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

$$\mathbb{P}[T_1] \propto W_{T_1} f \quad (1)$$

where W_{T_1} is a learned weight matrix.

Subsequent tokens are predicted as:

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

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.

28 2 Simulating hand drawings

29 We introduce noise into the \LaTeX rendering process by:

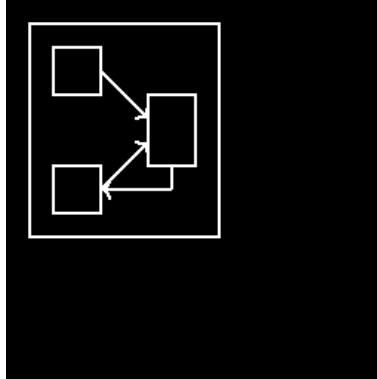
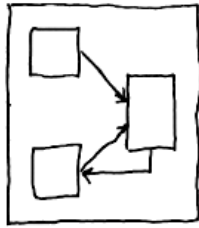
- 30 • Rescaling the image intensity by a factor chosen uniformly at random from $[0.5, 1.5]$
- 31 • Translating the image by ± 3 pixels chosen uniformly random
- 32 • Rendering the \LaTeX using the `pencildraw` style, which adds random perturbations to the
33 paths drawn by \LaTeX in a way designed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

35 3 Generating synthetic training data

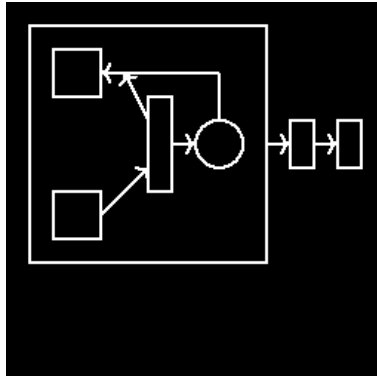
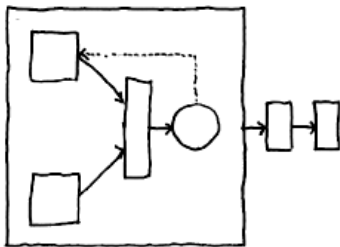
36 I need to describe the procedure that I used to sample the scenes. it's pretty involved

37 4 Full results on drawings data set

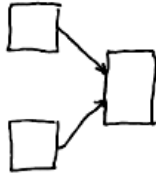
38 Below we show our full data set of drawings. The leftmost column is a hand drawing. The middle
39 column is a rendering of the most likely trace discovered by the neurally guided SMC sampling
40 scheme. The rightmost column is the program we synthesized from a ground truth execution trace of
41 the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



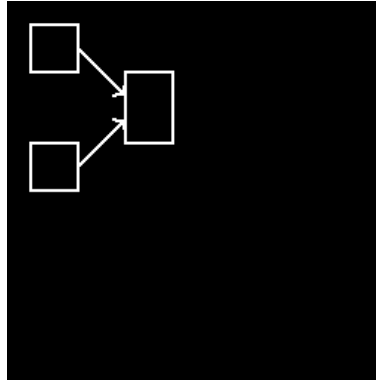
```
line(6,2,3,2,arrow = True,solid
line(6,2,6,3,arrow = False,solid
rectangle(0,0,8,9);
rectangle(5,3,7,6);
reflect(reflect(y = 9)){
line(3,2,5,4,arrow = True,solid
rectangle(1,6,3,8)
}
```



Solver timeout

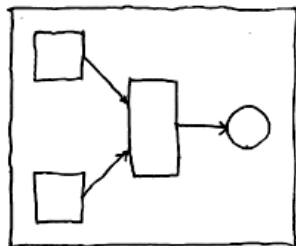


46

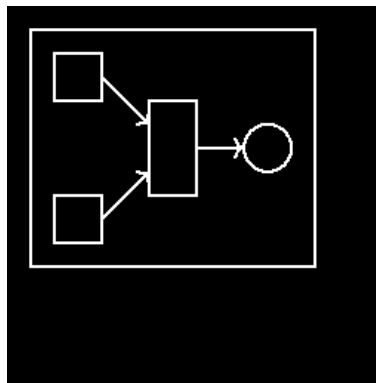


```
rectangle(4,2,6,5);
reflect(reflect(y = 7)){
line(2,6,4,4,arrow = True,solid
rectangle(0,5,2,7)
}
```

47

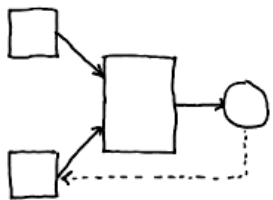


48

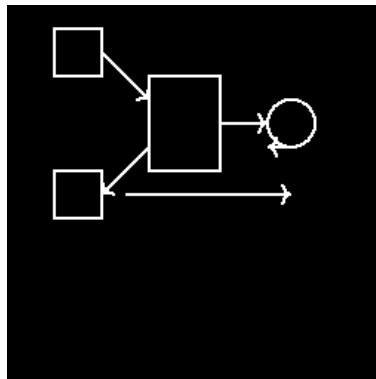


```
circle(10,5);
line(7,5,9,5,arrow = True,solid
rectangle(0,0,12,10);
reflect(reflect(y = 10)){
line(3,8,5,6,arrow = True,solid
rectangle(1,1,3,3);
rectangle(5,3,7,7)
}
```

49



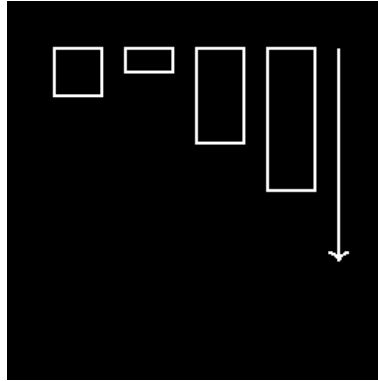
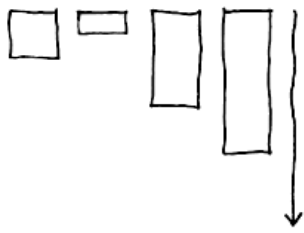
50



```
circle(10,4);
line(10,1,10,3,arrow = False,solid
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)
}
```

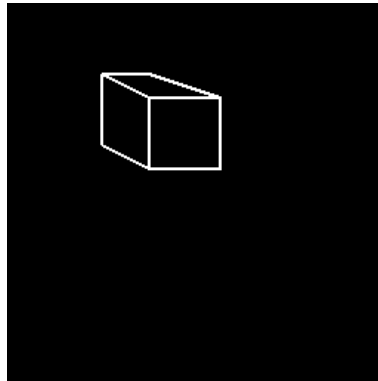
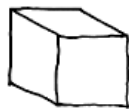
51

52



Solver timeout

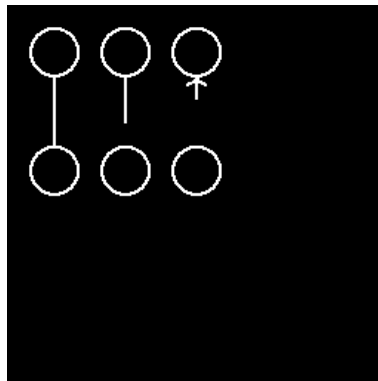
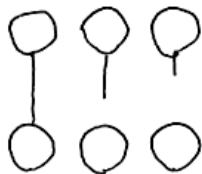
53



54

```
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
}
```

55

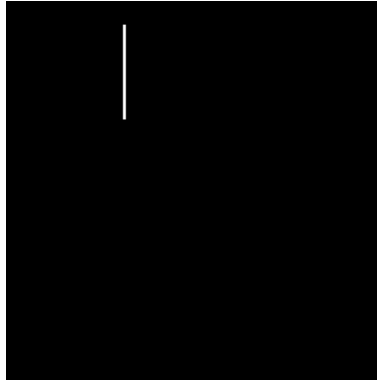


56

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

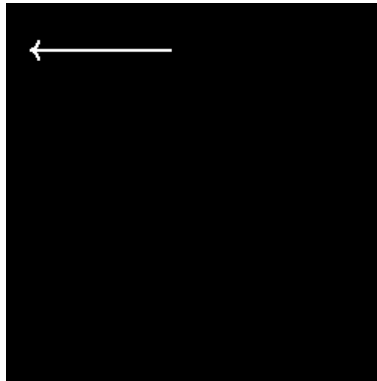
57

58



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

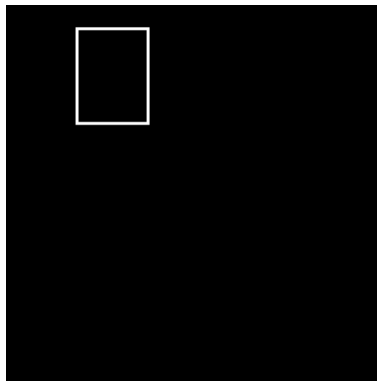
59



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

60

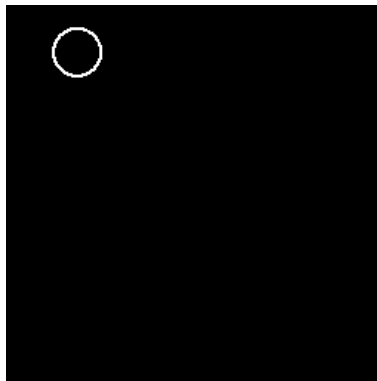
61



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

62

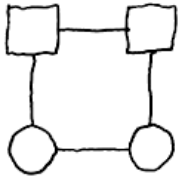
63



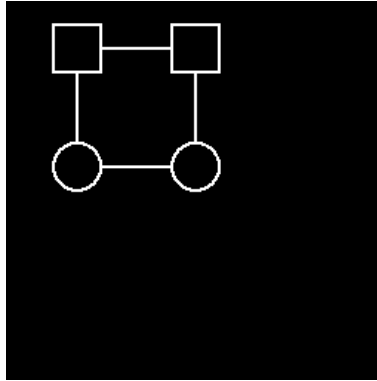
```
circle(1,1)
```

64

65

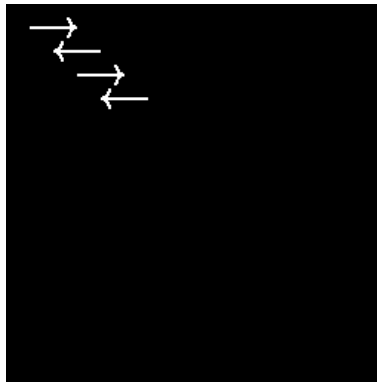
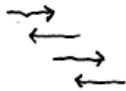


66



```
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)
}
```

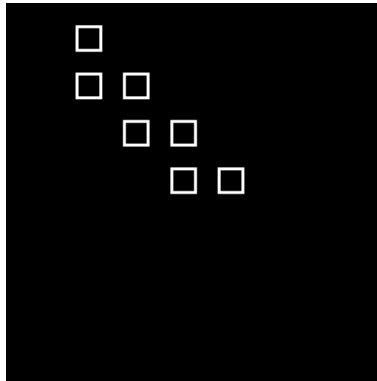
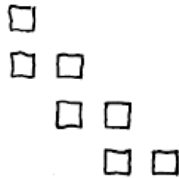
67



68

```
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
```

69



70

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

71

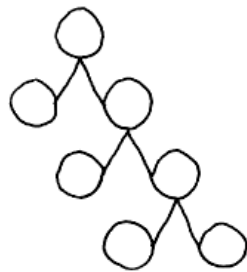


72

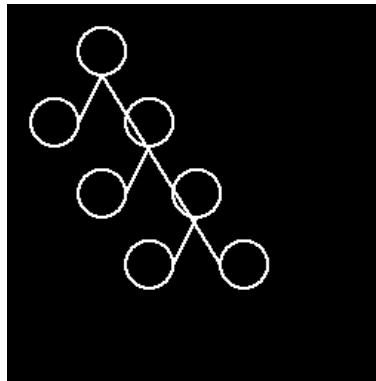


```
line(0,3,2,3,arrow = False,solid
line(2,1,4,1,arrow = False,solid
line(1,2,3,2,arrow = False,solid
line(3,0,5,0,arrow = False,solid
```

73

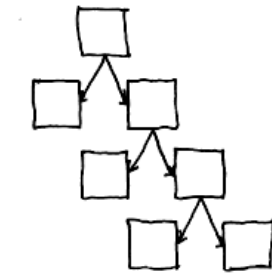


74

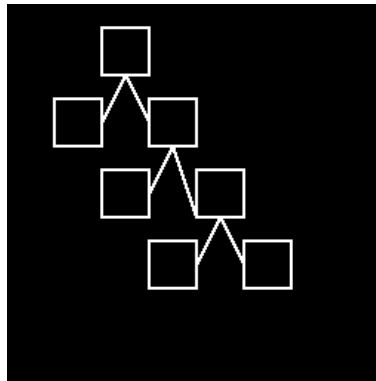


Solver timeout

75



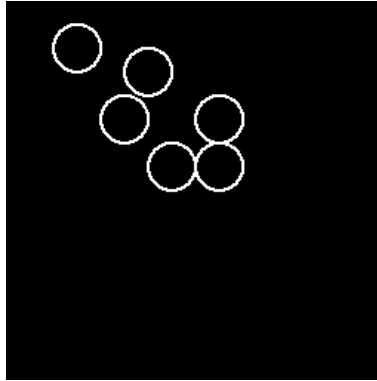
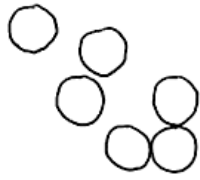
76



```
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
}
```

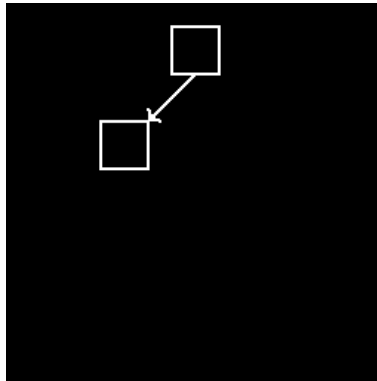
77

78



Solver timeout

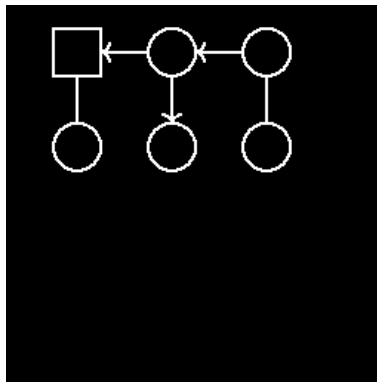
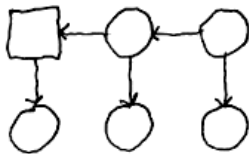
79



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

80

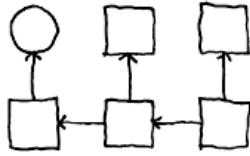
81



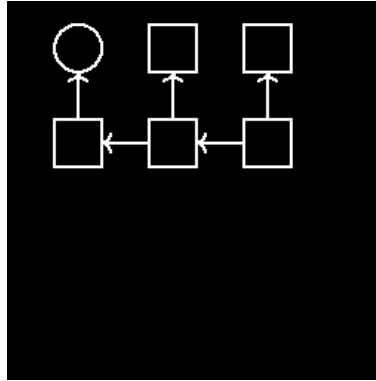
Solver timeout

82

83

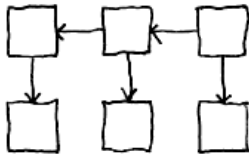


84

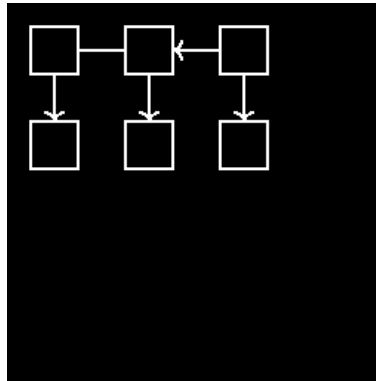


```
circle(1,5);
for (i < 3){
  if (i > 0){
    line(-4*i + 12,1,-4*i + 10,1,arrow = Tr
    rectangle(-4*i + 12,4,-4*i + 14,
  }line(-4*i + 9,2,-4*i + 9,4,arrow = Tr
  rectangle(-4*i + 8,0,-4*i + 10,2)
}
```

85

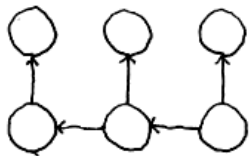


86

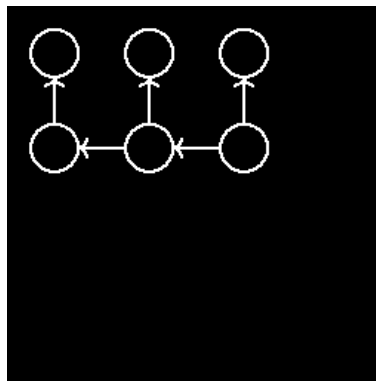


```
for (i < 3){
  if (i > 0){
    line(4*i,5,4*i + -2,5,arrow = Tr
  }line(4*i + 1,4,4*i + 1,2,arrow = Tr
  rectangle(4*i,0,4*i + 2,2);
  rectangle(4*i,4,4*i + 2,6)
}
```

87

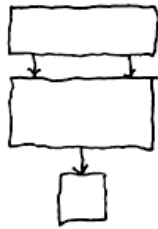


88

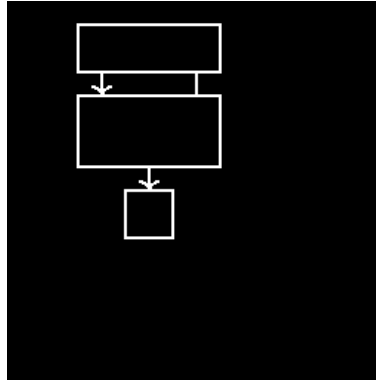


```
for (i < 3){
  line(-4*i + 9,2,-4*i + 9,4,arrow = Tr
  for (j < 2){
    circle(-4*i + 9,4*j + 1);
    line(-4*j + 8,1,-4*j + 6,1,arrow = Tr
  }
}
```

89



90

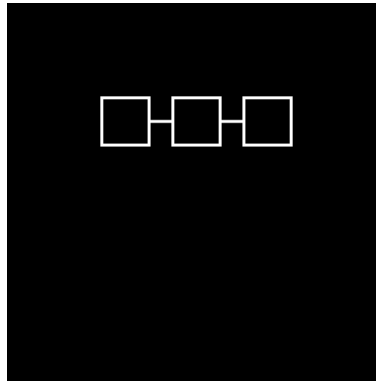


```
reflect(reflect(x = 6)){
  for (i < 3){
    if (i > 0){
      line(-2*i + 7,-4*i + 11,-2*i + 7
      rectangle(0,-4*i + 11,6,-3*i + 1
    }rectangle(2,0,4,2)
  }
}
```

91



92

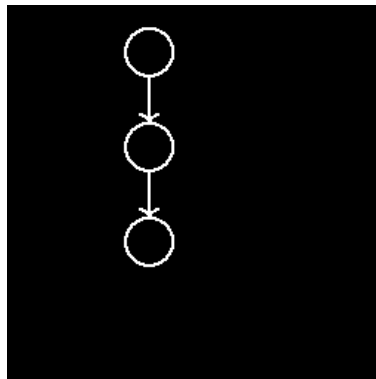


```
for (i < 3){
  if (i > 0){
    line(-3*i + 9,1,-3*i + 8,1,arrow
  }rectangle(-3*i + 6,0,-3*i + 8,2
}
```

93



94



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

95



96



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



97



98



99

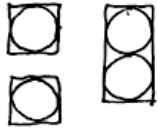


```
line(0,2,2,2,arrow = False,solid
line(0,0,0,2,arrow = False,solid
```

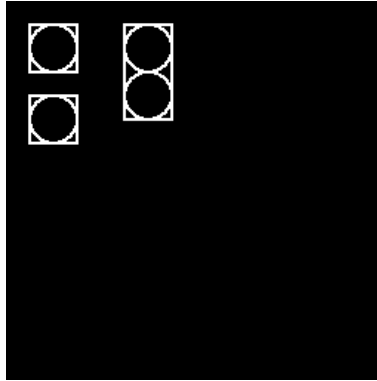
100

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

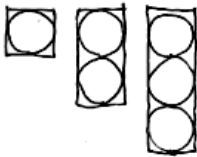
101



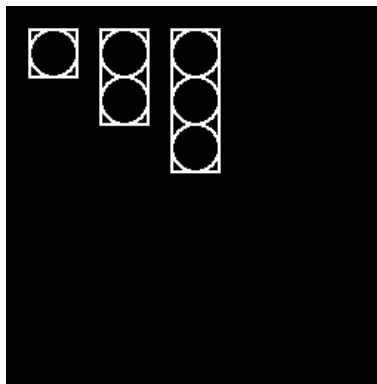
102



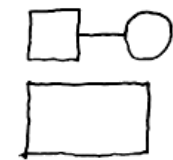
```
for (i < 3){
  if (i > 0){
    circle(5,2*i);
    circle(1,-3*i + 7);
    rectangle(0,-3*i + 6,2,-3*i + 8);
  }rectangle(4,1,6,5)
}
```



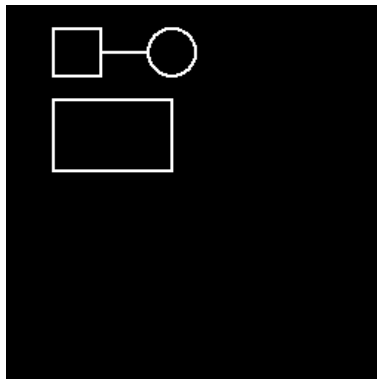
103



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



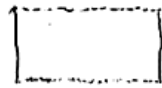
105



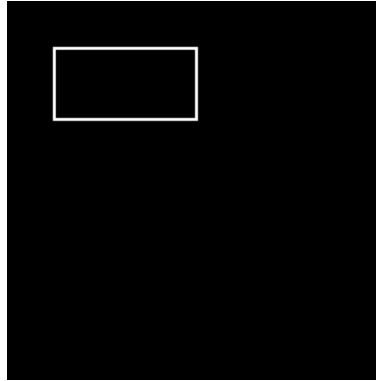
```
circle(5,5);
line(2,5,4,5,arrow = False,solid);
rectangle(0,0,5,3);
rectangle(0,4,2,6)
```

106

107



108

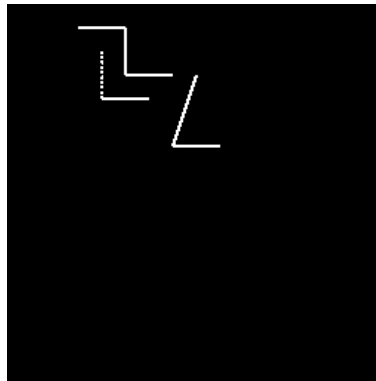


```
line(0,0,6,0,arrow = False,solid
line(0,3,6,3,arrow = False,solid
line(6,0,6,3,arrow = False,solid
line(0,0,0,3,arrow = False,solid
```

109

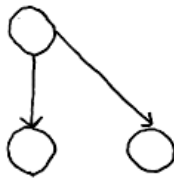


110

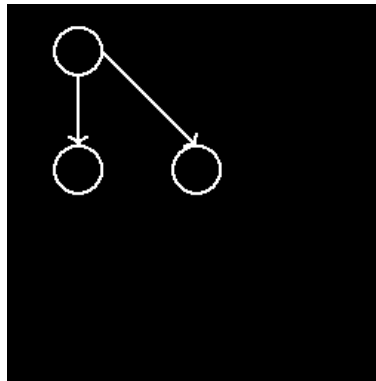


Solver timeout

111



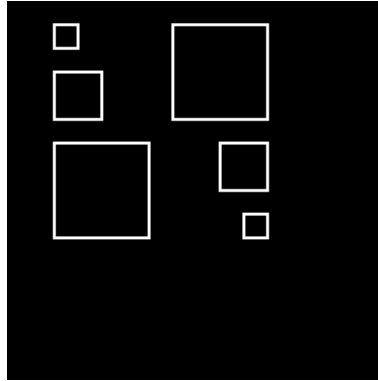
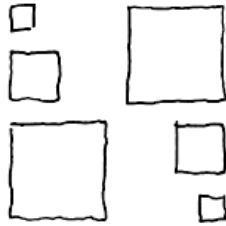
112



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

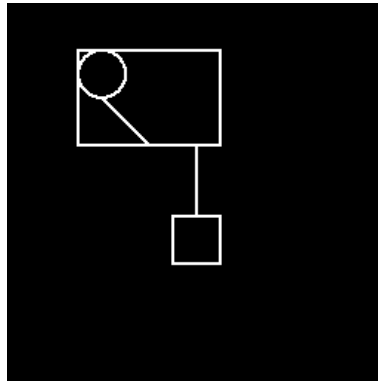
113

114



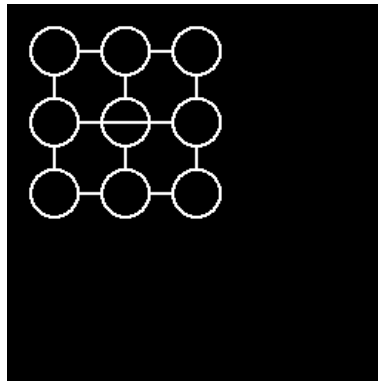
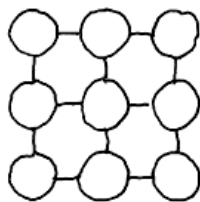
Solver timeout

115



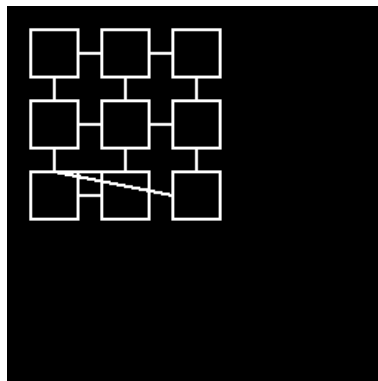
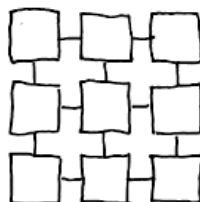
```
for (i < 3){
  if (i > 0){
    line(-4*i + 9, 5*i + -3, -2*i + 7,
    rectangle(-4*i + 8, 5*i + -5, 6, 7*
  }circle(1, 8)
}
```

117



Solver timeout

119



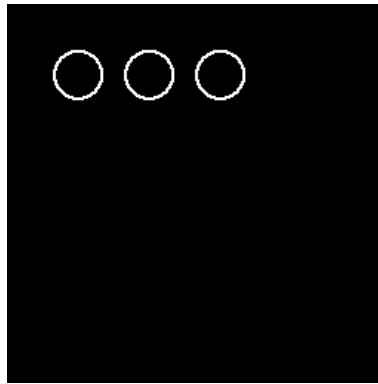
Solver timeout

120

121

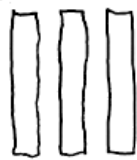


122

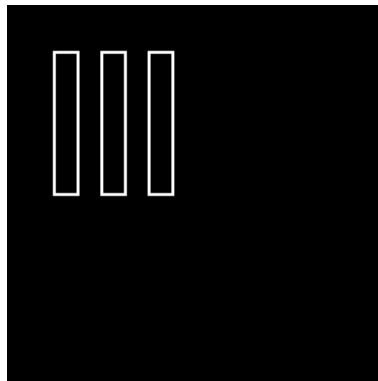


```
for (i < 3){
  circle(3*i + 1,1)
}
```

123



124

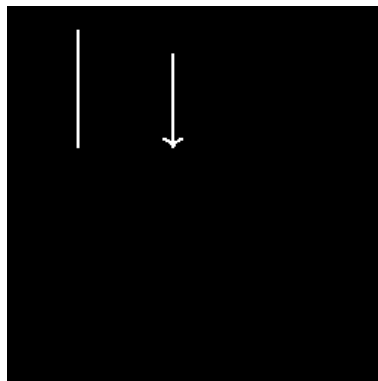


```
for (i < 3){
  rectangle(2*i,0,2*i + 1,6)
}
```

125



126

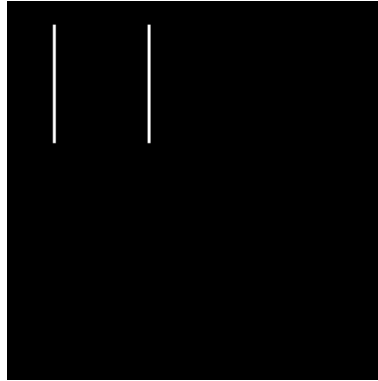


```
line(0,0,0,5,arrow = False,solid
line(4,1,4,5,arrow = False,solid
line(4,0,4,1,arrow = False,solid
```

127

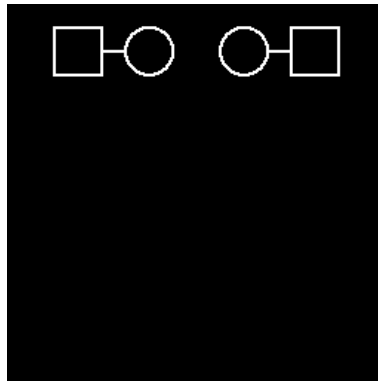


128



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

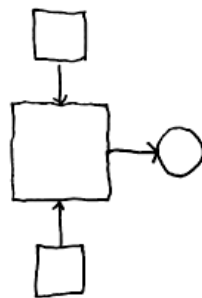
129



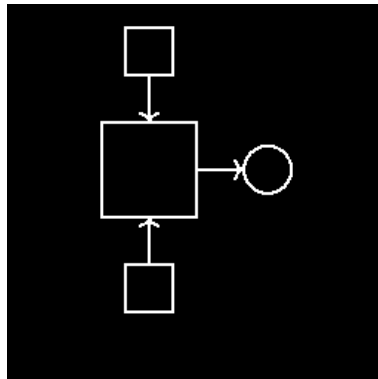
130

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

131



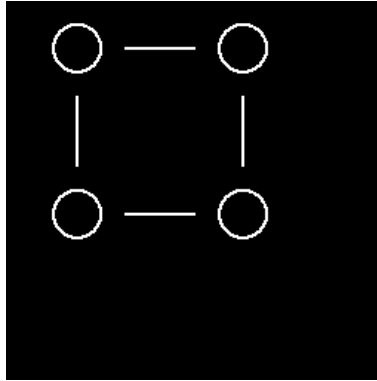
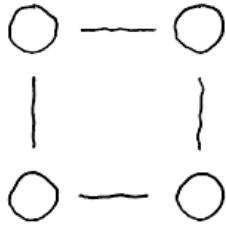
132



```
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)
}
```

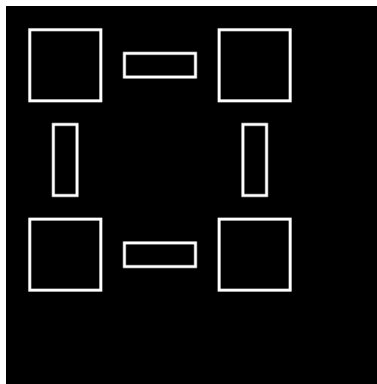
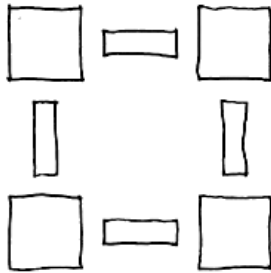
133

134



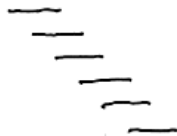
```
reflect(reflect(x = 9)){
  line(1,3,1,6,arrow = False,solid
  reflect(reflect(y = 9)){
    circle(1,8);
    line(3,1,6,1,arrow = False,solid
  }
}
```

135



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

137

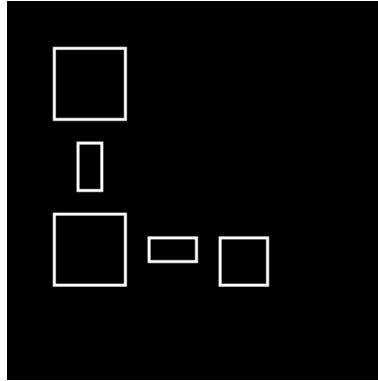
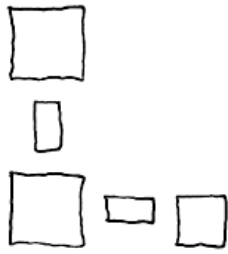


```
for (i < 3){
  line(-2*i + 5,2*i,-2*i + 7,2*i,a
  line(-2*i + 4,2*i + 1,-2*i + 6,2
}
```

138

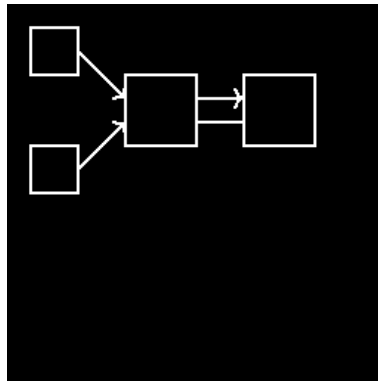
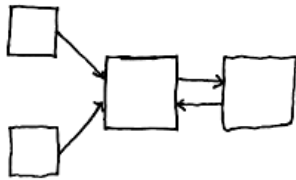
139

140



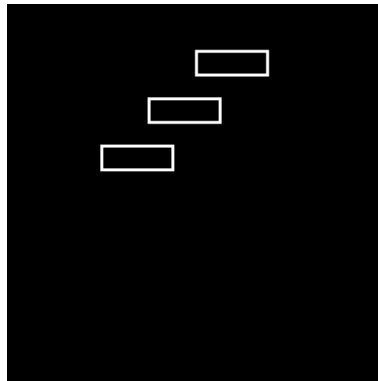
```
for (i < 3){
  if (i > 0){
    rectangle(-3*i + 10,i + -1,-3*i
    rectangle(0,-7*i + 14,3,-7*i + 1
  }rectangle(1,4,2,6)
}
```

141



Solver timeout

143

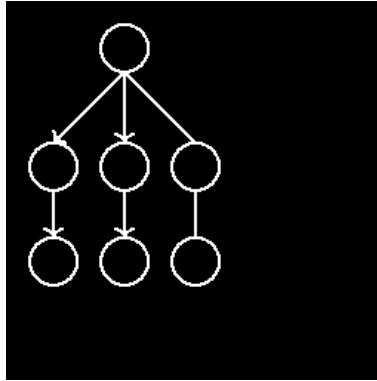
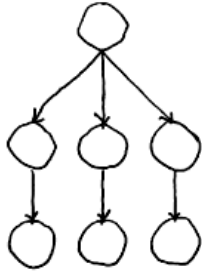


```
for (i < 3){
  rectangle(2*i,2*i,2*i + 3,2*i +
}
```

144

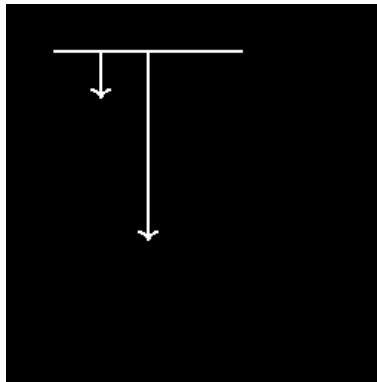
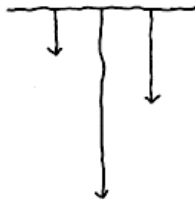
145

146



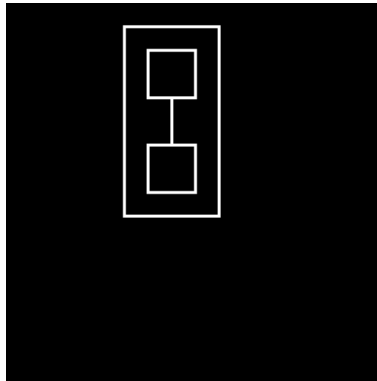
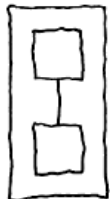
```
circle(4,10);
for (i < 3){
circle(3*i + 1,5);
circle(3*i + 1,1);
line(4,9,3*i + 1,6,arrow = True,
line(3*i + 1,4,3*i + 1,2,arrow =
}
```

147



```
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
```

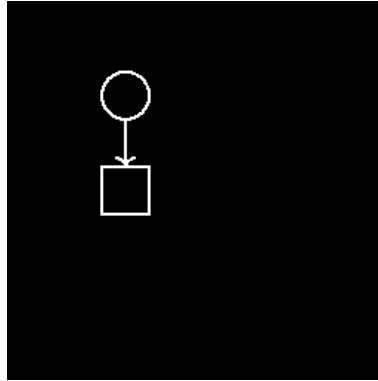
149



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

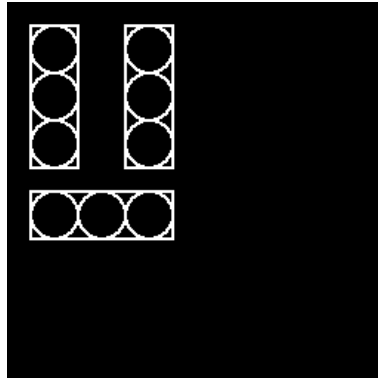
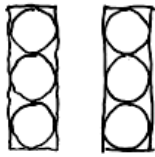
151

152



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

153

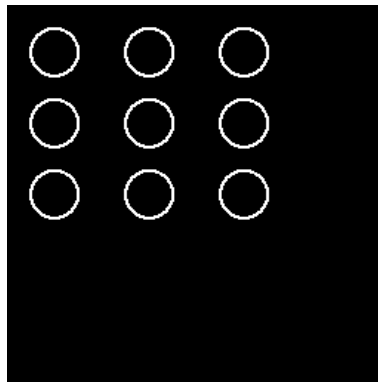
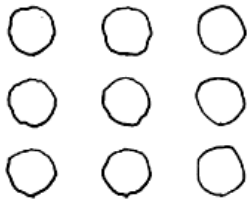


Solver timeout

154



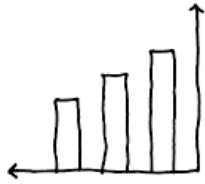
155



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

156

157

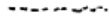


158

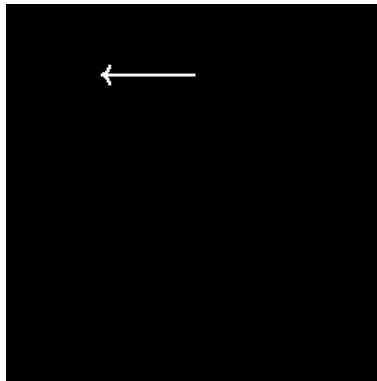


```
for (i < 3){
  if (i > 0){
    line(8,0,8*i + -8,7*i + -7,arrow
  }rectangle(-2*i + 6,0,-2*i + 7,-
}
```

159

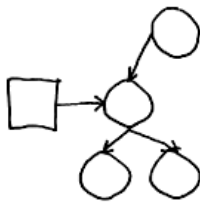


160

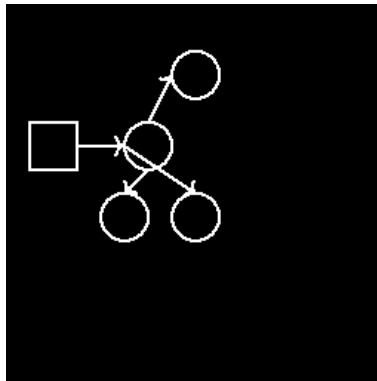


```
line(4,0,0,0,arrow = False,solid
```

161



162

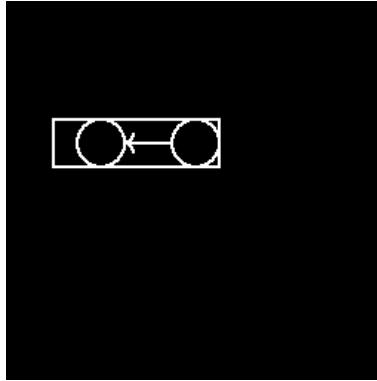


Solver timeout

163



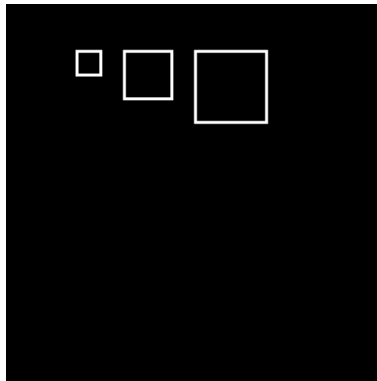
164



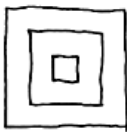
```
circle(2,1);
circle(6,1);
line(5,1,3,1,arrow = True,solid
rectangle(0,0,7,2)
```



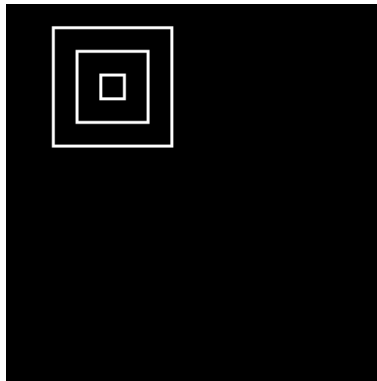
165



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



166

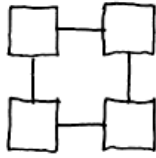


```
for (i < 3){
rectangle(i,i,-1*i + 5,-1*i + 5)
}
```

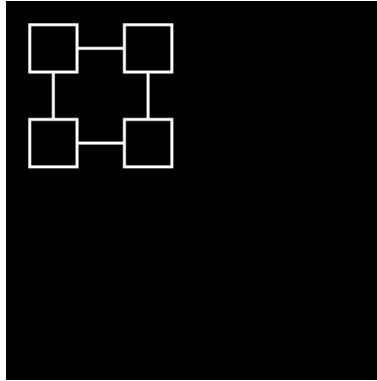
167

168

169

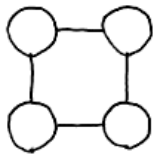


170

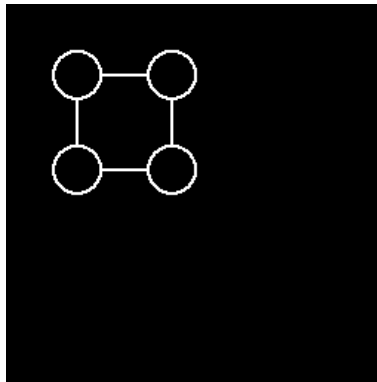


```
reflect(reflect(x = 6)){
  line(5,2,5,4,arrow = False,solid
  reflect(reflect(y = 6)){
    line(2,1,4,1,arrow = False,solid
    rectangle(4,4,6,6)
  }
}
```

171

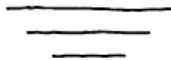


172

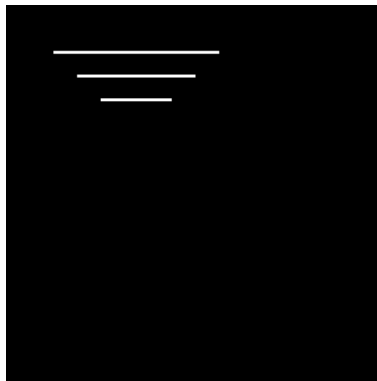


```
reflect(reflect(x = 6)){
  line(1,2,1,4,arrow = False,solid
  reflect(reflect(y = 6)){
    circle(5,1);
    line(2,1,4,1,arrow = False,solid
  }
}
```

173

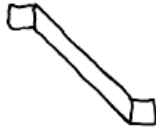


174

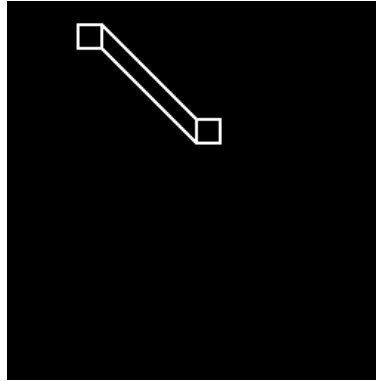


```
for (i < 3){
  line(i,-1*i + 2,-1*i + 7,-1*i +
}
```

175

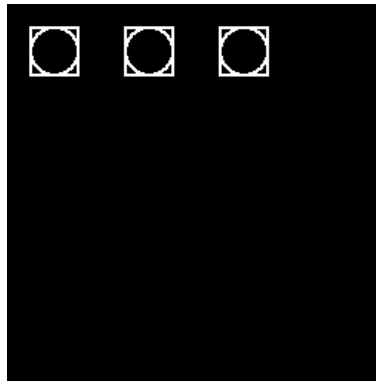


176



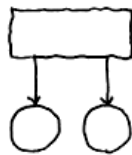
```
line(1,4,5,0,arrow = False,solid
line(1,5,5,1,arrow = False,solid
rectangle(0,4,1,5);
rectangle(5,0,6,1)
```

177

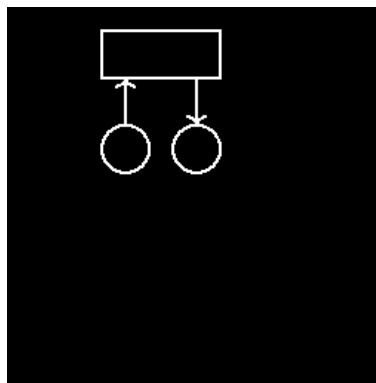


```
for (i < 3){
circle(4*i + 1,1);
rectangle(4*i,0,4*i + 2,2)
}
```

178



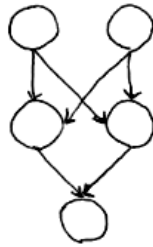
179



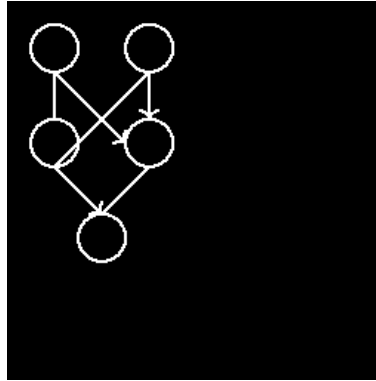
```
rectangle(0,4,5,6);
reflect(reflect(x = 5)){
circle(1,1);
line(1,4,1,2,arrow = True,solid
}
```

180

181

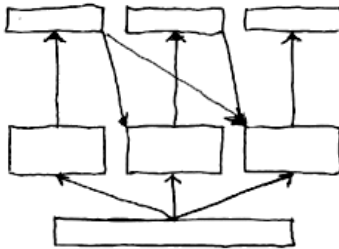


182



```
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 + 12,arrow = True,solid)
}
}
}
```

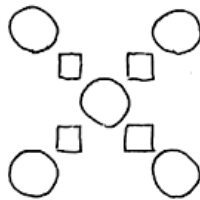
183



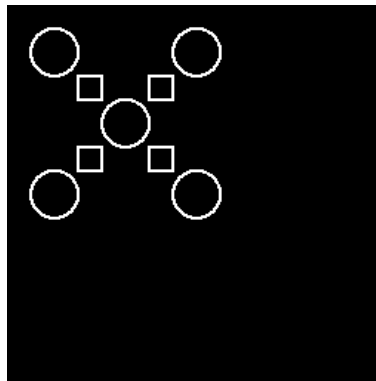
184

Sampled no finished traces. Solver timeout

185



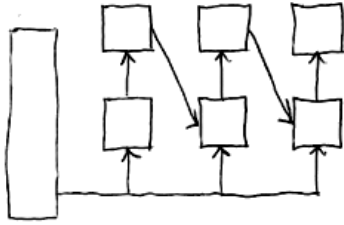
186



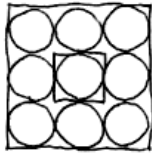
```
reflect(reflect(y = 8)){
for (i < 3){
if (i > 0){
rectangle(3*i + -1,-3*i + 8,3*i + 1,-3*i + 8);
circle(3*i + 1,-3*i + 7)
}
}
}
```

187

188

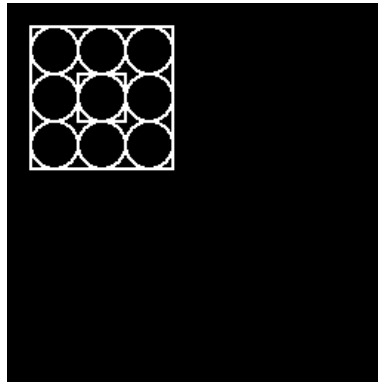


189



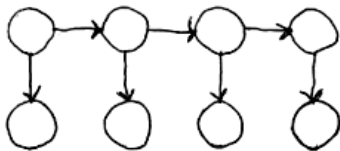
190

Sampled no finished traces. Solver timeout

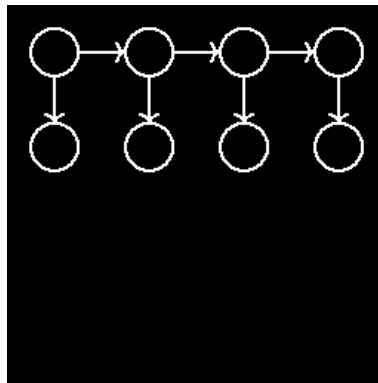


```
for (i < 3){
  if (i > 0){
    rectangle(2*i + -2,2*i + -2,-2*i
  }for (j < 3){
    circle(2*j + 1,2*i + 1)
  }
}
```

191



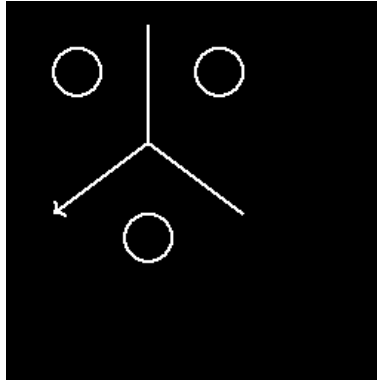
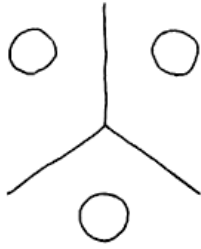
192



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

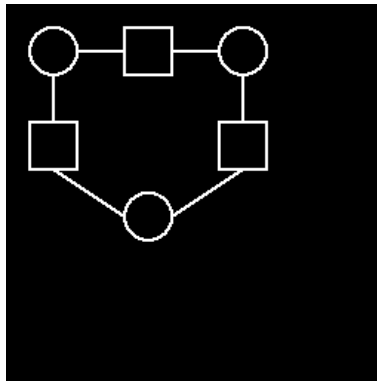
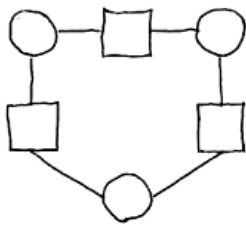
193

194



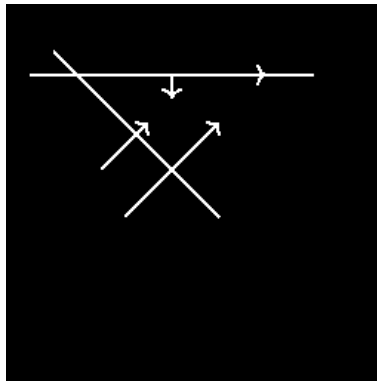
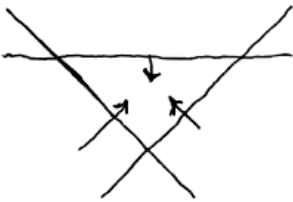
```
line(4,5,4,10,arrow = False,solid
reflect(reflect(x = 8)){
circle(4,1);
circle(1,8);
line(4,5,8,2,arrow = False,solid
}
```

195



Solver timeout

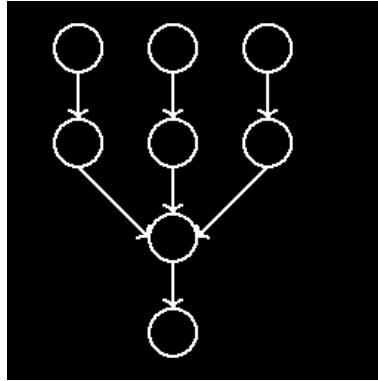
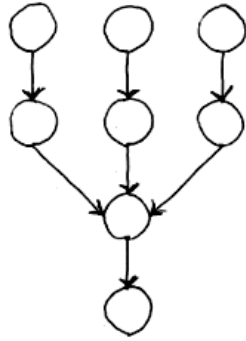
197



```
line(0,6,12,6,arrow = False,solid
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,solid
}
```

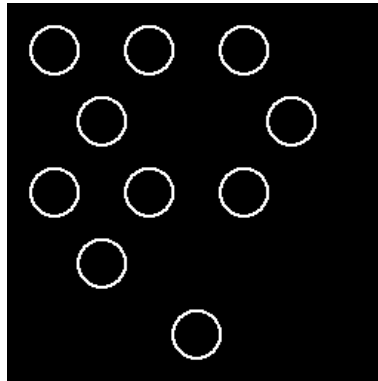
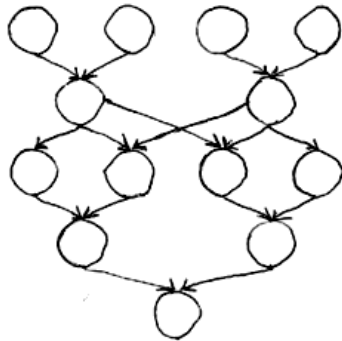
199

200



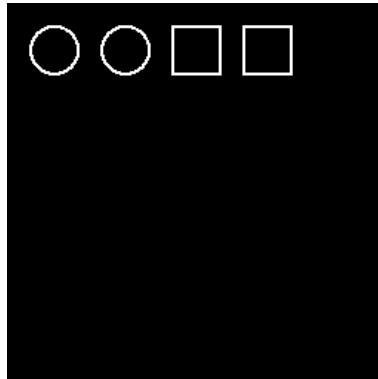
Solver timeout

201



Solver timeout

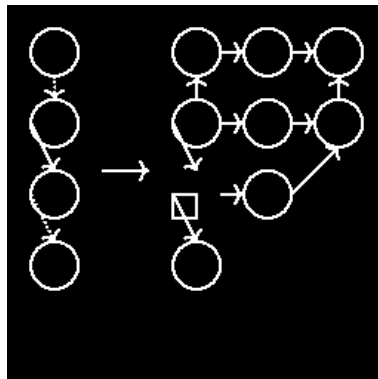
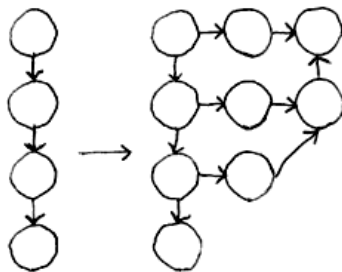
203



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

204

205

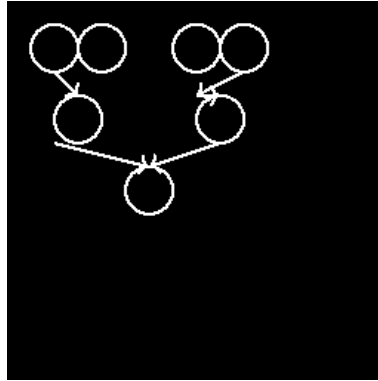
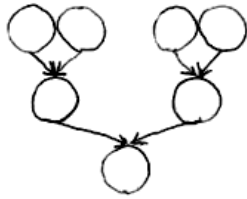


Solver timeout

206

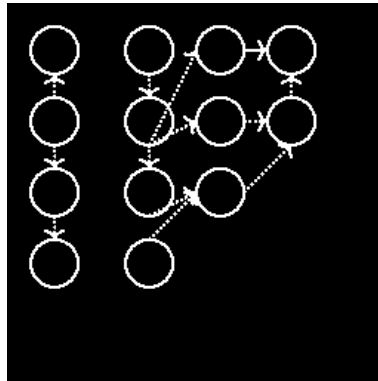
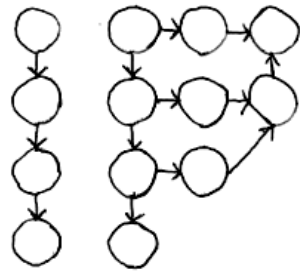
207

208



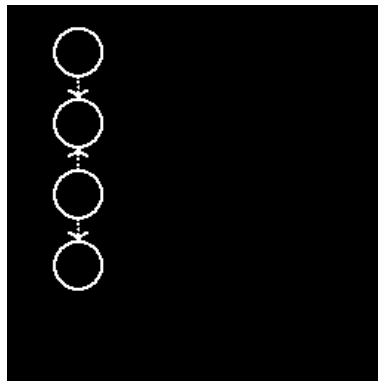
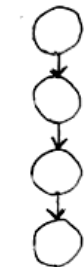
Solver timeout

209



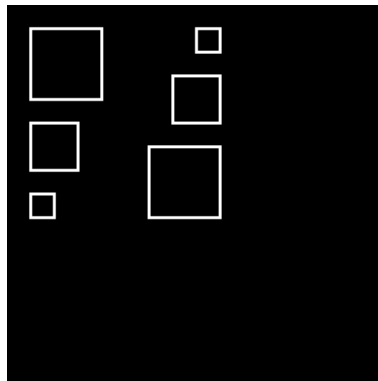
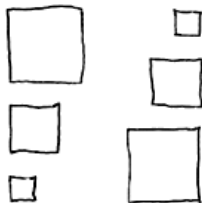
Solver timeout

211



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

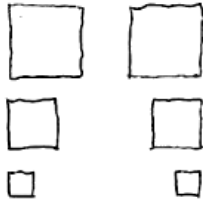
213



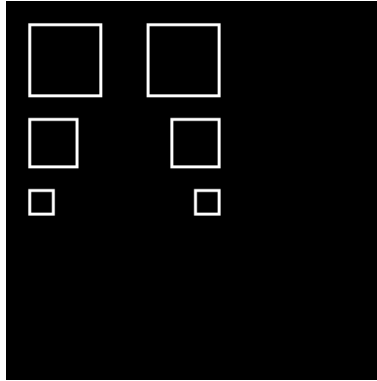
Solver timeout

214

215

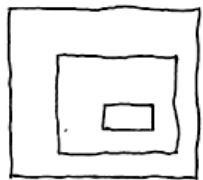


216

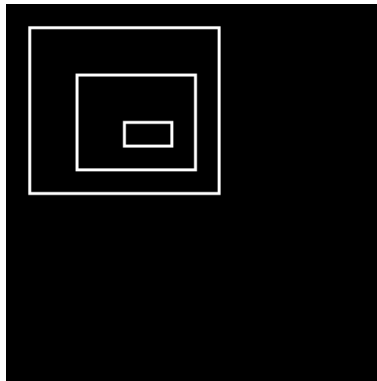


```
reflect(reflect(x = 8)){
rectangle(0,0,1,1);
rectangle(0,5,3,8);
rectangle(0,2,2,4)
}
```

217

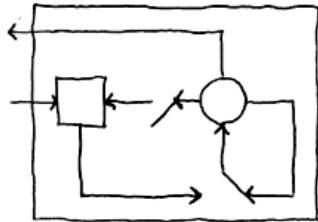


218

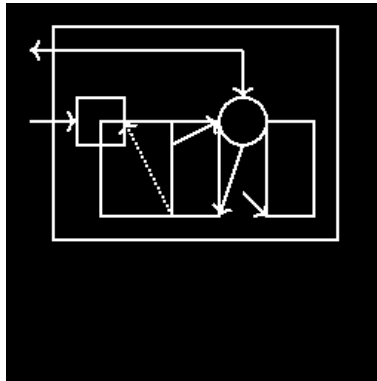


```
for (i < 3){
rectangle(2*i,i,-1*i + 8,-2*i + 8)
}
```

219

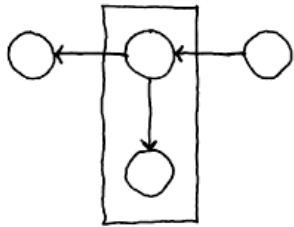


220

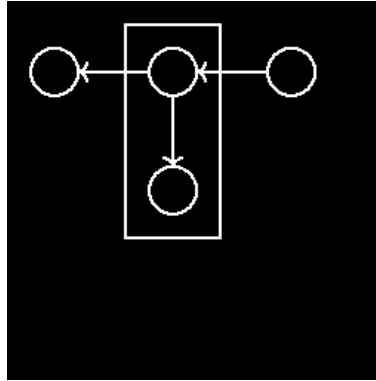


Solver timeout

221

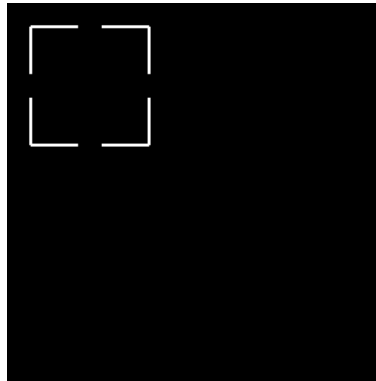
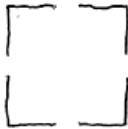


222



```
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,ar
}circle(-5*i + 11,7)
}
```

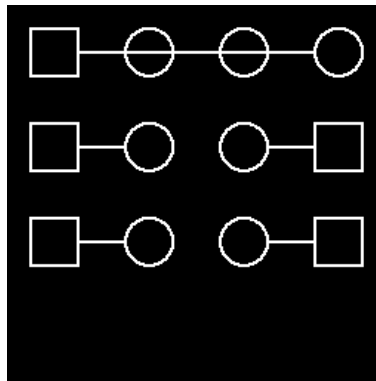
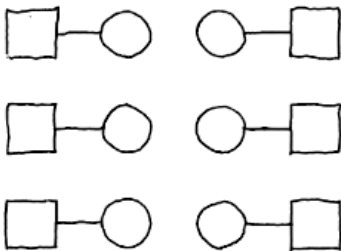
223



224

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

225

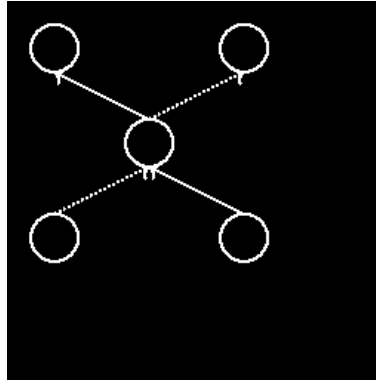
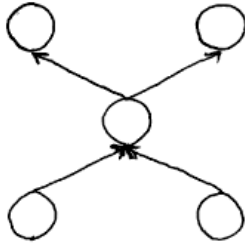


226

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

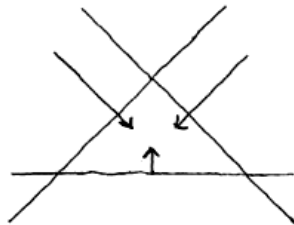
227

228



```
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)
  }
}
```

229

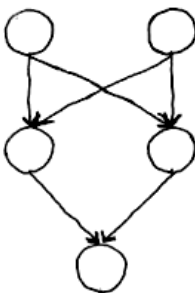


230

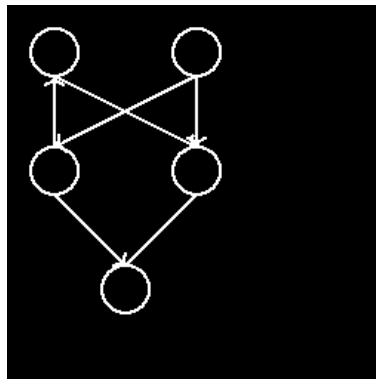
Sampled no finished traces.

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

231



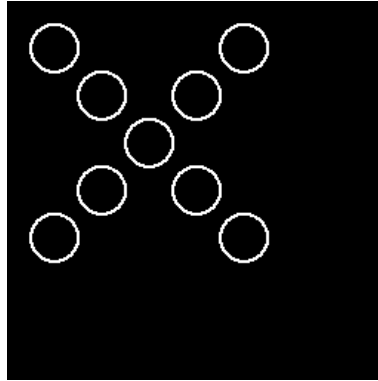
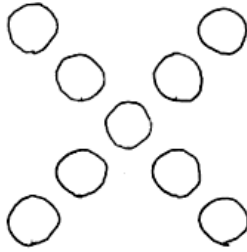
232



```
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 = True,solid = True)
    }line(7,5,4,2,arrow = True,solid = True)
  }
}
```

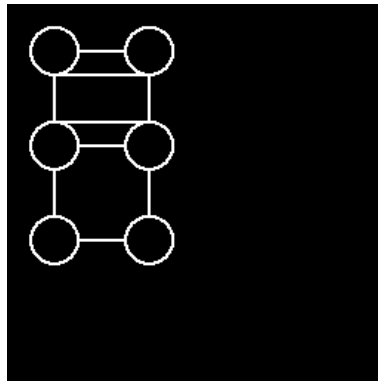
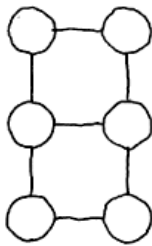
233

234



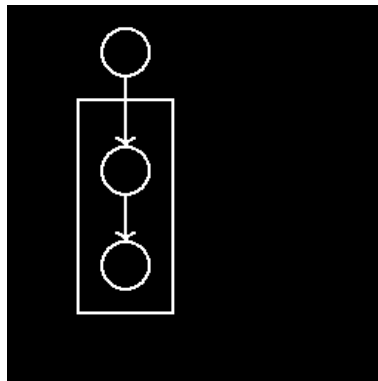
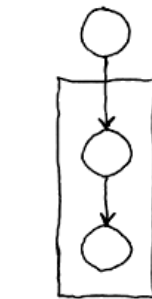
```
reflect(reflect(x = 10)){
  circle(9,1);
  for (i < 4){
    circle(-2*i + 7,2*i + 3)
  }
}
```

235



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

237

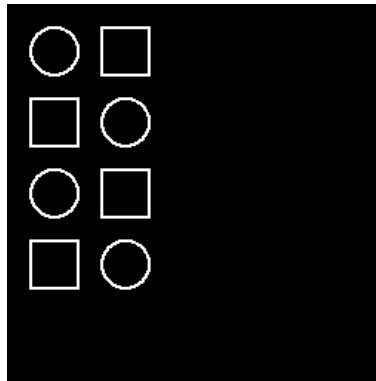
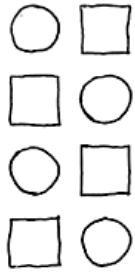


```
rectangle(0,0,4,9);
for (i < 3){
  if (i > 0){
    circle(2,-4*i + 10);
    line(2,-5*i + 15,2,-4*i + 11,arrow)
  }circle(2,11)
}
```

238

239

240



```
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)  
}
```