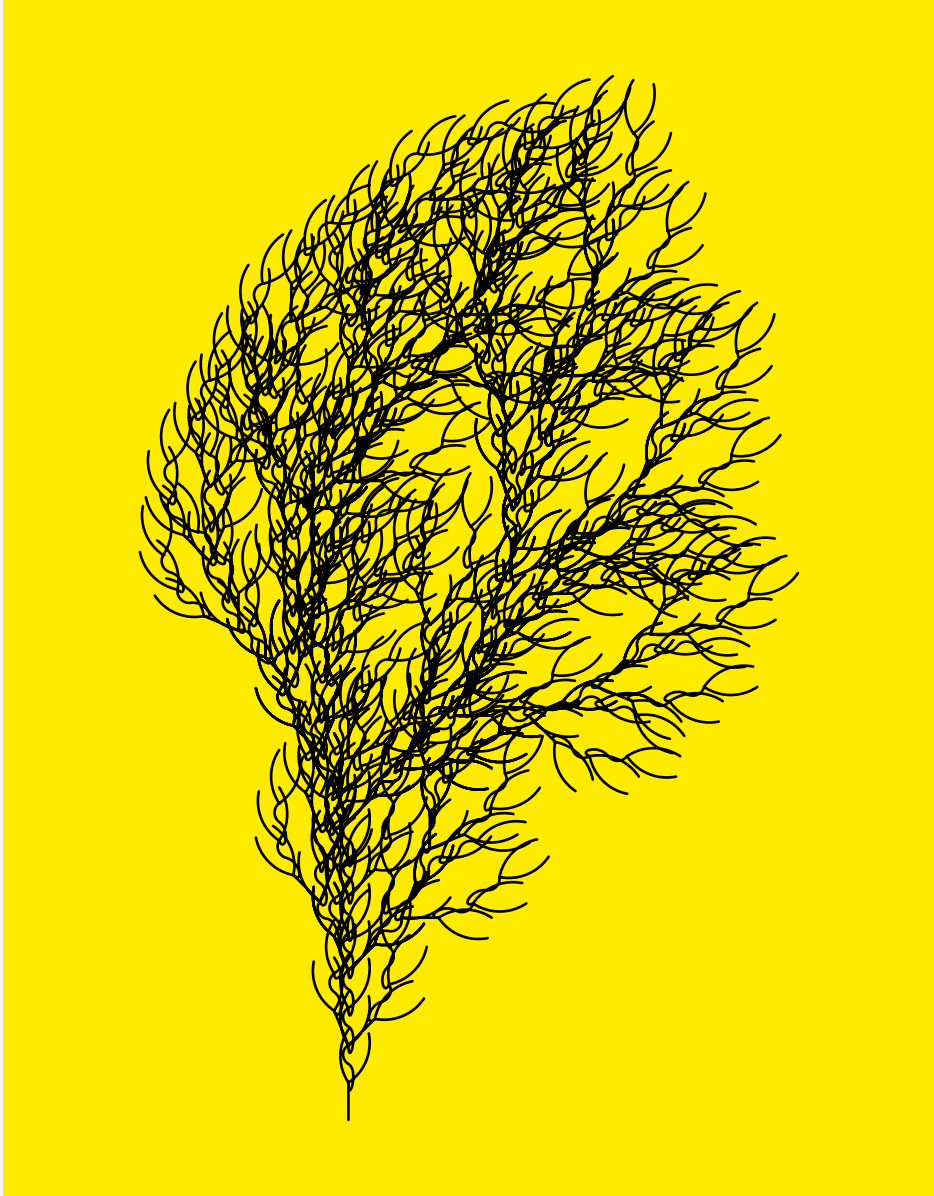
for i = "C:1", "C:3", "C:5", "C:6" :
  draw p withcolor i ;
  p := p rotated 90 ;
endfor
```

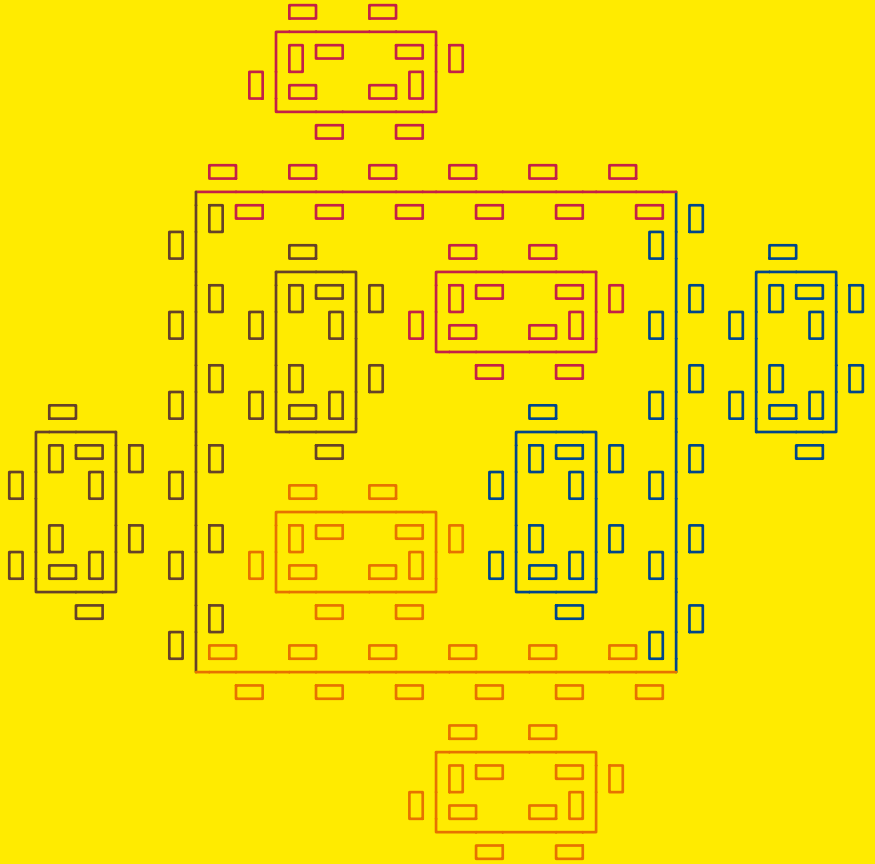
## A TREE?

```
path p ;
p := lmt_lsystem [
  angle     = 22.5,
  iteration = 4,
  scale     = 7,
  curved    = true,
  syntax    = [
    start   = "F",
    F       = "FF-[-F+F+F]+[+F-F-F]",
  ]
] ;

p := p shifted - center p ;

draw p withpen pencircle scaled 1 ;
```





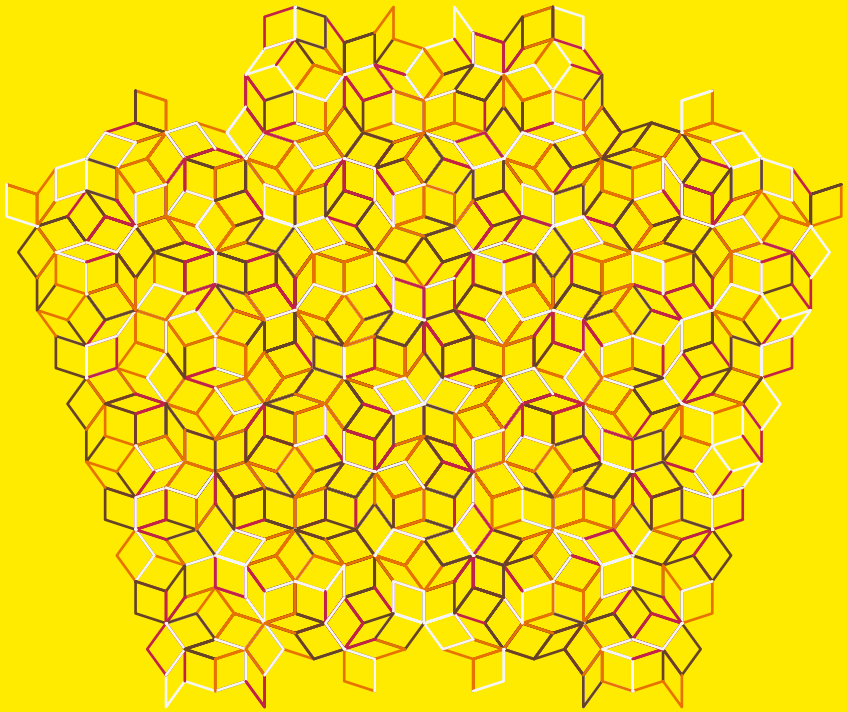
## USING COLORS

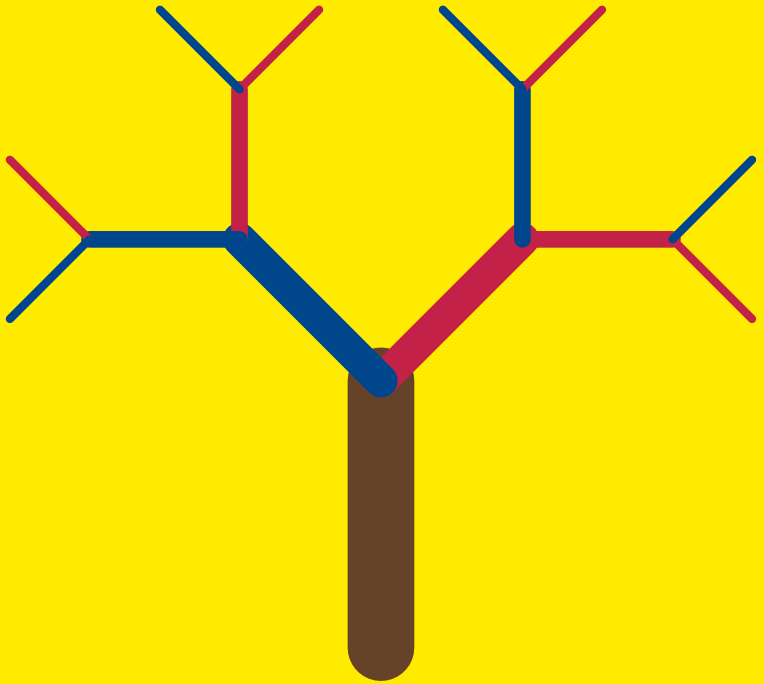
```
picture p ;
p := lmt_lsystem [
  angle     = 90,
  iteration = 2,
  scale     = 5,
  syntax    = [
    start = "{withcolor 'C:1' } F+
             {withcolor 'C:3' } F+
             {withcolor 'C:5' } F+
             {withcolor 'C:6' } F",
        F = "F+f-FF+F+FF+Ff+FF-f+FF-F-FF-Ff-FFF",
        f = "ffffff",
  ]
] ;

p := p shifted - center p ;
draw p withpen pencircle scaled 1 ;
```

## A PENROSE TILING

```
picture p ;
p := lmt_lsystem [
  angle     = 36,
  rotation  = 90,
  iteration = 5,
  scale     = 12,
  syntax=[
    start = "{withcolor 'C:1'}+W--X---Y--Z",
        W = "Y++Z----X{withcolor 'C:3'}[-Y----W]++",
        X = "+Y--Z{withcolor 'C:4'}[---W--X]+",
        Y = "-W++X{withcolor 'C:5'}[+++Y++Z]-",
        Z = "--Y++++W{withcolor 'C:6'}[+Z++++X]--X",
  ],
] ;
p := p shifted - center p ;
draw p withpen pencircle scaled 1 ;
```





## A SCALED TREE

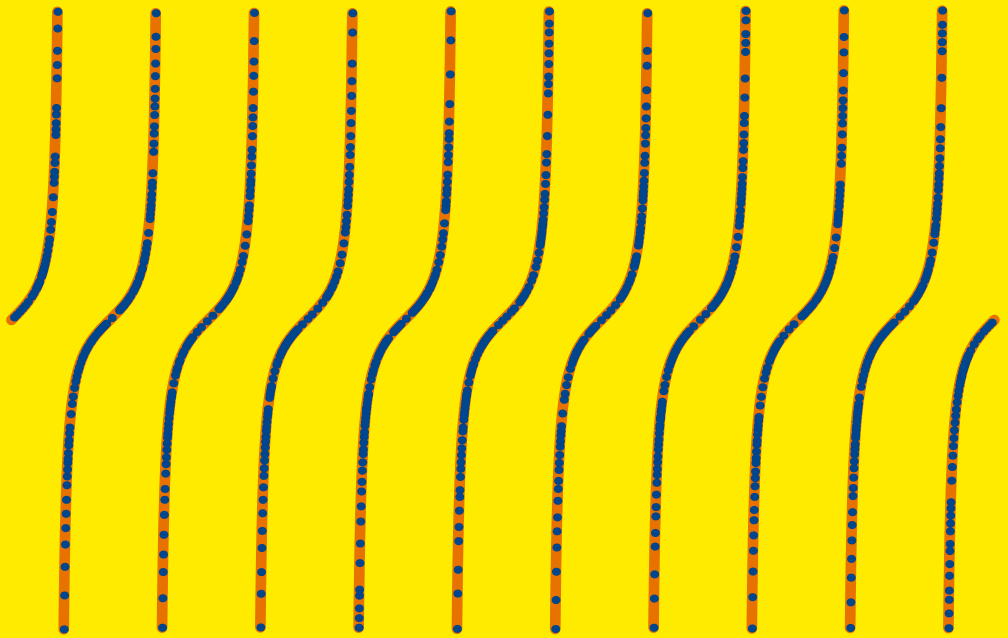
```
picture p ;
p := lmt_lsystem [
  angle     = 45, scale     = 100,
  iteration = 4, linewidth = .5,
  syntax    = [
    start = " { withpen pencircle scaled lw
                withcolor 'C:5' } FX ",
    X = "( sx := sx * 0.75 ; sy := sy * 0.75 ;
           lw := lw / 2 ; )
         [ { withpen pencircle scaled lw
             withcolor 'C:3' } -FX ] +FX
         { withpen pencircle scaled lw
           withcolor 'C:1' }
         ( lw := lw * 2 ; ) ",
    F = "",
  ],
] ;
p := p shifted - center p ;
draw p ;
```

## OSCILLATED?

```
draw lmt_followtext [
    text = "\TitleFont\WORD Ulam",
    path = reverse halfcircle,
    autoscaleup = "max",
] xsized 14cm
  withcolor "C:1" ;

fill lmt_oscillated [
    text = "\TitleFont\WORD Ulam",
    path = reverse halfcircle,
] xsized 14cm
  yshifted -6cm
  withcolor "C:6" ;
```

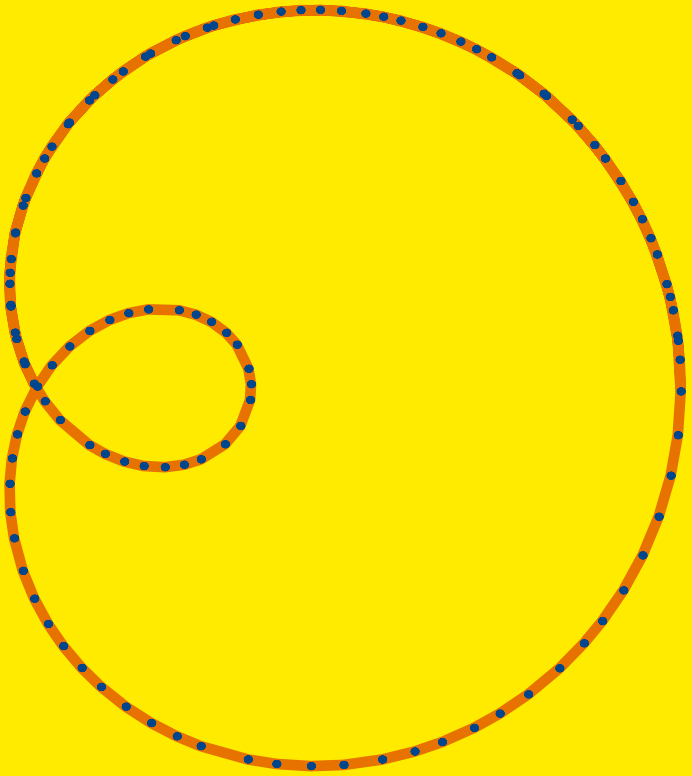




## ADAPTIVE PLOTS

```
path p ; p := lmt_samplefunction [
    preamble  = "local tan = math.tan",
    code      = "return tan(x)",
    xmin      = -5*pi,
    xmax      =  5*pi,
    ymin      =  -10,
    ymax      =   10,
  % shape     = "curve",
  % tight     = true,
  % tolerance = 0.001,
] xsized 13cm ;

draw    p withpen pencircle scaled 4
          withcolor "C:6" ;
drawdot p withpen pencircle scaled 3
          withcolor "C:1" ;
```

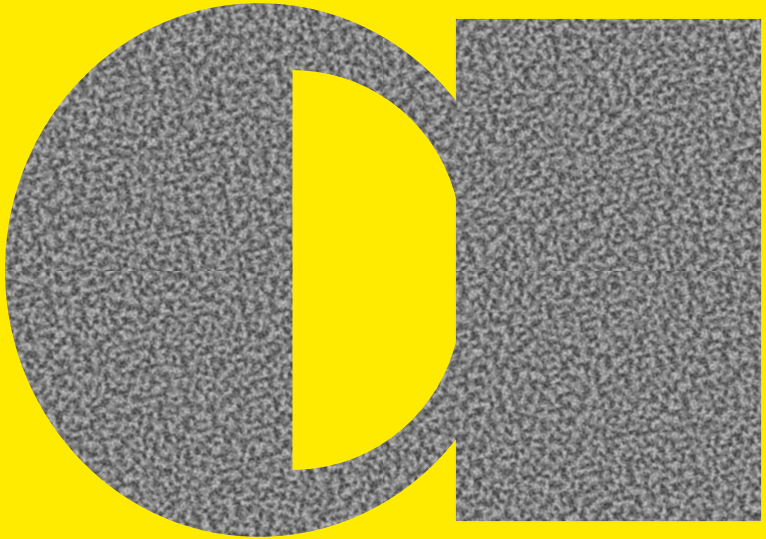


## PARAMETRIC PLOTS

```
path p ; p := lmt_parametricplot [
    preamble  = "local sin, cos = math.sin, math.cos",
    xcode     = "2*cos(t/3)*cos(t)",
    ycode     = "2*cos(t/3)*sin(t)",
    tmin      =   0,
    tmax      =   4*pi,
    tolerance = 0.001,
] ysized 10cm ;

p := p shifted - center p ;

draw    p withpen pencircle scaled 4
          withcolor "C:6" ;
drawdot p withpen pencircle scaled 3
          withcolor "C:1" ;
```



## A LITTLE BIT OF EVERYTHING

```
picture p ;
p:= (
      lmt_outline [
          kind = "outline",
          text = "\TitleFont a",
      ]
    ) xsized 10cm
  withpattern image (
      draw lmt_noise [
          bytemap      = 4,
          nx           = 1500,    ny          = 1500,
          nz           = 3,       iterations  = 2,
          frequency    = 0.05,    amplitude   = 2,
          persistence  = 0.5,     lacunarity  = 2,
          minimum      = 0,       maximum     = 255,
          method       = "angle", angle       = 45,
      ] xsized 10cm ;
  ) ;

p := p shifted - center p ; draw p ;
```

## THE WORK PROCESS

- we see a challenge or need a distraction
- we explore and prototype in regular MetaPost code
- if we need something clean or performing we add to MetaPost
- we make plenty of examples
- and of course chat and joke and ponder
- we prefer to provide a `lmt_whatever [ key = value]` interface
- we also wrap up in an article and add to the manual
- when doing that code might improve

It happens in `mp-lmtx.mpxl` and `mlib-*.lmt` files:

| | |
|---|---|
| `lmt_lsystem` | Lua |
| `lmt_parametricplot` | Lua |
| `lmt_sampleplot` | Lua |
| `lmt_oscillated` | MetaPost |
| `lmt_noise` | Lua and MetaPost |

