Supplement to: Inferring Graphics Programs from Images

Anonymous Author(s)
Affiliation
Address
email

1 Neural network architecture

1.1 Convolutional network

The convolutional network takes as input 2 256×256 images represented as a $2 \times 256 \times 256\times$ volume. These are passed through two layers of convolutions separated by ReLU nonlinearities and max pooling:

- Layer 1: 20 8×8 convolutions, 2 16×4 convolutions, 2 4×16 convolutions. Followed by 8×8 pooling with a stride size of 4.
- Layer 2: 10 8×8 convolutions. Followed by 4×4 pooling with a stride size of 4.

Training takes a little bit less than a day on a Nvidia TitanX GPU. The network was trained on 10^5 synthetic examples.

1.2 Autoregressive decoding of drawing commands

Given the image features f, we predict the first token using logistic regression:

$$\mathbb{P}[T_1] \propto W_{T_1} f \tag{1}$$

where W_{T_1} is a learned weight matrix.

Subsequent tokens are predicted as:

$$\mathbb{P}[T_n | T_{1:(n-1)}] \propto \text{MLP}_{T_1,n}(I \otimes \bigotimes_{j<n} \text{oneHot}(T_j)) \tag{2}$$

Thus each token of each drawing primitive has its own learned MLP. For predicting the coordinates of lines we found that using 32 hidden nodes with sigmoid activations worked well; for other tokens the MLP's are just logistic regression (no hidden nodes).

1.3 A learned likelihood surrogate

Our architecture for $L_{\text{learned}}(\text{render}(T_1)|\text{render}(T_2))$ has the same series of convolutions as the network that predicts the next drawing command. We train it to predict two scalars: $|T_1 - T_2|$ and $|T_2 - T_1|$. These predictions are made using linear regression from the image features followed by a ReLU nonlinearity; this nonlinearity makes sense because the predictions can never be negative but could be arbitrarily large positive numbers.

We train this network by sampling random synthetic scenes for T_1, and then perturbing them in small ways to produce T_2. We minimize the squared loss between the network's prediction and the ground truth symmetric differences. T_1 is rendered in a "simulated hand drawing" style which we describe next.

Submitted to 31st Conference on Neural Information Processing Systems (NIPS 2017). Do not distribute.

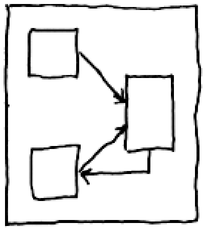
2 Simulating hand drawings

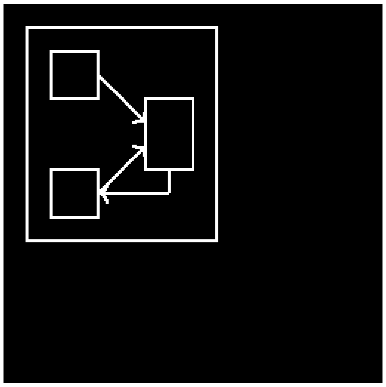
We introduce noise into the rendering process by:

- Rescaling the image intensity by a factor chosen uniformly at random from $[0.5, 1.5]$
- Translating the image by ±3 pixels chosen uniformly random
- Rendering the LaTeX using the `pencildraw` style, which adds random perturbations to the paths drawn by LaTeXin a way designed to resemble a pencil.
- Randomly perturbing the positions and sizes of primitive LaTeXdrawing commands

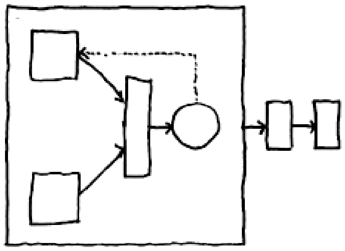
3 Full results on drawings data set

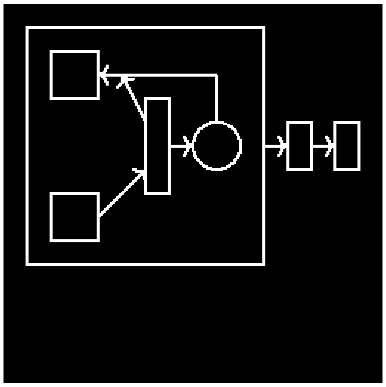
Below we show our full data set of drawings. The leftmost column is a hand drawing. The middle column is a rendering of the most likely trace discovered by the neurally guided SMC sampling scheme. The rightmost column is the program we synthesized from a ground truth execution trace of the drawing.



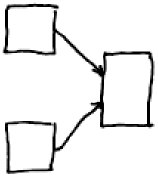


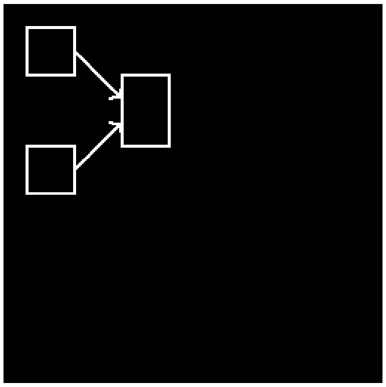
```
line(6,2,3,2,arrow = True,solid
line(6,2,6,3,arrow = False,solid
rectangle(0,0,8,9);
rectangle(5,3,7,6);
reflect(reflect(y = 9)){
line(3,2,5,4,arrow = True,solid
rectangle(1,6,3,8)
}
```



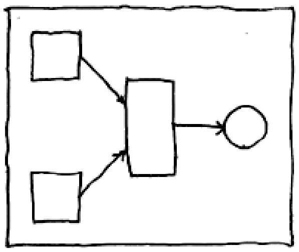


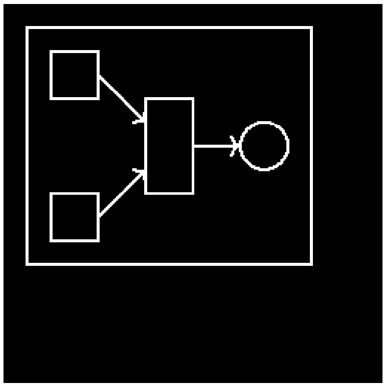
```
Solver timeout
```



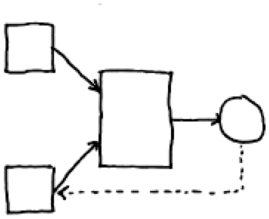


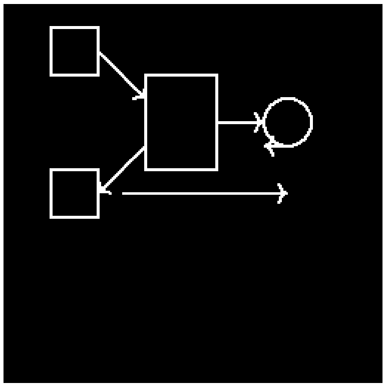
```
rectangle(4,2,6,5);
reflect(reflect(y = 7)){
line(2,6,4,4,arrow = True,solid
rectangle(0,5,2,7)
}
```



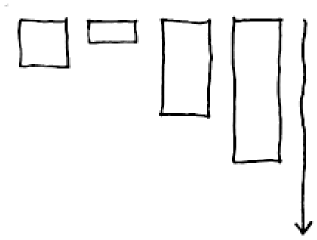


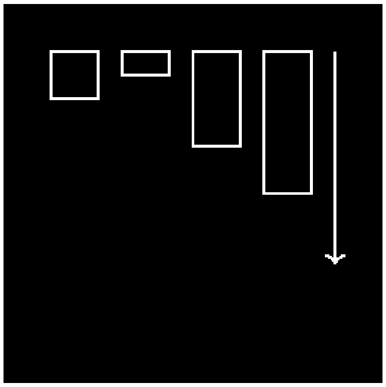
```
circle(10,5);
line(7,5,9,5,arrow = True,solid
rectangle(0,0,12,10);
reflect(reflect(y = 10)){
line(3,8,5,6,arrow = True,solid
rectangle(1,1,3,3);
rectangle(5,3,7,7)
}
```



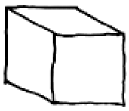


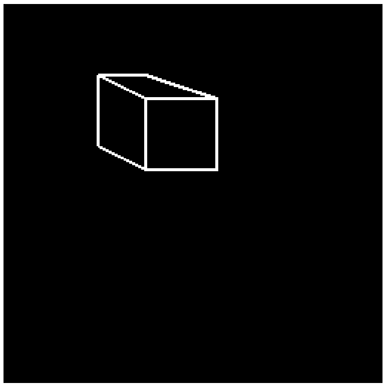
```
circle(10,4);
line(10,1,10,3,arrow = False,sol
line(10,1,2,1,arrow = True,solid
line(7,4,9,4,arrow = True,solid
reflect(reflect(y = 8)){
line(2,1,4,3,arrow = True,solid
rectangle(0,6,2,8);
rectangle(4,2,7,6)
}
```



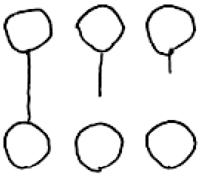


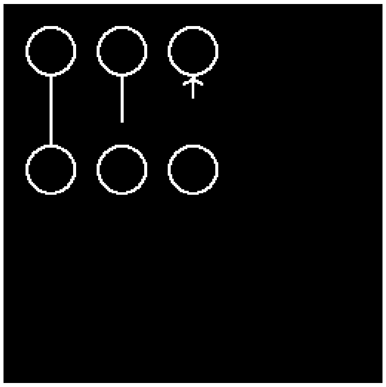
Solver timeout



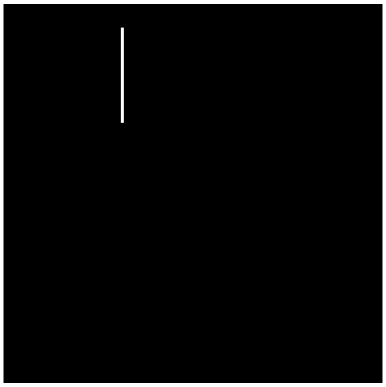


```
rectangle(2,0,5,3);
for (i < 3){
if (i > 0){
line(0,3*i + -2,3*i + -3,4,arrow
line(3*i + -3,4,3*i + -1,3,arrow
}line(0,1,2,0,arrow = False,soli
}
```

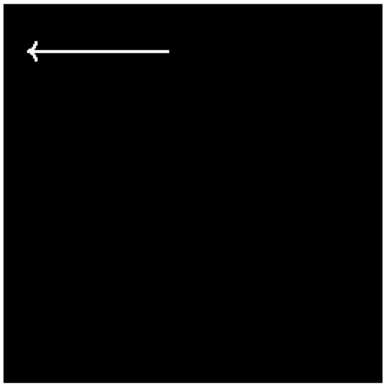




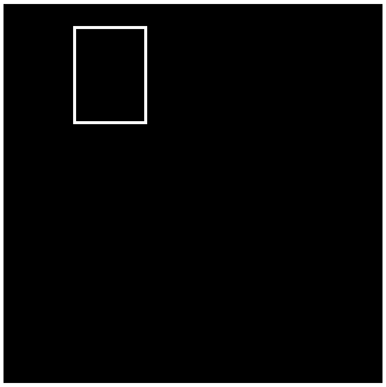
```
for (i < 3){
circle(-3*i + 7,6);
circle(-3*i + 7,1);
line(-3*i + 7,-1*i + 4,-3*i + 7,
}
```



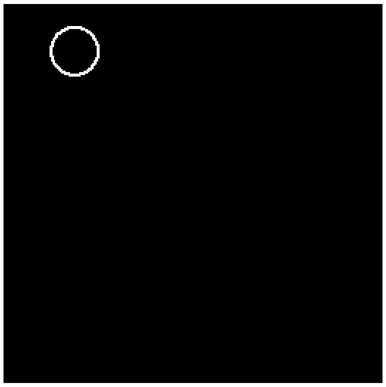
```
line(0,0,0,4,arrow = False,solid
```



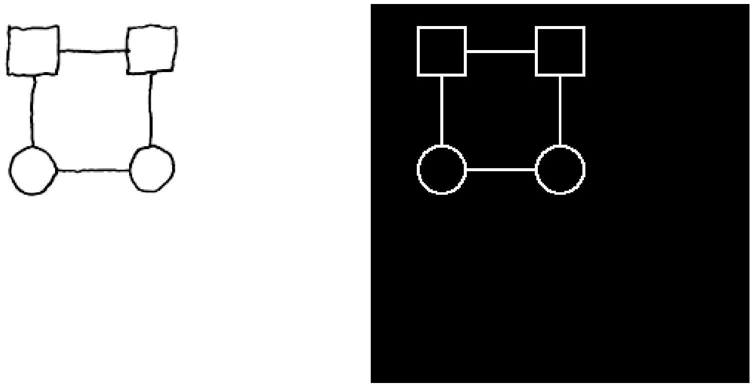
```
line(6,0,0,0,arrow = True,solid
```



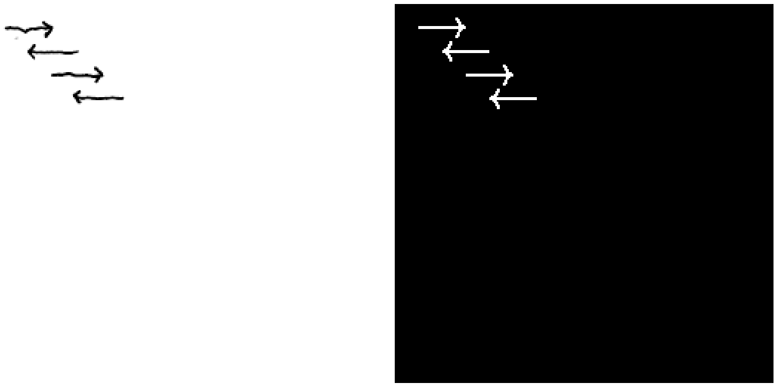
```
rectangle(0,0,3,4)
```



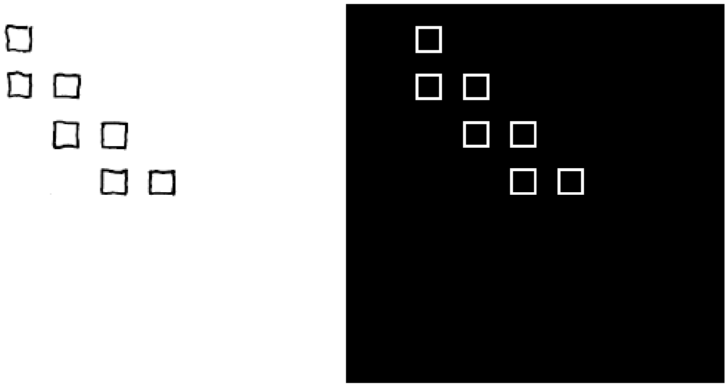
```
circle(1,1)
```



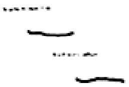
```
line(2,1,5,1,arrow = False,solid
reflect(reflect(x = 7)){
circle(1,1);
line(2,6,5,6,arrow = False,solid
line(1,2,1,5,arrow = False,solid
rectangle(0,5,2,7)
}
```

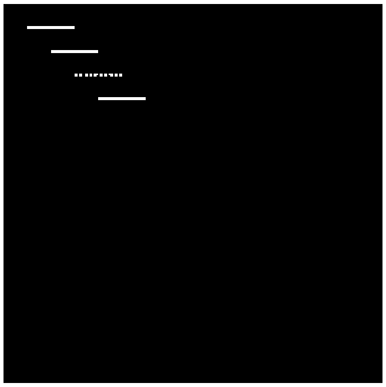


```
line(3,2,1,2,arrow = True,solid
line(5,0,3,0,arrow = True,solid
line(2,1,4,1,arrow = True,solid
line(0,3,2,3,arrow = True,solid
```

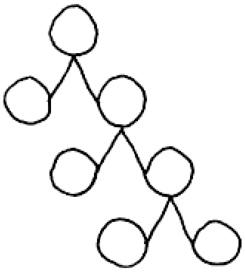


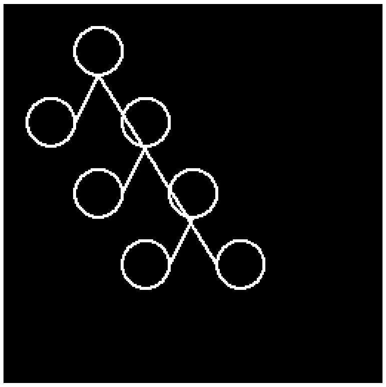
```
for (i < 4){
if (i > 0){
rectangle(2*i + -2,-2*i + 6,2*i
}rectangle(2*i,-2*i + 6,2*i + 1,
}
```



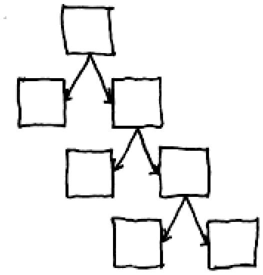


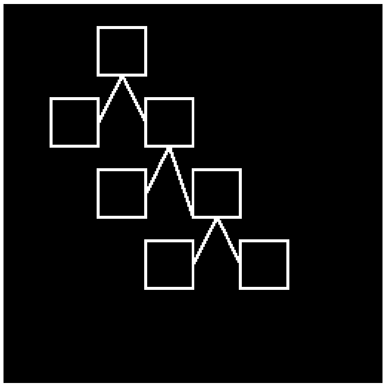
```
line(0,3,2,3,arrow = False,solid
line(2,1,4,1,arrow = False,solid
line(1,2,3,2,arrow = False,solid
line(3,0,5,0,arrow = False,solid
```



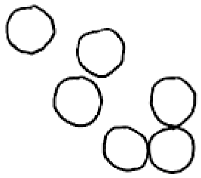


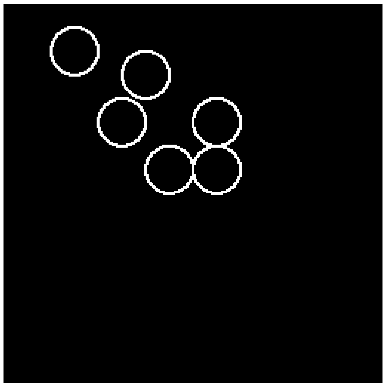
```
Solver timeout
```



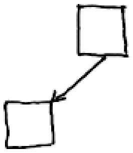


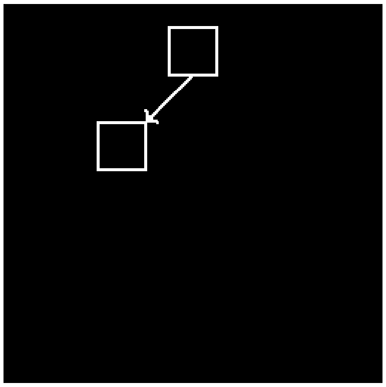
```
for (i < 4){
if (i > 0){
line(-2*i + 9,3*i,-2*i + 10,3*i
line(-2*i + 9,3*i,-2*i + 8,3*i +
rectangle(-2*i + 6,3*i + -3,-2*i
}rectangle(-2*i + 8,3*i,-2*i + 1
}
```



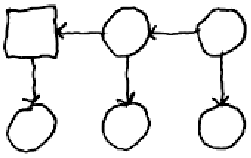


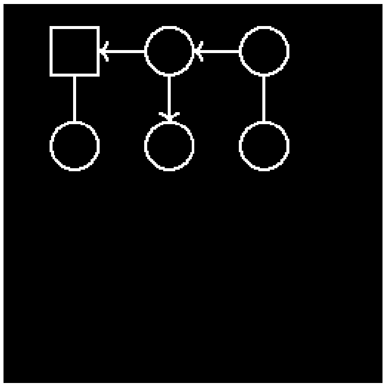
```
Solver timeout
```



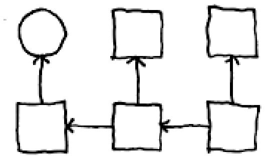


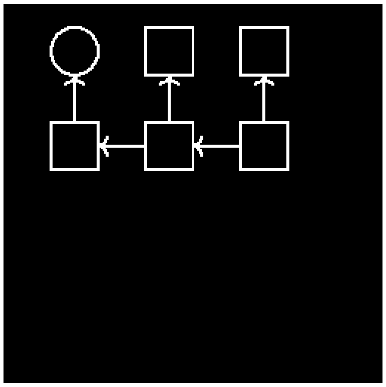
```
line(4,4,2,2,arrow = True,solid
rectangle(0,0,2,2);
rectangle(3,4,5,6)
```



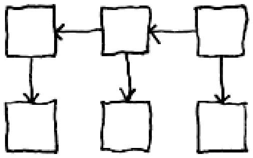


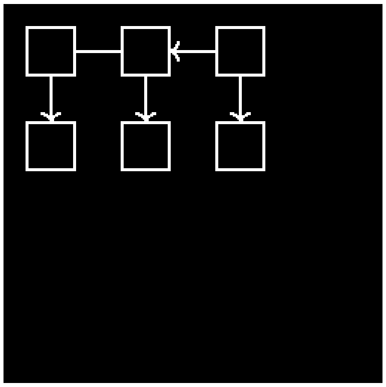
```
Solver timeout
```



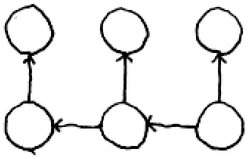


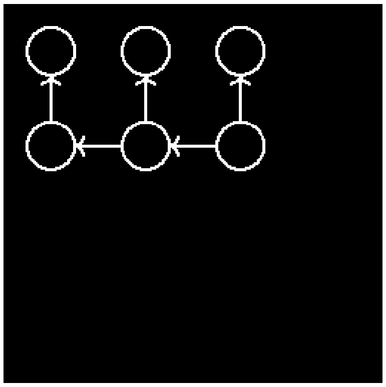
```
circle(1,5);
for (i < 3){
if (i > 0){
line(-4*i + 12,1,-4*i + 10,1,arr
rectangle(-4*i + 12,4,-4*i + 14,
}line(-4*i + 9,2,-4*i + 9,4,arro
rectangle(-4*i + 8,0,-4*i + 10,2
}
```



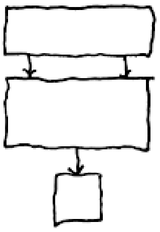


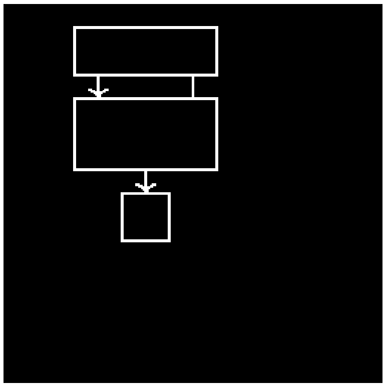
```
for (i < 3){
if (i > 0){
line(4*i,5,4*i + -2,5,arrow = Tr
}line(4*i + 1,4,4*i + 1,2,arrow
rectangle(4*i,0,4*i + 2,2);
rectangle(4*i,4,4*i + 2,6)
}
```



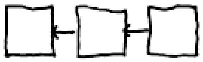


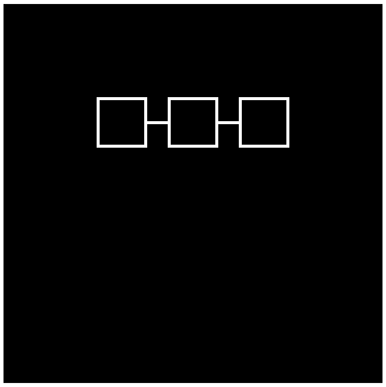
```
for (i < 3){
line(-4*i + 9,2,-4*i + 9,4,arrow
for (j < 2){
circle(-4*i + 9,4*j + 1);
line(-4*j + 8,1,-4*j + 6,1,arrow
}
}
```



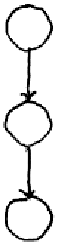


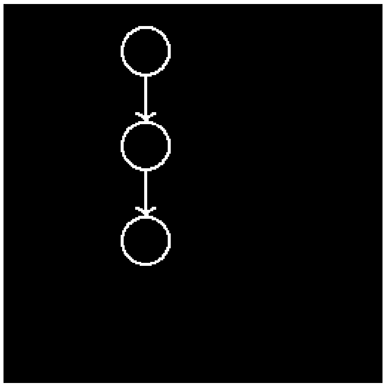
```
reflect(reflect(x = 6)){
for (i < 3){
if (i > 0){
line(-2*i + 7,-4*i + 11,-2*i + 7
rectangle(0,-4*i + 11,6,-3*i + 1
}rectangle(2,0,4,2)
}
}
```



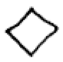


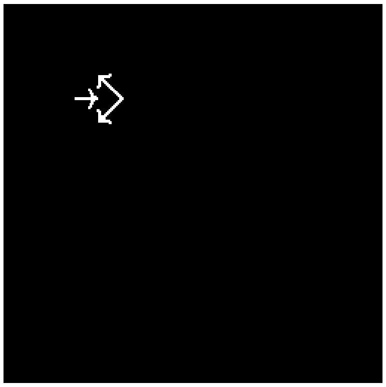
```
for (i < 3){
if (i > 0){
line(-3*i + 9,1,-3*i + 8,1,arrow
}rectangle(-3*i + 6,0,-3*i + 8,2
}
```





```
line(1,3,1,4,arrow = False,solid
for (i < 3){
if (i > 0){
line(1,-5*i + 13,1,-4*i + 10,arr
}circle(1,-4*i + 9)
}
```

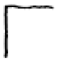


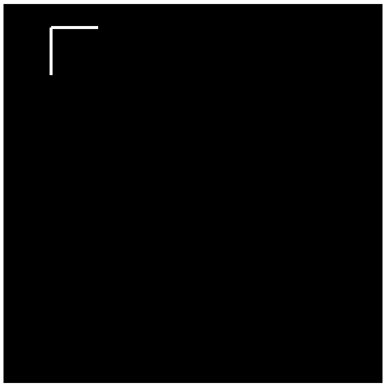


94

```
reflect(reflect(x = 2)){
line(0,1,1,2,arrow = False,solid
line(1,0,2,1,arrow = False,solid
}
```

95

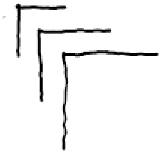


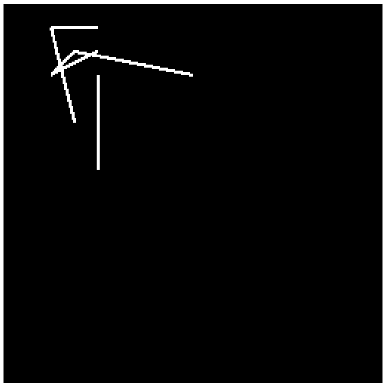


96

```
line(0,2,2,2,arrow = False,solid
line(0,0,0,2,arrow = False,solid
```

97

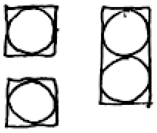


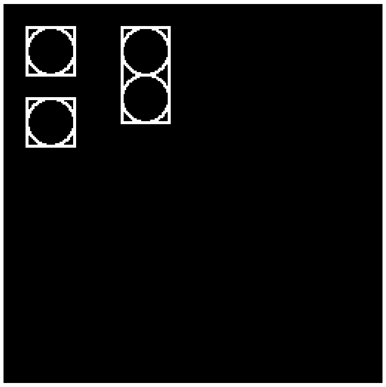


98

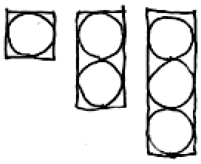
```
for (i < 3){
line(-1*i + 2,i + 4,-2*i + 6,i +
line(-1*i + 2,2*i,-1*i + 2,i + 4
}
```

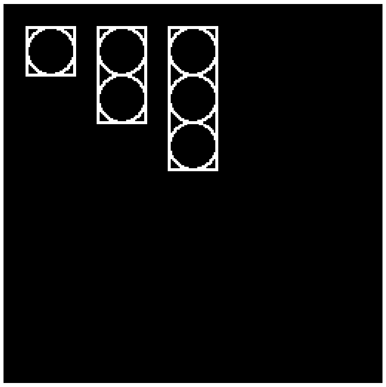
99



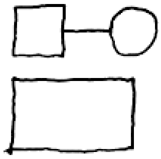


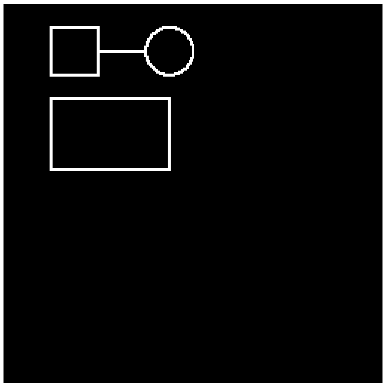
```
for (i < 3){
if (i > 0){
circle(5,2*i);
circle(1,-3*i + 7);
rectangle(0,-3*i + 6,2,-3*i + 8)
}rectangle(4,1,6,5)
}
```



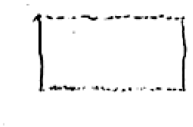


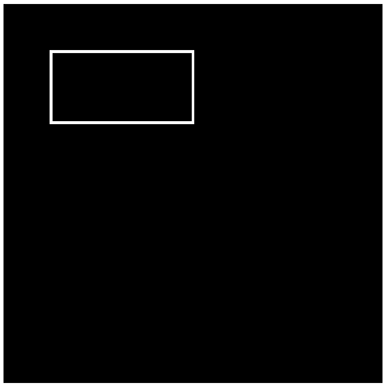
```
for (i < 3){
rectangle(3*i,-2*i + 4,3*i + 2,6
for (j < i + 1){
circle(3*i + 1,-2*j + 5)
}
}
```



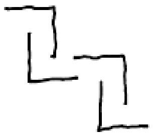


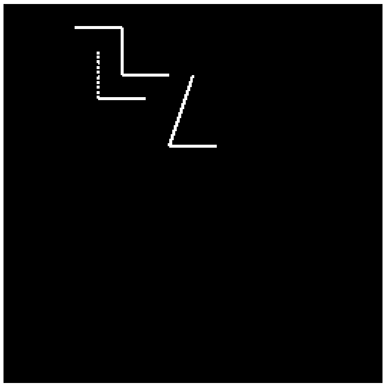
```
circle(5,5);
line(2,5,4,5,arrow = False,solid
rectangle(0,0,5,3);
rectangle(0,4,2,6)
```



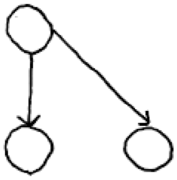


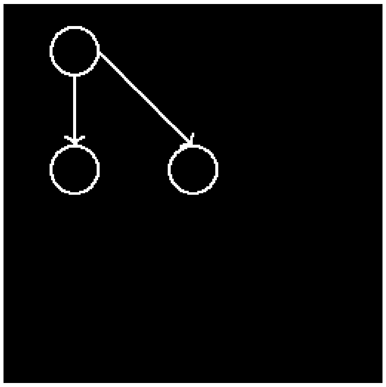
```
line(0,0,6,0,arrow = False,solid
line(0,3,6,3,arrow = False,solid
line(6,0,6,3,arrow = False,solid
line(0,0,0,3,arrow = False,solid
```



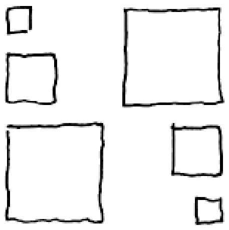


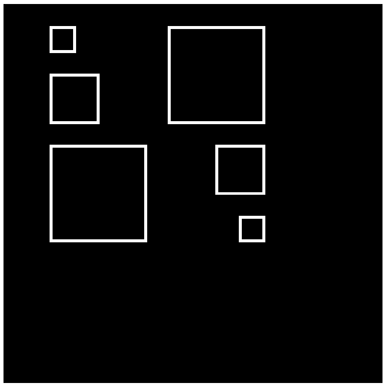
```
Solver timeout
```



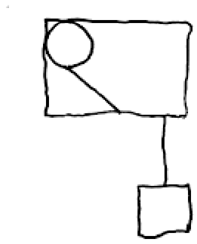


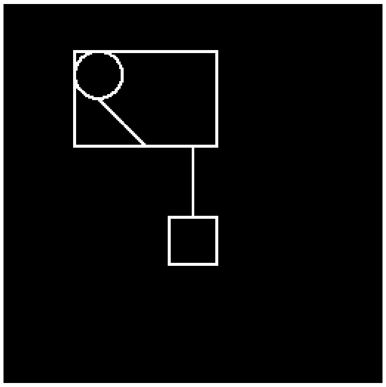
```
for (i < 3){
if (i > 0){
circle(5*i + -4,1);
line(i,i + 4,5*i + -4,2,arrow =
}circle(1,6)
}
```



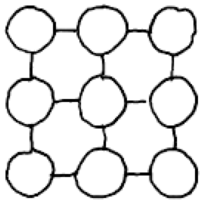


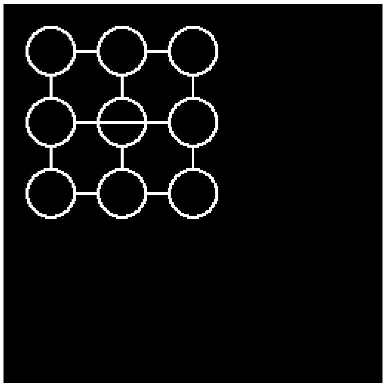
Solver timeout



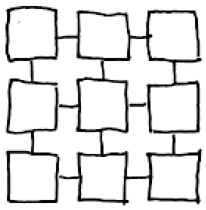


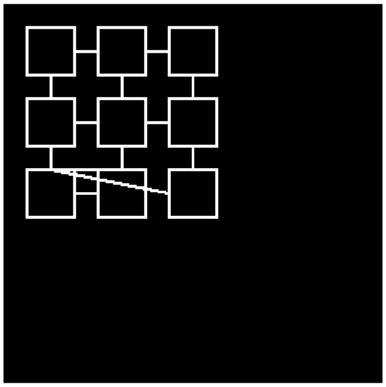
```
for (i < 3){
if (i > 0){
line(-4*i + 9,5*i + -3,-2*i + 7,
rectangle(-4*i + 8,5*i + -5,6,7*
}circle(1,8)
}
```



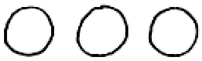


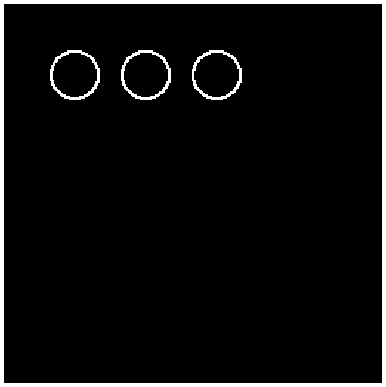
Solver timeout



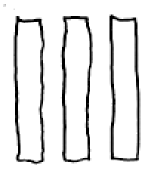


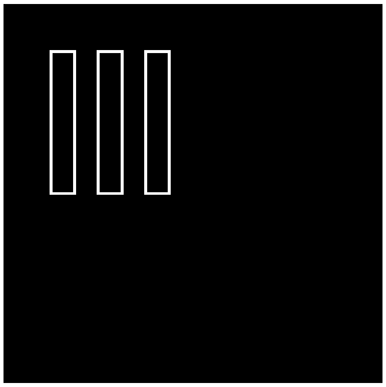
Solver timeout



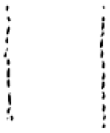


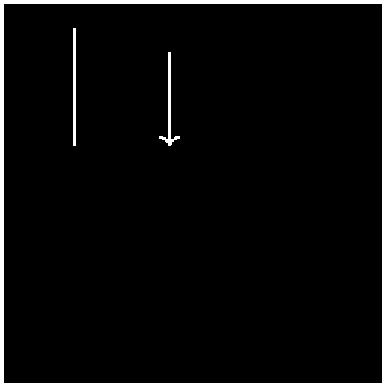
```
for (i < 3){
circle(3*i + 1,1)
}
```



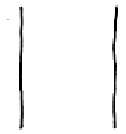


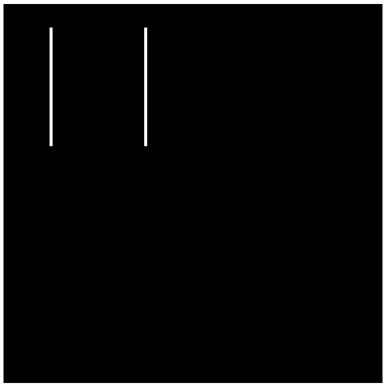
```
for (i < 3){
rectangle(2*i,0,2*i + 1,6)
}
```



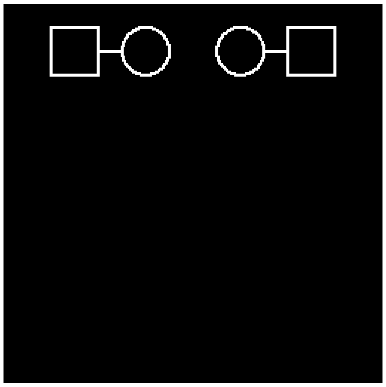


```
line(0,0,0,5,arrow = False,solid
line(4,1,4,5,arrow = False,solid
line(4,0,4,1,arrow = False,solid
```

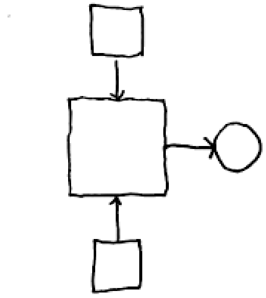


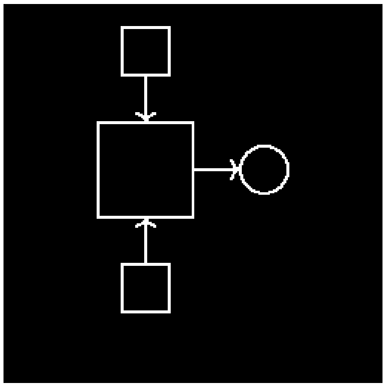


```
line(0,0,0,5,arrow = False,solid
line(4,0,4,5,arrow = False,solid
```

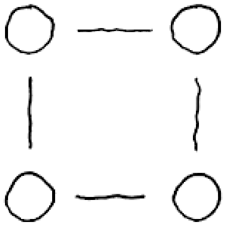


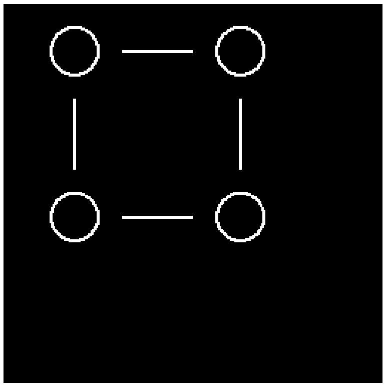
```
reflect(reflect(x = 12)){
circle(4,1);
line(9,1,10,1,arrow = False,soli
rectangle(10,0,12,2)
}
```



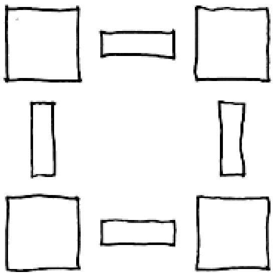


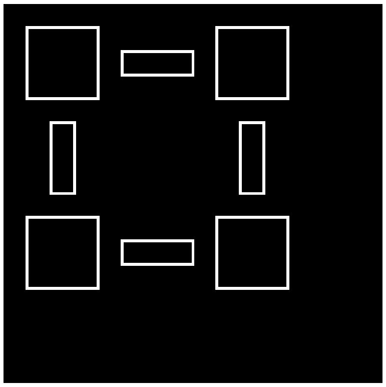
```
line(4,6,6,6,arrow = True,solid
rectangle(0,4,4,8);
reflect(reflect(y = 12)){
circle(7,6);
line(2,10,2,8,arrow = True,solid
rectangle(1,0,3,2)
}
```



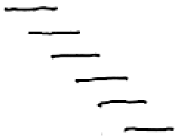


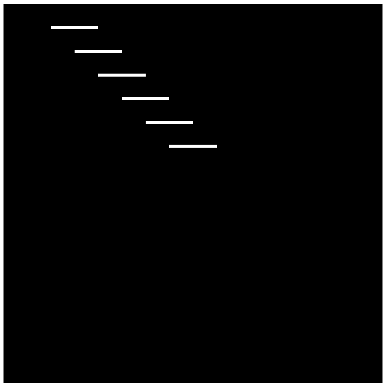
```
reflect(reflect(x = 9)){
line(1,3,1,6,arrow = False,solid
reflect(reflect(y = 9)){
circle(1,8);
line(3,1,6,1,arrow = False,solid
}
}
```



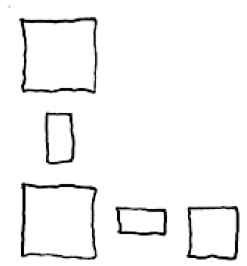


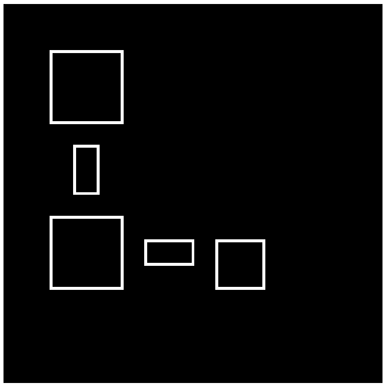
```
reflect(reflect(y = 11)){
rectangle(4,1,7,2);
reflect(reflect(x = 11)){
rectangle(1,4,2,7);
rectangle(0,8,3,11)
}
}
```



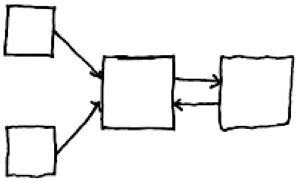


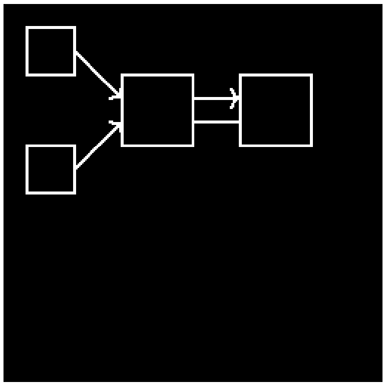
```
for (i < 3){
line(-2*i + 5,2*i,-2*i + 7,2*i,a
line(-2*i + 4,2*i + 1,-2*i + 6,2
}
```



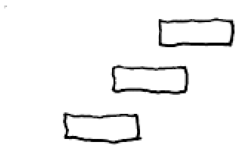


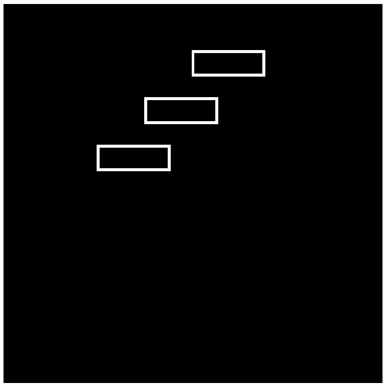
```
for (i < 3){
if (i > 0){
rectangle(-3*i + 10,i + -1,-3*i
rectangle(0,-7*i + 14,3,-7*i + 1
}rectangle(1,4,2,6)
}
```



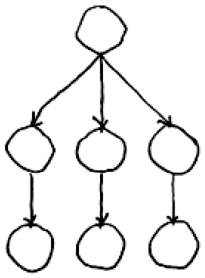


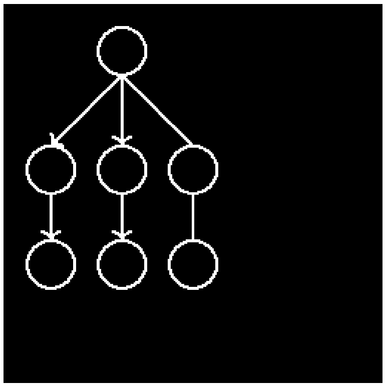
Solver timeout



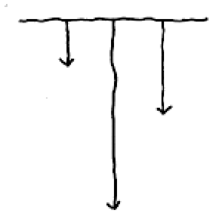


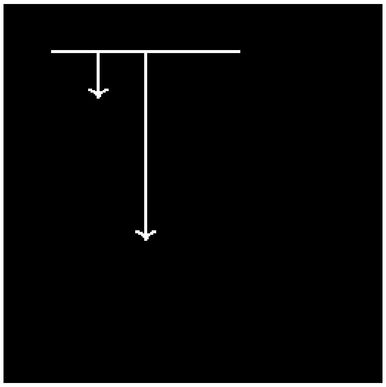
```
for (i < 3){
rectangle(2*i,2*i,2*i + 3,2*i +
}
```



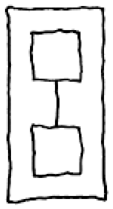


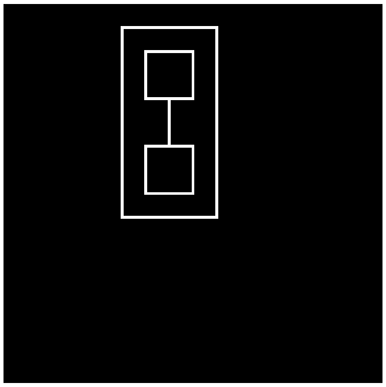
```
circle(4,10);
for (i < 3){
circle(3*i + 1,5);
circle(3*i + 1,1);
line(4,9,3*i + 1,6,arrow = True,
line(3*i + 1,4,3*i + 1,2,arrow =
}
```



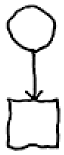


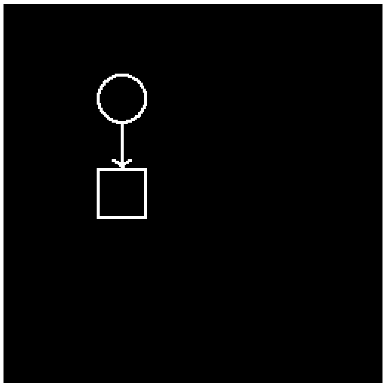
```
line(6,8,6,4,arrow = True,solid
line(0,8,8,8,arrow = False,solid
line(4,8,4,0,arrow = True,solid
line(2,8,2,6,arrow = True,solid
```



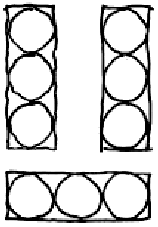


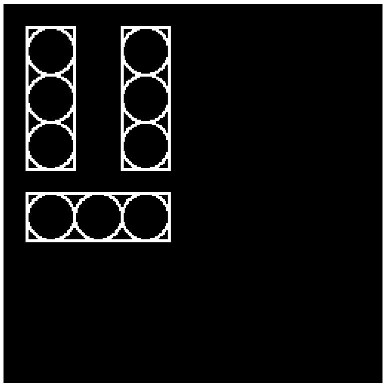
```
line(2,3,2,5,arrow = False,solid
rectangle(0,0,4,8);
reflect(reflect(y = 8)){
rectangle(1,5,3,7)
}
```



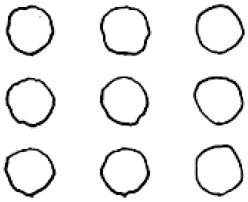


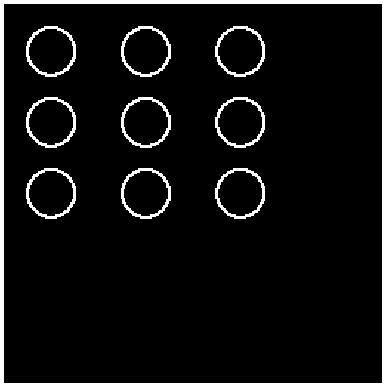
```
circle(1,5);
line(1,4,1,2,arrow = True,solid
rectangle(0,0,2,2)
```



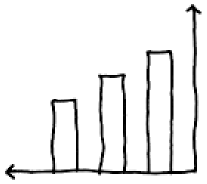


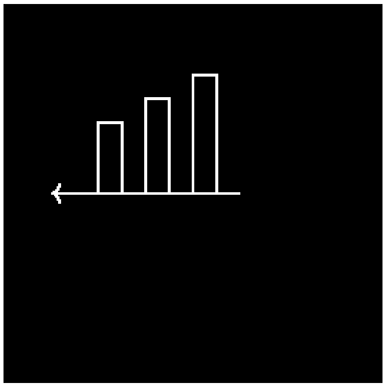
```
Solver timeout
```



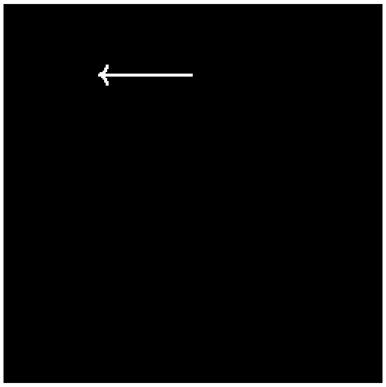


```
for (i < 3){
for (j < 3){
circle(-4*i + 9,-3*j + 7)
}
}
```

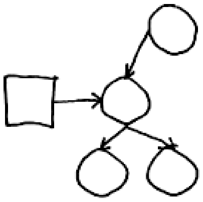


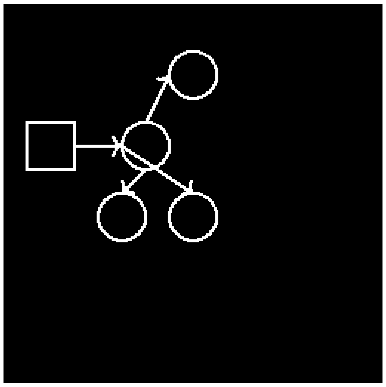


```
for (i < 3){
if (i > 0){
line(8,0,8*i + -8,7*i + -7,arrow
}rectangle(-2*i + 6,0,-2*i + 7,-
}
```

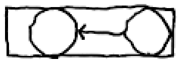


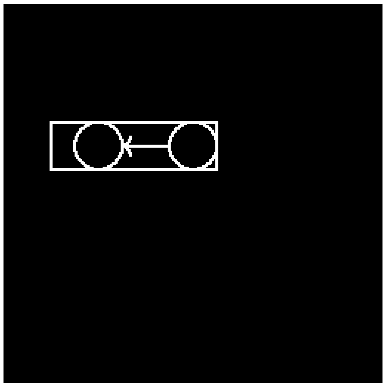
```
line(4,0,0,0,arrow = False,solid
```



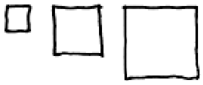


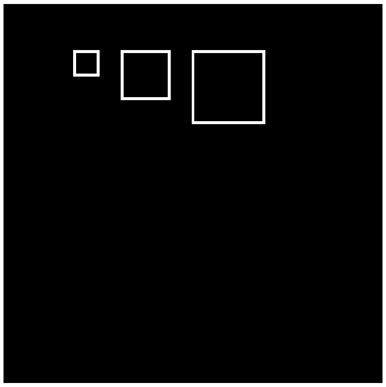
```
Solver timeout
```



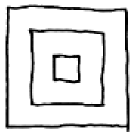


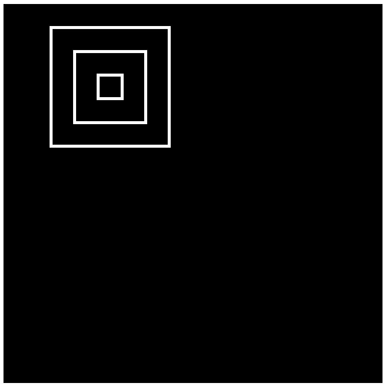
```
circle(2,1);
circle(6,1);
line(5,1,3,1,arrow = True,solid
rectangle(0,0,7,2)
```



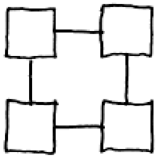


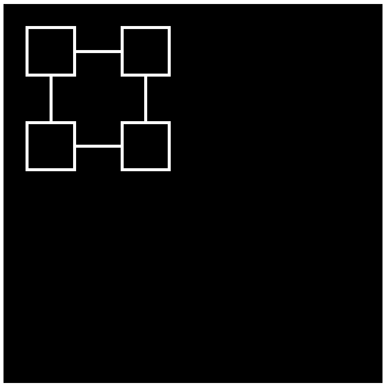
```
rectangle(5,0,8,3);
rectangle(0,2,1,3);
rectangle(2,1,4,3)
```



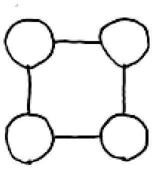


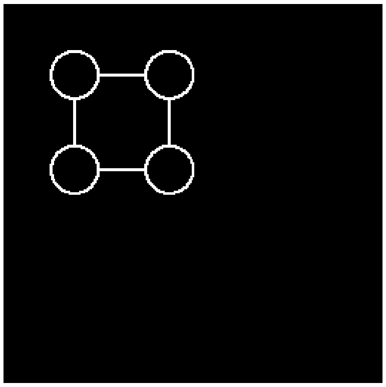
```
for (i < 3){
rectangle(i,i,-1*i + 5,-1*i + 5)
}
```



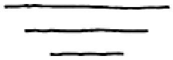


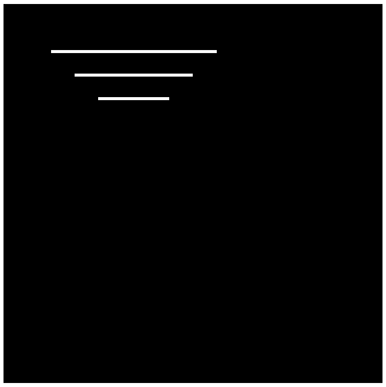
```
reflect(reflect(x = 6)){
line(5,2,5,4,arrow = False,solid
reflect(reflect(y = 6)){
line(2,1,4,1,arrow = False,solid
rectangle(4,4,6,6)
}
}
```



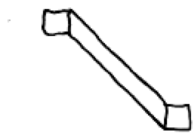


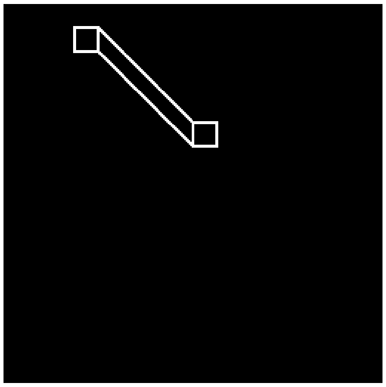
```
reflect(reflect(x = 6)){
line(1,2,1,4,arrow = False,solid
reflect(reflect(y = 6)){
circle(5,1);
line(2,1,4,1,arrow = False,solid
}
}
```



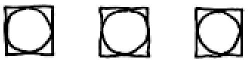


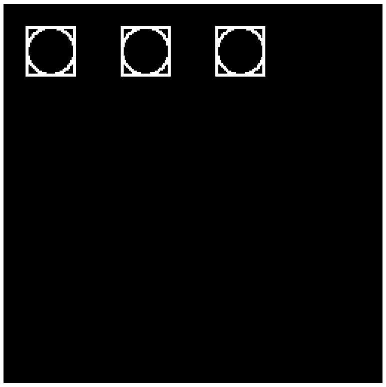
```
for (i < 3){
line(i,-1*i + 2,-1*i + 7,-1*i +
}
```



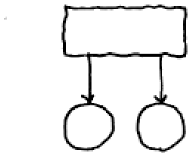


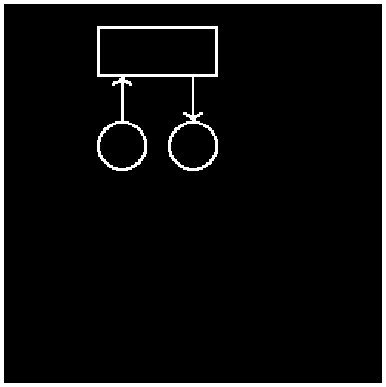
```
line(1,4,5,0,arrow = False,solid
line(1,5,5,1,arrow = False,solid
rectangle(0,4,1,5);
rectangle(5,0,6,1)
```



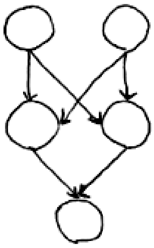


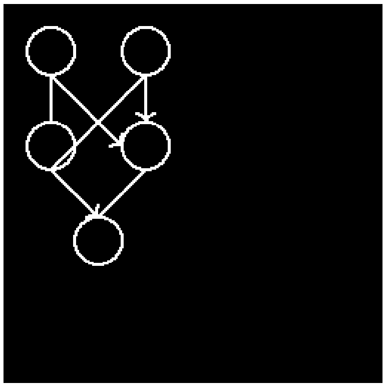
```
for (i < 3){
circle(4*i + 1,1);
rectangle(4*i,0,4*i + 2,2)
}
```



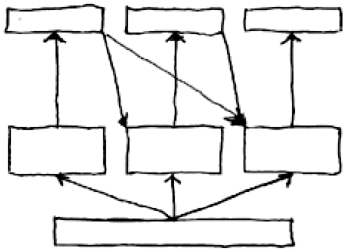


```
rectangle(0,4,5,6);
reflect(reflect(x = 5)){
circle(1,1);
line(1,4,1,2,arrow = True,solid
}
```



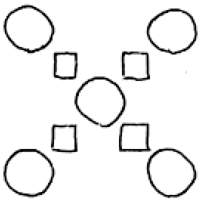


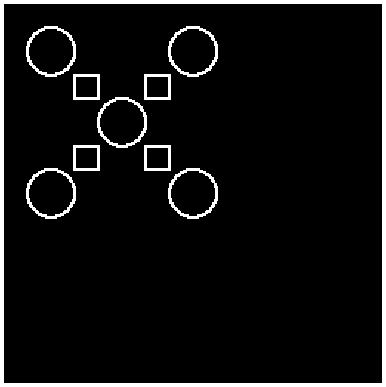
```
reflect(reflect(x = 6)){
circle(1,9);
for (i < 3){
if (i > 0){
circle(-2*i + 7,-4*i + 9);
line(5,-4*i + 12,-2*i + 7,-4*i +
}line(1,8,4,5,arrow = True,solid
}
}
```



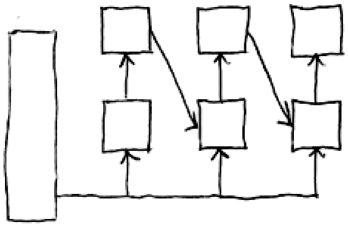
Sampled no finished traces.

```
Solver timeout
```



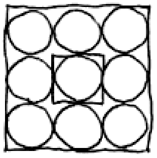


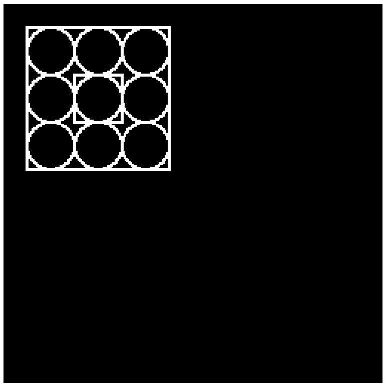
```
reflect(reflect(y = 8)){
for (i < 3){
if (i > 0){
rectangle(3*i + -1,-3*i + 8,3*i,
}circle(3*i + 1,-3*i + 7)
}
}
```



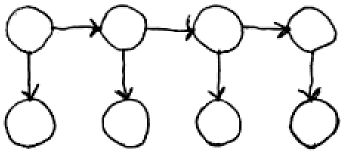
Sampled no finished traces.

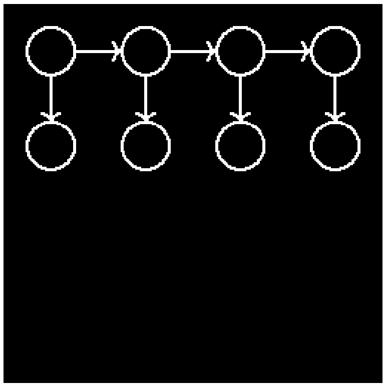
```
Solver timeout
```



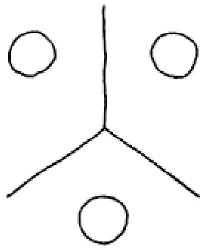


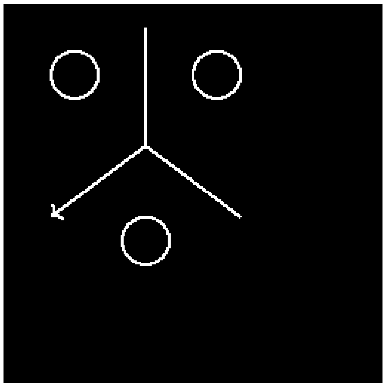
```
for (i < 3){
if (i > 0){
rectangle(2*i + -2,2*i + -2,-2*i
}for (j < 3){
circle(2*j + 1,2*i + 1)
}
}
```



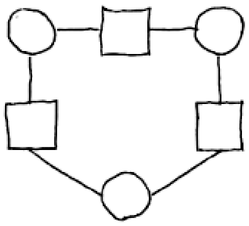


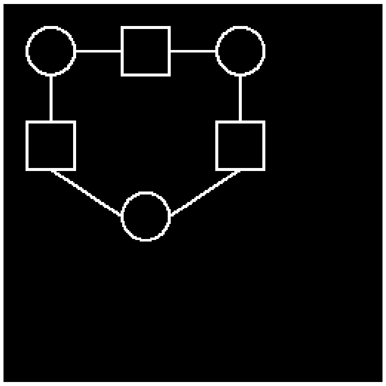
```
for (i < 4){
line(-4*i + 13,4,-4*i + 13,2,arr
for (j < 3){
if (j > 0){
circle(-4*i + 13,-4*j + 9)
}line(-4*j + 10,5,-4*j + 12,5,ar
}
}
```



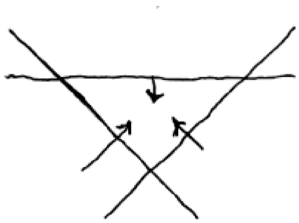


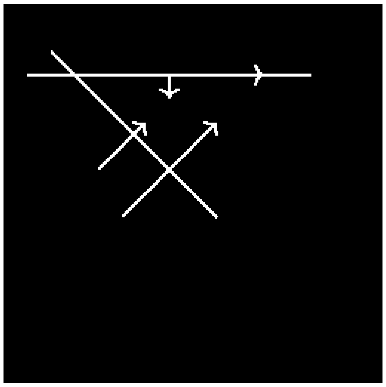
```
line(4,5,4,10,arrow = False,soli
reflect(reflect(x = 8)){
circle(4,1);
circle(1,8);
line(4,5,8,2,arrow = False,solid
}
```



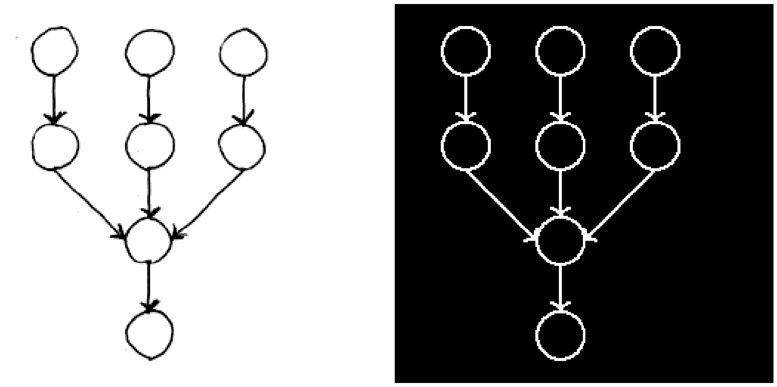


```
Solver timeout
```

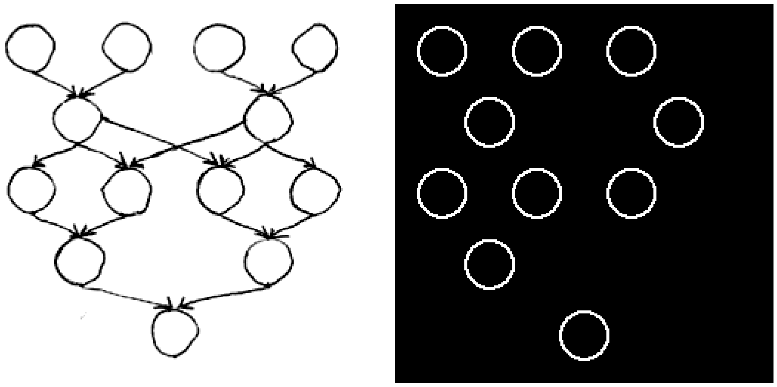




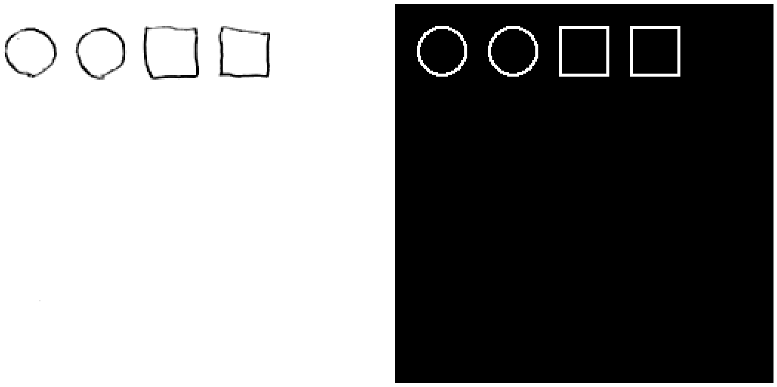
```
line(0,6,12,6,arrow = False,soli
line(8,3,7,4,arrow = True,solid
line(3,2,5,4,arrow = True,solid
line(6,6,6,5,arrow = True,solid
reflect(reflect(x = 12)){
line(4,0,12,8,arrow = False,soli
}
```



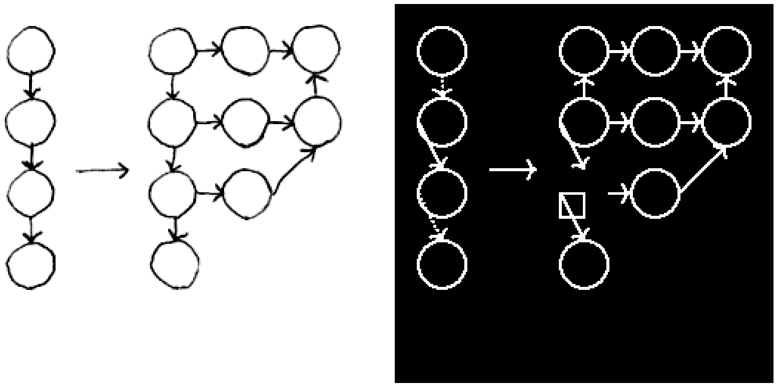
```
Solver timeout
```



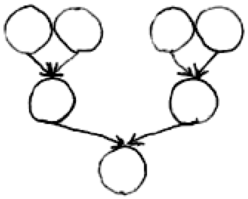
```
Solver timeout
```

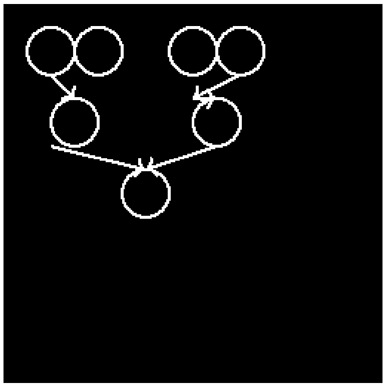


```
circle(1,1);
circle(4,1);
rectangle(9,0,11,2);
rectangle(6,0,8,2)
```

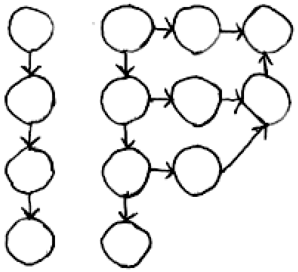


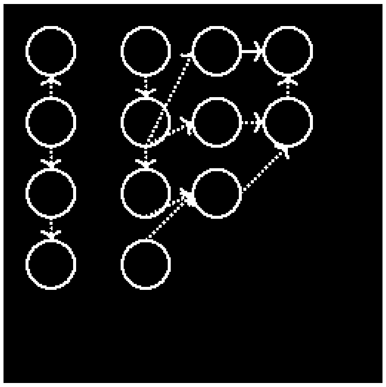
```
Solver timeout
```



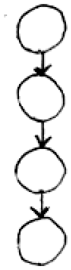


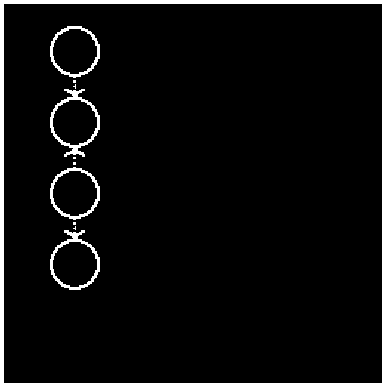
Solver timeout



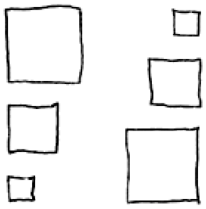


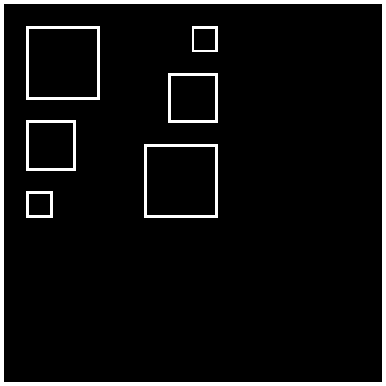
Solver timeout



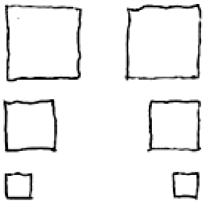


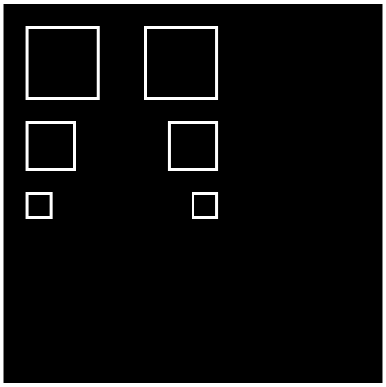
```
for (i < 4){
if (i > 0){
line(1,-3*i + 12,1,-3*i + 11,arr
}circle(1,-3*i + 10)
}
```



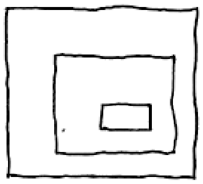


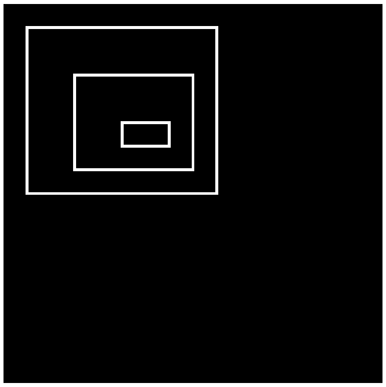
Solver timeout



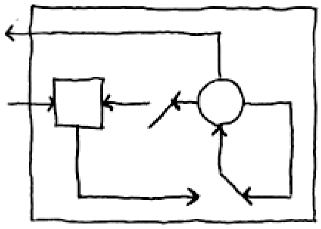


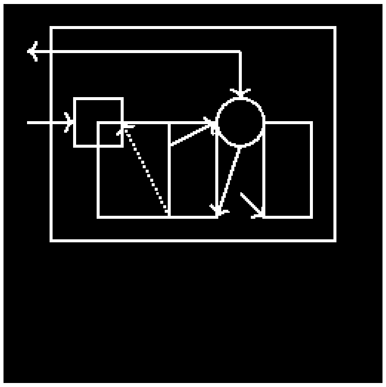
```
reflect(reflect(x = 8)){
rectangle(0,0,1,1);
rectangle(0,5,3,8);
rectangle(0,2,2,4)
}
```



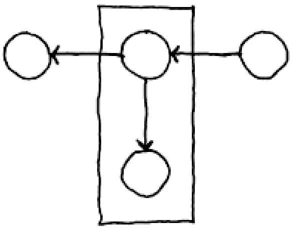


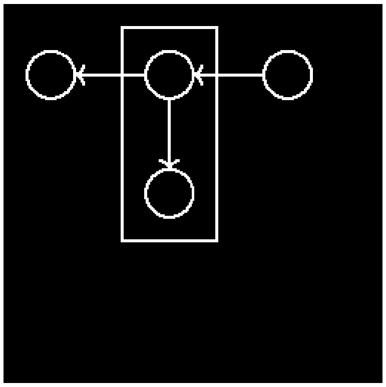
```
for (i < 3){
rectangle(2*i,i,-1*i + 8,-2*i +
}
```



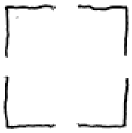


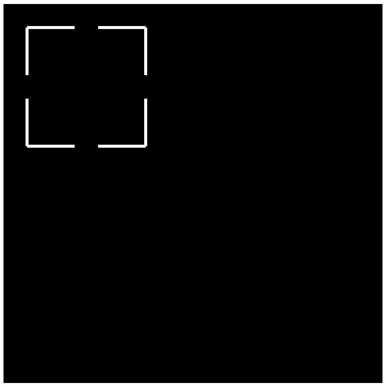
```
Solver timeout
```



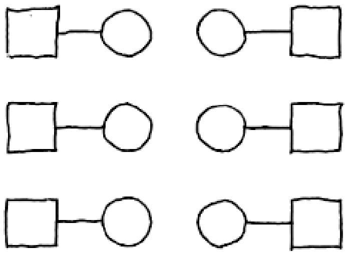


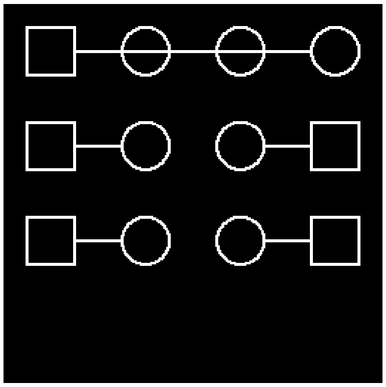
```
circle(6,2);
line(6,6,6,3,arrow = True,solid
rectangle(4,0,8,9);
for (i < 3){
if (i > 0){
line(-5*i + 15,7,-5*i + 12,7,arr
}circle(-5*i + 11,7)
}
```



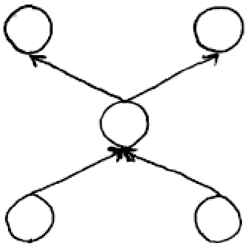


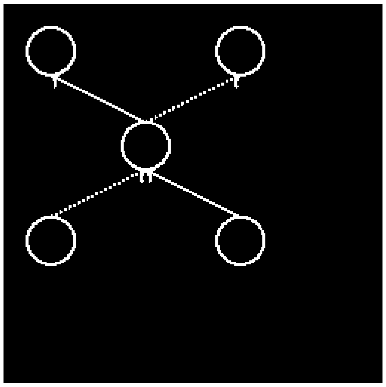
```
reflect(reflect(y = 5)){
reflect(reflect(x = 5)){
line(0,0,2,0,arrow = False,solid
line(0,0,0,2,arrow = False,solid
}
}
```



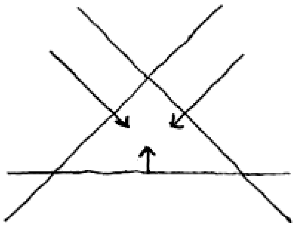


```
reflect(reflect(x = 14)){
for (i < 3){
circle(5,-4*i + 9);
line(10,-4*i + 9,12,-4*i + 9,arr
rectangle(12,-4*i + 8,14,-4*i +
}
}
```



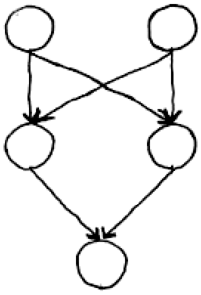


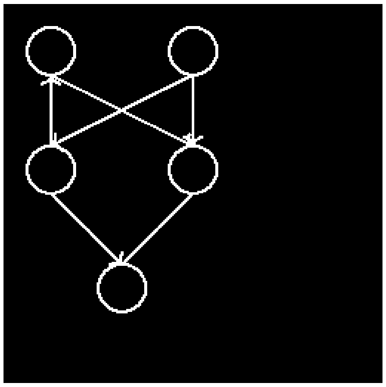
```
reflect(reflect(x = 10)){
for (i < 3){
if (i > 0){
line(-4*i + 13,4*i + -2,-4*i + 9
}circle(-4*i + 9,4*i + 1)
}
}
```



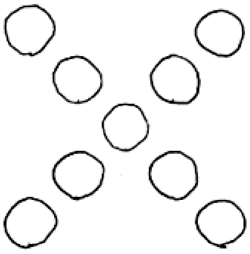
Sampled no finished traces.

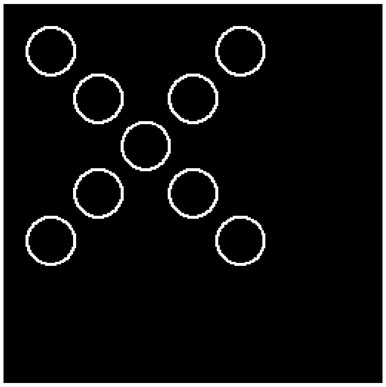
```
line(0,2,12,2,arrow = False,solid = T
line(6,2,6,3,arrow = True,solid = True
reflect(reflect(x = 12)){
line(0,0,9,9,arrow = False,solid = Tru
line(10,7,7,4,arrow = True,solid = Tru
}
```



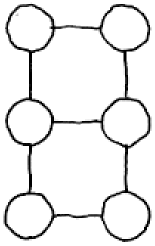


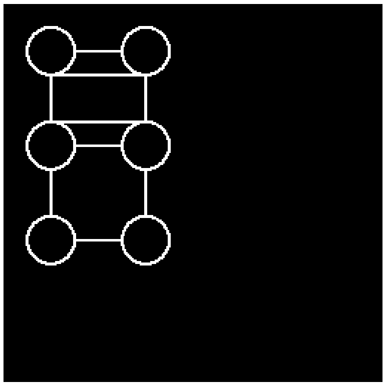
```
circle(4,1);
reflect(reflect(x = 8)){
for (i < 3){
if (i > 0){
circle(1,-5*i + 16);
line(1,10,6*i + -5,7,arrow = Tru
}line(7,5,4,2,arrow = True,solid
}
}
```



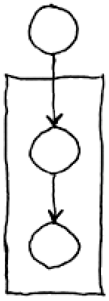


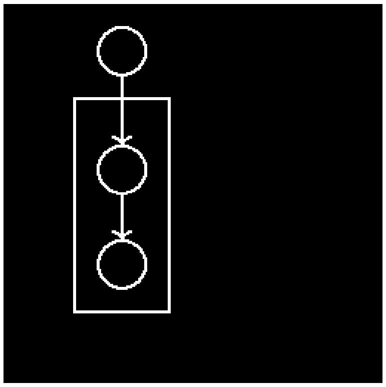
```
reflect(reflect(x = 10)){
circle(9,1);
for (i < 4){
circle(-2*i + 7,2*i + 3)
}
}
```





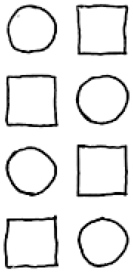
```
reflect(reflect(x = 6)){
for (i < 3){
if (i > 0){
line(1,-4*i + 10,1,-4*i + 12,arr
}circle(1,-4*i + 9);
line(2,-4*i + 9,4,-4*i + 9,arrow
}
}
```

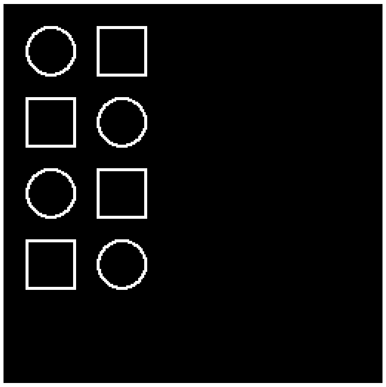




```
rectangle(0,0,4,9);
for (i < 3){
if (i > 0){
circle(2,-4*i + 10);
line(2,-5*i + 15,2,-4*i + 11,arr
}circle(2,11)
}
```

238





```
for (i < 2){
circle(4,6*i + 1);
circle(1,6*i + 4);
rectangle(3,6*i + 3,5,6*i + 5);
rectangle(0,6*i,2,6*i + 2)
}
```