

On the “progression transformed into art” project

Krzysztof Pszczoła

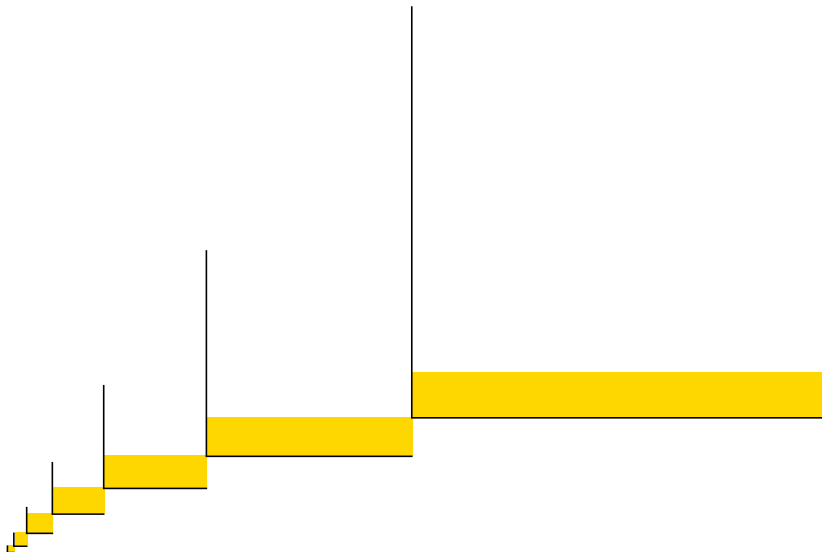
Instytut Matematyki UJK, Poland, pszczola@fr.pl

BachTeX 2015

In the project “[progression transformed into art](#)” we have [graphics](#)’ posinging all of the following three properties:

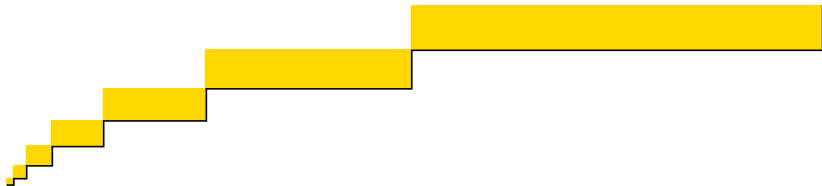
- they [visualize](#) certain [progressions](#) (arithmetic, geometric. . .),
- they are [generated](#) with a short and elegant [METAPOST](#) code (that’s a working assumption, but as it happens with assumptions, the realizations might differ),
- they are [visually attractive](#) (that’s again an assumption; see above).

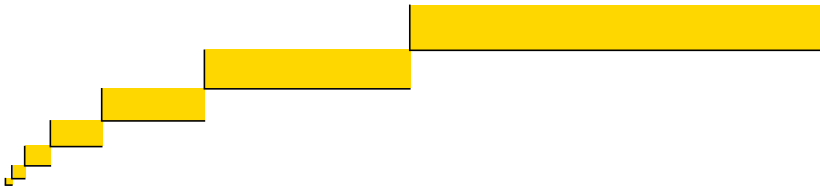
The web-page of the project is <http://pro-trans.tumblr.com/>.

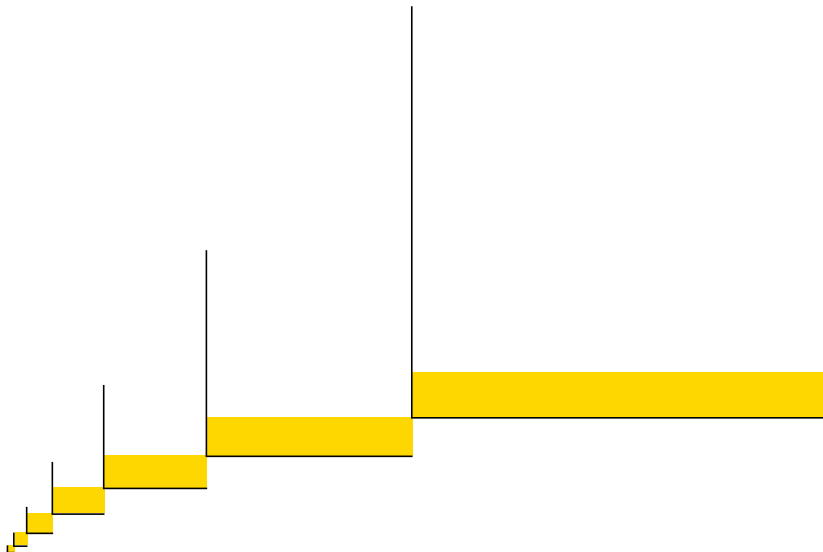


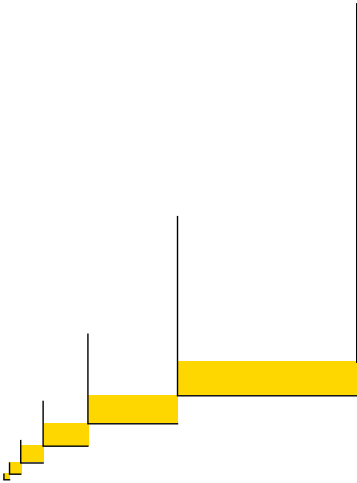












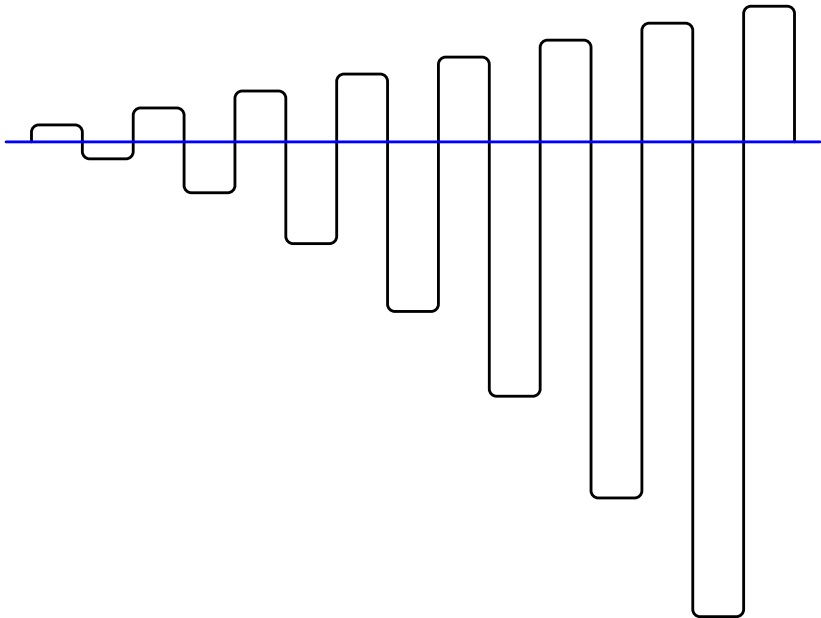
```

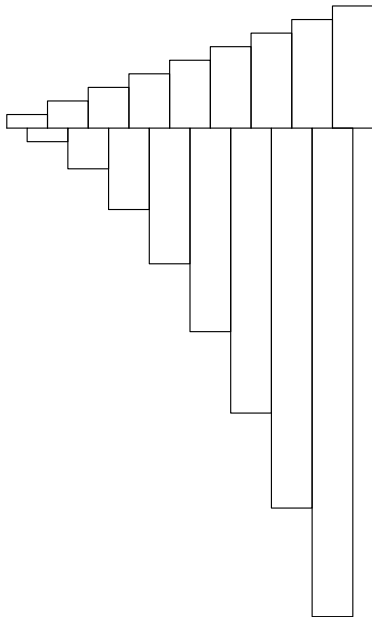
color gold;
gold := (255/255,215/255,0/255);

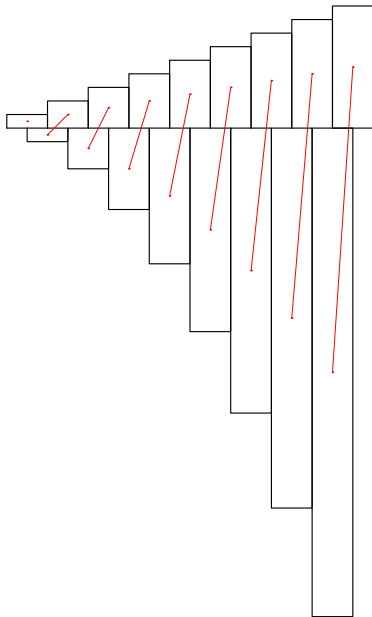
beginfig(1);
  m:=7;
  w:=100p;
  r:=w;
  q:=2;
  width:=w;
  height:=w;
  sx:=0;
  sy:=0;
  h1h:=h1p;
  pickup penquare scaled 2;
  for i+0 upto m-1
    fill (0,0)--(width,0)--(width,height)--(0,height)--cycle
    shifted (sx,sy) withcolor gold; % bars
    draw (sx+width-h1h,sy-h1h)--(sx-h1h,sy-h1h)--(sx-h1h,sy+width-h1h); % lines
    sx:=sx+widthh;
    sy:=sy+height;
    width:=width*q; % geometric progression
    height:=height*r; % arithmetic progression
  endfor;
endfig;

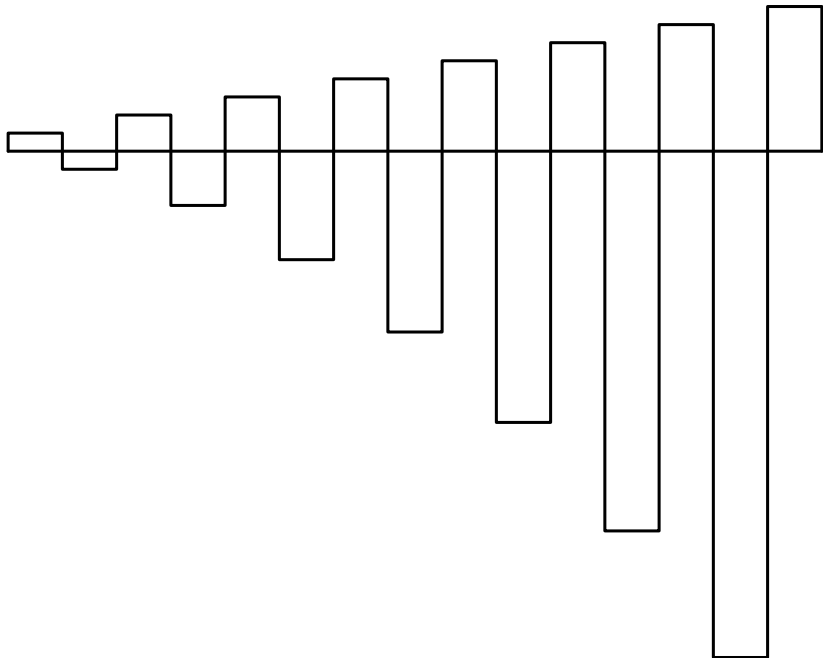
```

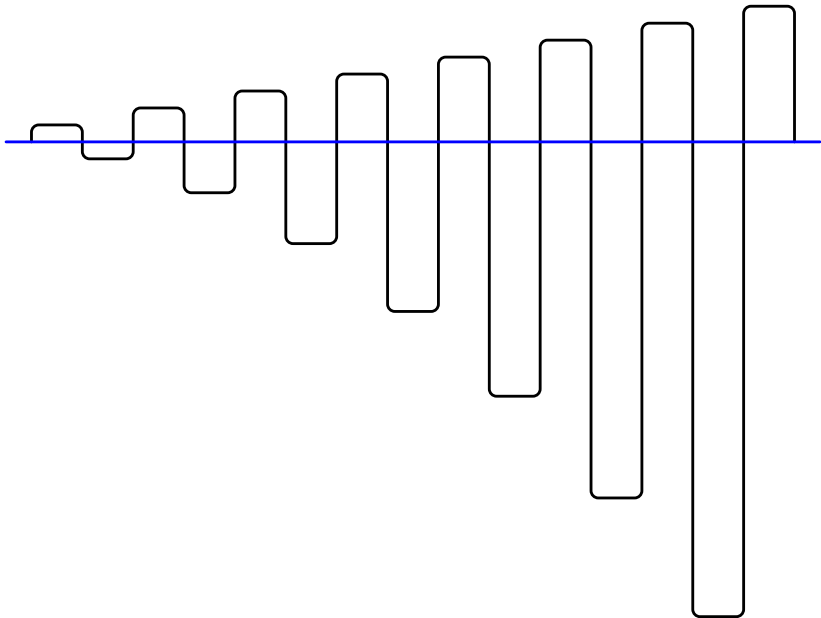
progression transformed into art

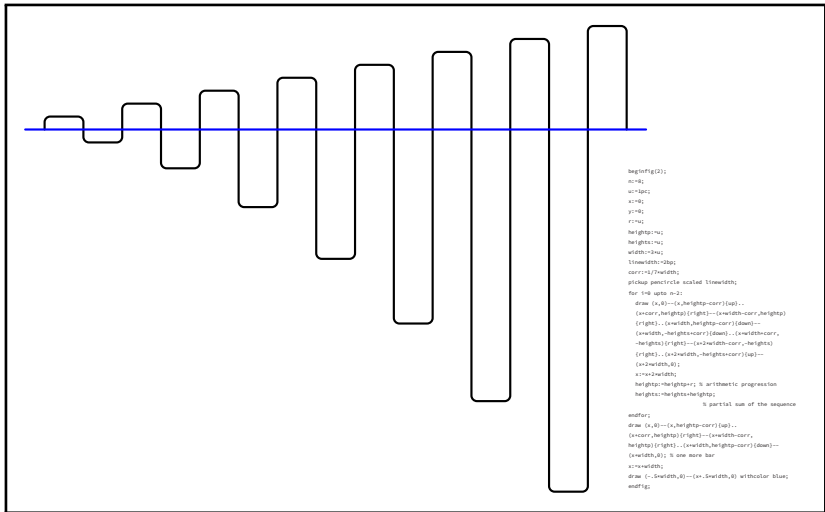








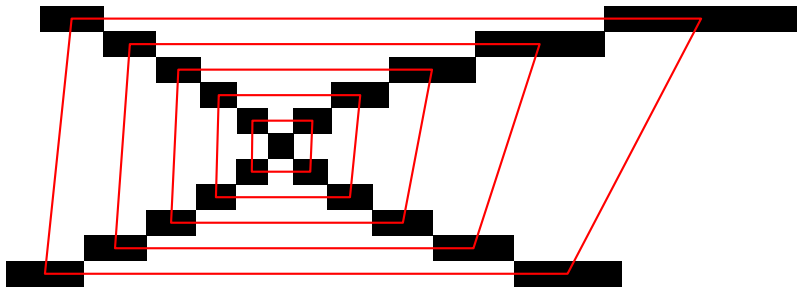


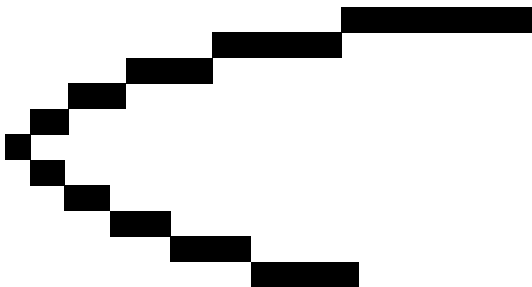


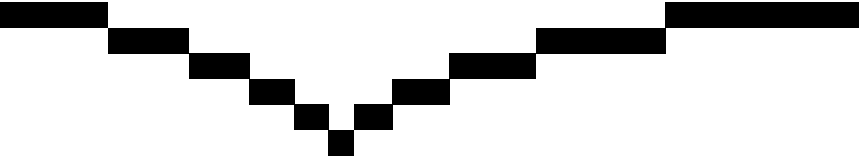
```

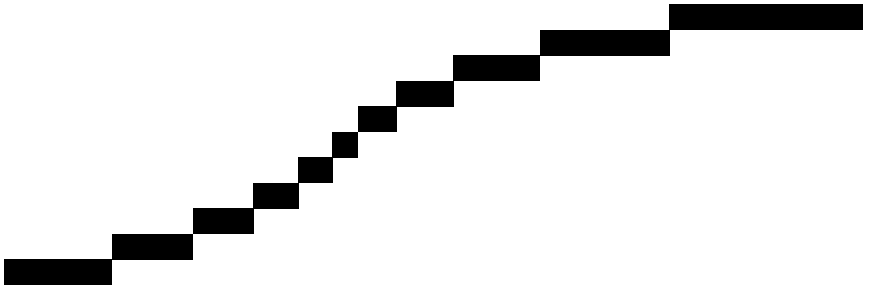
beginfig(2);
nc:=8;
w:=1pc;
x:=0;
y:=0;
r:=w;
height:=w;
width:=3w;
linewidth:=2bp;
corr:=1/7*width;
pickup pencircle scaled linewidth;
for i=0 upto n-1;
  draw (x,0)--(x,height-corr){up}..
    (x+corr,height){r|gt}--(x*width-corr,height)
    {r|gt}..(x*width,height-corr){down}--
    (x*width,-height+corr)--(x*width-corr,
    -height){r|gt}--(x+2*width-corr,-height)
    {r|gt}..(x+2*width,-height+corr){up}--
    (x+2*width,0);
  x:=x+2*width;
  height:=height+r; % arithmetic progression
  height:=height+height;
                    % partial sum of the sequence
endfor;
draw (x,0)--(x,height-corr){up}..
(x+corr,height){r|gt}--(x*width-corr,
height){r|gt}..(x*width,height-corr){down}--
(x*width,0); % one more bar
x:=x+width;
draw (x,-.5*width,0)--(x+.5*width,0) withcolor blue;
endfig;

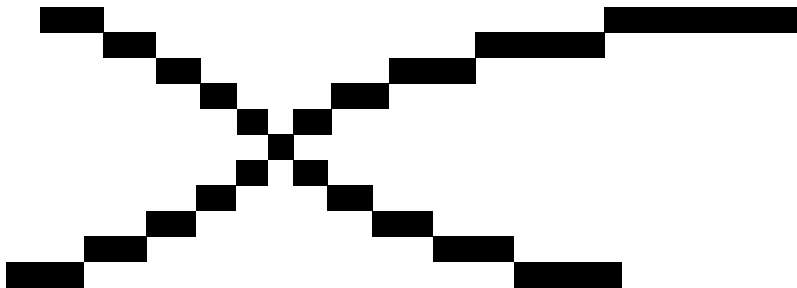
```

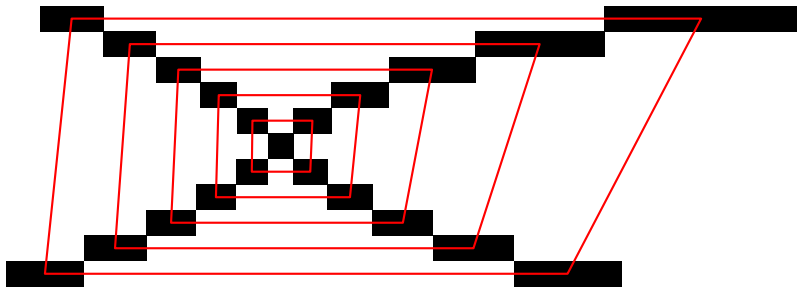


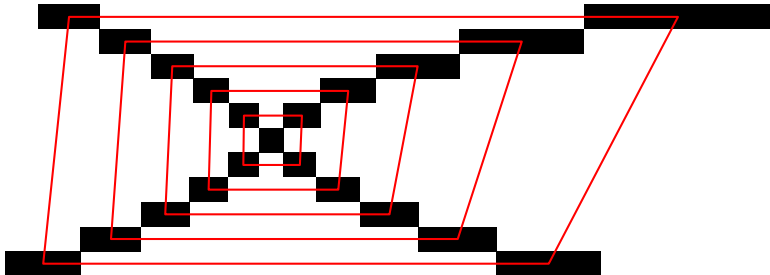












```

beginfig(33);
pair A[]; B[]; C[]; D[];
m:=0;
w:=w0;
% center
fill (0,0)--(w,0)--(w,w)--(0,w)--cycle;
% SE branch
fill (0,0)--(w,0)--(w,0)--(w,w)--(0,w)--cycle;
width:=w/2;
height:=w;
x:=w;
y:=0;
for 1+0 upto n-1
  fill (0,0)--(width,0)--(width,height)--(0,height)
  --cycle shifted (x,y);
  A[]:=1/2[(x,y),(x+width,y+height)];
  x:=width;
  y:=y+height;
  width:=width/2;
endfor;

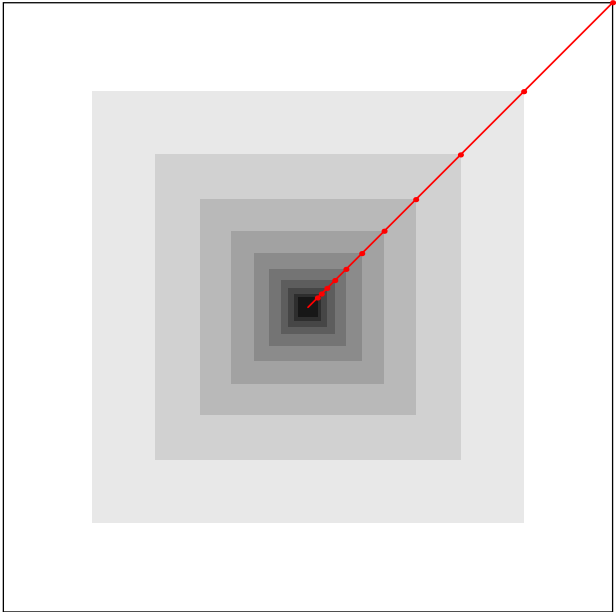
% SW branch
fill (0,0)--(w,0)--(w,0)--(w,w)--(0,w)--cycle;
width:=w/2;
height:=w;
x:=0;
y:=0;
for 1+0 upto n-1
  fill (0,0)--(width,0)--(width,height)--(0,height)
  --cycle shifted (x-w,width,y+height);
  B[]:=1/2[(x-w,width,y+height),(x-w,y)];
  x:=width;
  y:=y+height;
  width:=width/2;
endfor;

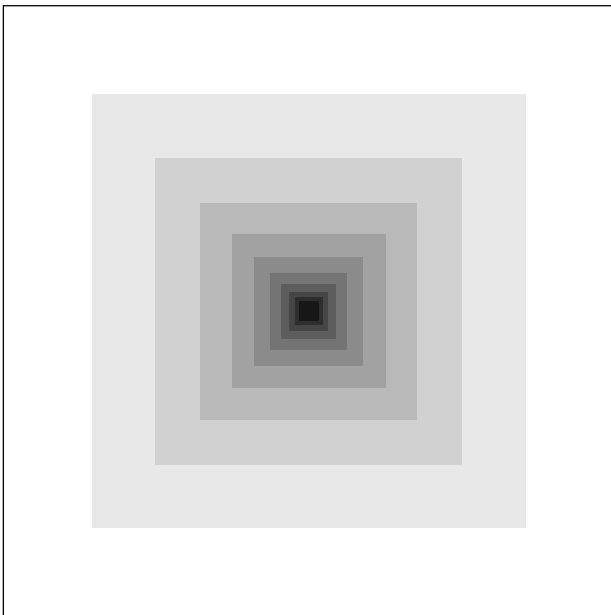
% NW branch
fill (0,0)--(w,0)--(w,0)--(w,w)--(0,w)--cycle;
width:=w/2;
height:=w;
x:=0;
y:=w;
for 1+0 upto n-1
  fill (0,0)--(width,0)--(width,height)--(0,height)
  --cycle shifted (x-w,width,y+height);
  C[]:=1/2[(x-w,width,y+height),(x-w,y)];
  x:=width;
  y:=y+height;
  width:=width/2;
endfor;

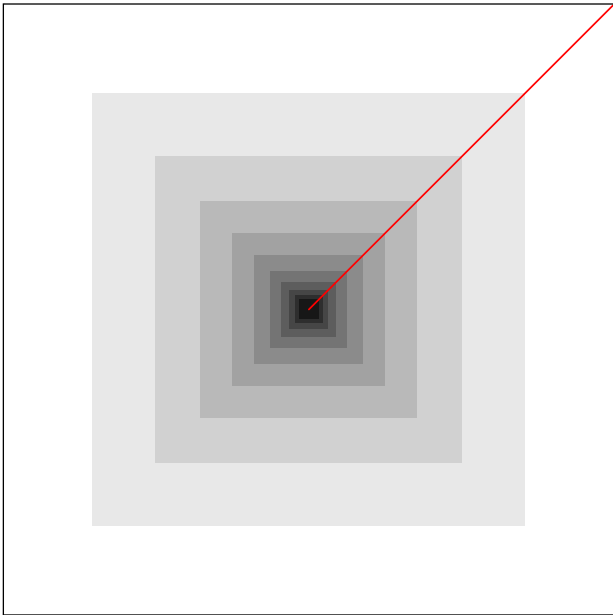
% NE branch
fill (0,0)--(w,0)--(w,0)--(w,w)--(0,w)--cycle;
width:=w/2;
height:=w;
x:=w;
y:=w;
for 1+0 upto n-1
  fill (0,0)--(width,0)--(width,height)--(0,height)
  --cycle shifted (x-w,width,y+height);
  D[]:=1/2[(x-w,width,y+height),(x,y+height)];
  x:=width;
  y:=y+height;
  width:=width/2;
endfor;

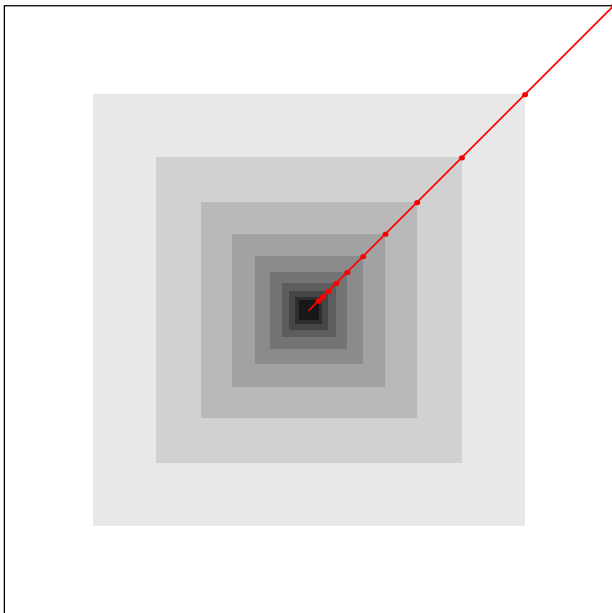
% red lines
pickup pencircle scaled 1;
for 1+0 upto n-1
  draw A[]--B[]--C[]--D[]--cycle withcolor red;
endfig;

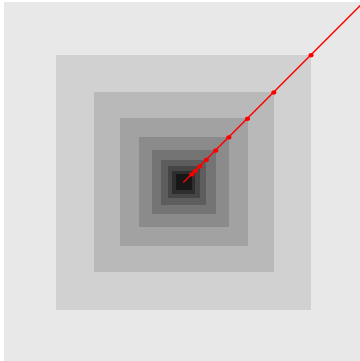
```







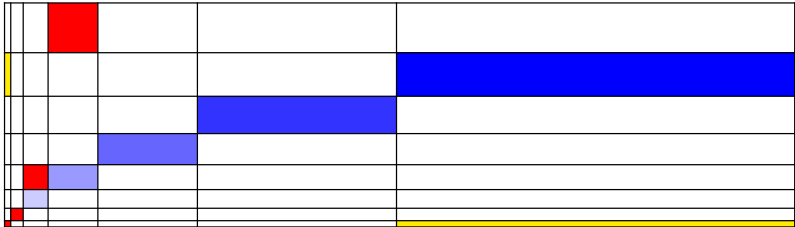




```

beginfig(4);
n:=11;
w:=50cm;
correction:=.75dp;
r:=.707; % =\sqrt{2}/2
d:=w*r;
for i=1 upto n-1;
  fill (d,d)--(-d,d)--(-d,-d)--(d,-d)--cycle
  withcolor ((n-i)/n);
  draw (d-correction,d-correction)
  withpen pencircle scaled 2pc
  withcolor red;
  d:=d*r;
endfor;
d:=w/3dp;
draw (d,d)--(-d,d)
  withpen pencircle scaled 4
  withcolor red;
draw (d,d)--(-d,d)--(-d,-d)--(d,-d)--cycle
  withpen pensquare scaled 6dp;
draw (d,d)
  withpen pencircle scaled 2pc
  withcolor red;
endfig;

```

```

color yellow;
yellow := (255/255,255/255,0/255);
beginfig(5);
n:=7;
w:=1px;
azero:=w;
r:=w;
bzero:=w;
q:=2;
x:=0;
y:=0;
width:=0;
height:=0;
picloop penname scaled 3.4;
width:=azero*(1+q**(n-1))/(1+q);
height:=(2+azero*(n+r)*(n+1))/2;
% paint square on red
fill (0,0)--(w,0)--(w,w)--(0,w)--cycle
withcolor red;
fill (w,0)--(2w,0)--(2w,w)--(w,w)--cycle
withcolor red;

```

```

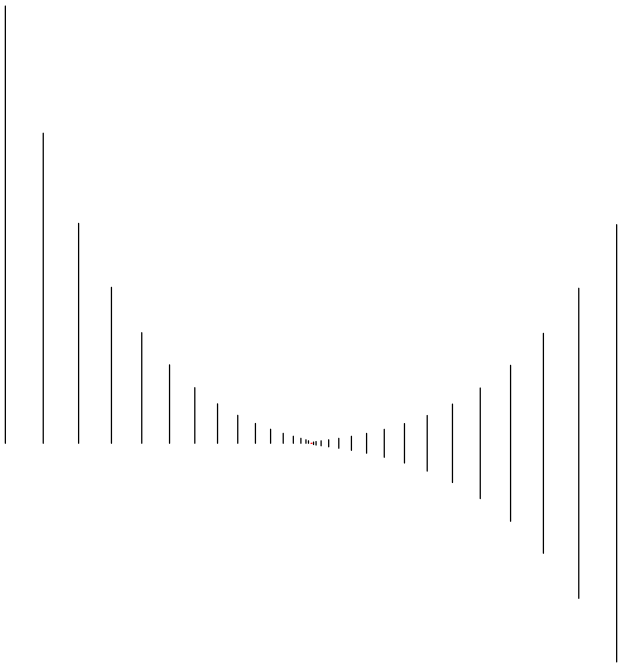
fill (2w,0)--(7w,0)--(7w,10w)--(2w,10w)--cycle
withcolor red;
fill (7w,20w)--(15w,20w)--(15w,30w)--(7w,30w)--cycle
withcolor red;
% paint diagonal in different shades of blue
fill (2w,w)--(7w,w)--(7w,6w)--(2w,6w)--cycle
withcolor .8[blue,white];
fill (7w,6w)--(15w,6w)--(15w,10w)--(7w,10w)--cycle
withcolor .6[blue,white];
fill (15w,10w)--(23w,10w)--(23w,15w)--(15w,15w)--cycle
withcolor .4[blue,white];
fill (23w,15w)--(31w,15w)--(31w,23w)--(23w,23w)--cycle
withcolor .2[blue,white];
fill (31w,23w)--(127w,23w)--(127w,28w)--(31w,28w)--cycle
withcolor .0[blue,white];
% add some yellow accents
fill (0,22w)--(1w,22w)--(1w,20w)--(0,20w)--cycle
withcolor yellow;
fill (62w,0)--(127w,0)--(127w,w)--(62w,w)--cycle
withcolor yellow;

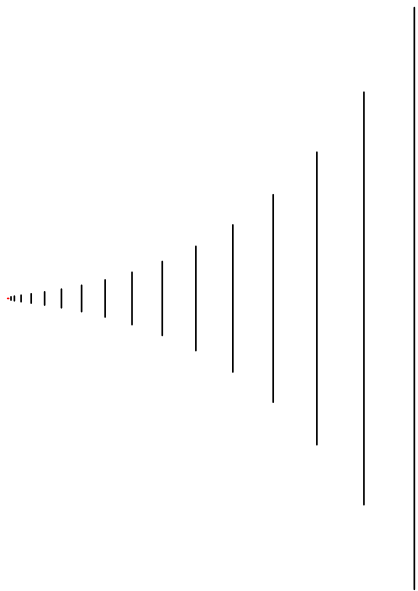
```

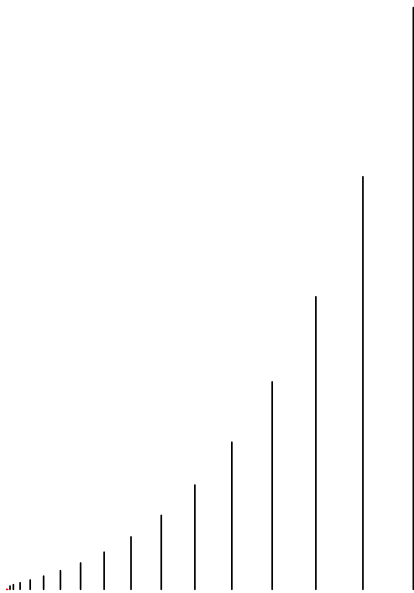
```

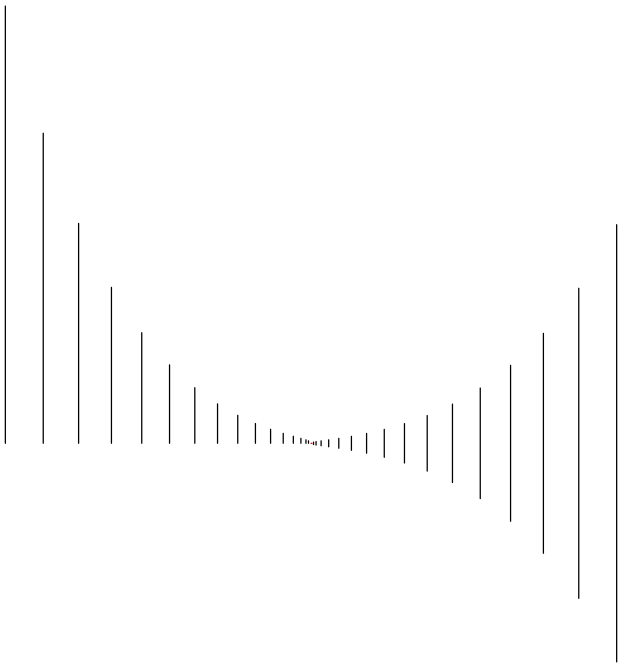
% make grid
draw (0,height)--(0,0)--(width,0);
x:=+azero;
y:=+azero;
for i=0 upto n-1;
draw (x,0)--(x,height);
bzero:=azero*q; % geometric progression
x:=+azero;
draw (0,y)--(width,y);
azero:=azero*r; % arithmetic progression
y:=+azero;
endfor;
draw (0,y)--(width,y);
azero:=azero+r;
endfig;

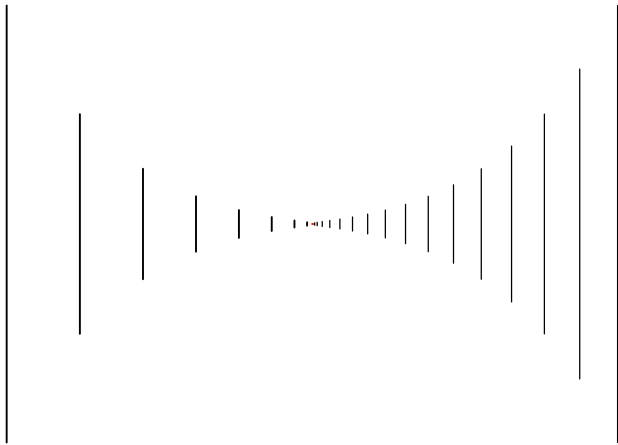
```

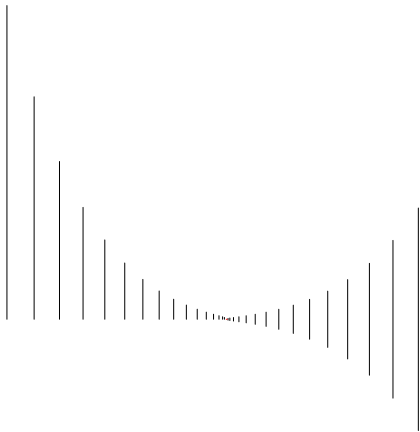












```
beginfig(0);  
n:=20;  
w:=1pc;  
r:=w;  
q:=2.41;  
height:=w;  
exp:=w;  
x:=exp;  
pickup pencircle scaled 6;  
for i=0 upto n-1:  
  draw (x,-.5*height)--(.5*height); % right side  
  draw (-x,0)--(-x,height); % left side  
  height:=height*q; % geometric progression  
  x:=x*exp;  
  exp:=exp*r; % arithmetic progression  
endfor;  
draw (0,0) withcolor red;  
endfig;
```





```

beginFig(0);
nc*7;
nc:=a;
r:=u;
widthblack:=u;
widthwhite:=u;
height:=u;
x:=0;
y:=0;
sep:=1.5*height;
% just to add white margin
max:=0.5*height+((1+nc)/2+((nc-1)-1))=u;
fill (-0.5*height,sep-0.5*height)--(max,sep-0.5*height)--
(max,1.5*height)--(-0.5*height,1.5*height)--cycle withcolor white;

% drawing bars
for i=0 upto n-1:
    fill (0,0)--(widthblack,0)--(widthblack,height)--
    (0,height)--cycle shifted (x,0);
    x:=widthblack;
    widthblack:=widthblack*q; % geometric progression
    fill (0,0)--(widthwhite,0)--(widthwhite,height)--
    (0,height)--cycle shifted (x,sep);
    x:=widthwhite;
    widthwhite:=widthwhite+r; % arithmetic progression
endfor;
fill (0,0)--(widthblack,0)--(widthblack,height)--
(0,height)--cycle shifted (x,0);
x:=widthblack;
widthblack:=widthblack*q; fill (0,0)--(widthwhite,0)--(widthwhite,height)--(0,height)--cycle shifted
(x,sep); x:=widthwhite; widthwhite:=widthwhite+r; endfor; fill (0,0)--(widthblack,0)--(widthblack,height)--
(0,height)--cycle shifted (x,0); x:=widthblack; draw (x,sep-hN)*(.15h,-.2*height);
endFig;

```

There is enough material for the [exhibition](#) in a small / medium size [art gallery](#); if someone is interested – just let me know.

The thoughts which appears while working on the project lead me to an idea of teaching basics of programming to elementary and secondary school pupils based on the METAPOST language.

The main advantage here is the almost obvious debugging: if the generated picture is not as planned, that something is wrong.

Coauthors of such a schoolbook are welcome.

Thank you for your attention

All comments are welcome.