vordenker
Summer-Edition 2017

— vordenker-archive —

Rudolf Kaehr

(1942-2016)

Title

Extending Architectures of Cellular Automata

A wee sketch on the complementarity of directed morphoCAs

Archive-Number / Categories

3_39 / K12, K09, K11

Publication Date

2014

Keywords / Topics

Morphograms, Celluilar Automata, Semiotics

Disciplines

Computer Science, Artificial Intelligence and Robotics, Logic and Foundations of Mathematics, Cybernetics, Theory of Science

Abstract

A short introduction to strategies of complexity reductions, using shape grammar to derive cellular automata rule patterns.

Chain of abstractions

CA -> Shape Grammar(SG) -> Morphogrammatics

SG -> CA, MG -> CA, morphoCA

The combination of shape grammar (SG) for managing the input and CA for managing the output brings together the human intuitive approach (visualization of the abstract) with a computational system that can generate large design solution spaces in a tractable manner.

Shape grammars can be used analytically, as in reverse engineering, for characterizing and classifying designs and patterns of designs, referred to as styles in architecture. A shape grammar includes a vocabulary of shapes and a set of spatial relations to control the positioning of shapes in the vocabulary.

It is interesting to note that merely increasing r from 1 to 2 and maintaining the colors at two increases the rule space from 256 to 4.3 billion.

Citation Information / How to cite

Rudolf Kaehr: "Extending Architectures of Cellular Automata", www.vordenker.de (Sommer Edition, 2017) J. Paul (Ed.), http://www.vordenker.de/rk/rk_Extending-Architectures-of-Cellular-Automata_2014.pdf

Categories of the RK-Archive

Extending Architectures of MorphoCellular Automata

A wee sketch on the complementarity of directed morphoCAs

Dr. phil Rudolf Kaehr

ISSN 2041-4358
(work in progress, vers. 0.1, November 2014)

Gotthard Gunther (1900 - 1984)

After 50 Years. What happend to Morphogrammatics?

Introducing directed morphoCAs

Motivations

What are cellular automata?

What is the meaning of morphoCAs?

Introduction

New morphoCA Rules

RulesSchemes CA$^{(5,4,5)}$

Junctional morphoRules CA$^{(5,2,2)}$

Junctional morphoRules CA$^{(5,2,5)}$

Transjunctional morphoRules CA$^{(5,3,5)}$

Transjunctional morphoRules CA$^{(5,4,5)}$

Morphogram tables

Junctional morphograms MG$^{(5,2)}$

Morphograms MG$^{(5,5)}$

Introducing directed morphoCAs

Motivations

Strategies of complexity reductions

Chain of abstractions
CA -> Shape Grammar -> Morphogrammatics
SG -> CA, MG -> CA, morphoCA

Using Shape Grammar to Derive Cellular Automata Rule Patterns (Complex Systems, 17 (2007) 79–102**)**
Thomas H.Speller, Jr.a, Daniel Whitney, Edward Crawley

Frustration
“This paper shows how shape grammar can be used to derive cellular automata (CA) rules. Searching the potentially *astronomical* space of CA rules for relevance to a particular context has *frustrated* the wider application of CA as powerful computing systems.

“An approach is offered using *shape grammar* to visually depict the desired conditional rules of a behavior or system architecture (a form - function) under investigation, followed by a transcription of these rules as patterns into CA.

Visualization of the abstract
The combination of shape grammar for managing the input and CA for managing the output brings together the human intuitive approach (*visualization of the abstract*) with a computational system that can generate large design solution spaces in a tractable manner.

“The formula for calculating the *rule size space* in a one - dimensional system is $k^{k^{(2r+1)}}$, where k represents the color possibilities for each state and *r* is the range or radius of the neighborhood.
It is interesting to note that merely increasing *r* from 1 to 2 and maintaining the colors at two increases the rule space from 256 to 4.3 billion.

Shape grammar
“Shape grammar is a formal generative approach that has been applied to creating architectural forms.
Shape grammar is a precise and at the same time intuitive methodology in the visual medium for generating languages of design.
Shape grammars can be used analytically, as in reverse engineering, for characterizing and classifying designs and patterns of designs, referred to as styles in architecture. A shape grammar includes a vocabulary of shapes and a

set of spatial relations to control the positioning of shapes in the vocabulary.

“Shape grammars are a geometrical design adaptation of Noam Chomsky’s formal (phrase structure or transformational) grammars and are *recursively* enumerable, having the capability of producing unrestricted languages. Thus, shape grammars are systems of rules for characterizing the composition of designs in *spatial* languages.”

Motivations

What are cellular automata?

Cellular automata are well known, mathematically elaborated and documented.
Main features are described by Rudy Rucker and John Walker in *Exploring Cellular Automata* as:

(1) **Parallelism** means that the individual cell updates are performed independently of each other. That is, we think of all of the updates being done at once.
(2) **Locality** means that when a cell is updated, its new color value is based solely on the old color values of the cell and of its nearest neighbors.
(3) **Homogeneity** means that each cell is updated according to the same rules.

Typically the color values of the cell and of its nearest eight neighbors are combined according to some logico-algebraic formula, or are used to locate an entry in a preset lookup table.

Cellular automata can act as good models for physical, biological and sociological phenomena. The reason for this is that each person, or cell, or small region of space "updates" itself independently (*parallelism*), basing its new state on the appearance of its immediate surroundings (*locality*) and on some generally shared laws of change (*homogeneity*).

https://www.fourmilab.ch/cellab/manual/

After Alex Wuensche, CA rules are defining the *geno*-type of dynamic systems, while the graphic and sonic patterns of the dynamics are showing the *pheno*-structure of the dynamic system.

This distinction is of importance and helps to understand the epistemological status of morphoCAs as ‘second-order’ constructs: the geno-type of (the geno-type of the pheno-typ) of CAs. Morphograms are the inscription of the ‘geno-type’ of the rules of CAs.

Mathematically, CAs are quit simple constructs:

A CA is a special automaton defined as a quadruple:
CA = (Γ, Q, N, δ) with
Γ: interconnection graph,
Q: set of states,
N: neighborhood definition,
δ: local dynamics (rules).
$\Gamma(x)_i = \delta(x_{i-1}, x_i, x_{i+1}) = x_i'$
(Γ,Q) : cell space,
configuration: C
global transitions: Γ: C - C

Combinatorics

For 1D CA: neighborhoods are $2^3 = 8$,

rule space : $2^8 = 256$.

For 2D CA: neighborhood $2^9 = 512$,

rule space: 2^{512} = 2^512

2^512

```
13 407 807 929 942 597 099 574 024 998 205 846 127 479 365 820 592 393 377 723 561 443 721 764 ∴
  030 073 546 976 801 874 298 166 903 427 690 031 858 186 486 050 853 753 882 811 946 569 946 ∴
  433 649 006 084 096
```

What is the meaning of morphoCAs?

There are some implications and presumptions in the classical definition of CAs that are not been explicitly mentioned.

a) morphoCAs are based on morphograms, and morphograms are defined *retro-recursively* as pre-semiotic *patterns* (morphé) and simultaneously as *dynamic* rules,
b) morphoCAs are topologically *oriented,*

c) morphoCAs are complex and *heterogenic* structurations (systems),
d) morphoCAs have a *heterarchical* form of organization.

The orientedness of classical CAs is not mentioned in the theory of CA because it is single and therefore trivial.

Retro-recursivity, i.e.*memristivity* is not known for classical CAs. Recursivity is reduced to memory-free and context-independent succession of the calculation, i.e. the applications of the elementary CA rules. Memory in classical CAs is a supplementary and therefore a secundary construct.

Parallelism is extended and replaced by *interactivity, homogeneity by heterogeneity (*discontexturality) and locallity by *polycontexturality*.

Mimickry

1) straight-directed: $\Gamma(x)_i = \delta(x_{i-1}, x_i, x_{i+1}) = x_i'$
2) right-directed: $\Gamma(x)_i = \delta_r(x_{i-1}, x_i, x_{i+1}, x_{i+2}) = x_{i+1}'$
3) left-directed: $\Gamma(x)_i = \delta_l(x_{i-2}, x_{i-1}, x_i, x_{i+1}) = x_{i-1}'$
4) left&right directed: $= \delta_{rl}(x_{i-2}, x_{i-1}, x_i, x_{i+1}) = (x_{i-1}', x_i')$

Combinatorics for morphoCAs

Introduction

General morphoCA schemes

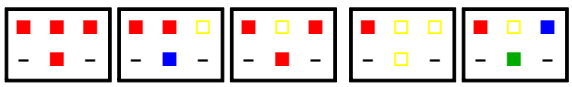
```
Presentation[
 ArrayPlot[CellularAutomaton[
  ruleDef (MG^(m,n)), morphoRules, ruleSpace
  {init}, steps],
 {Graphics}]]

presentation =
 {TabView, SlideViev, MenuView, etc}
```

How does is work? Relabeling and separation.

```
ReLabel[L_List] :=
 L /. Map[#[[1]] → #[[2]] &,
   Transpose[{DeleteDuplicates[L],
     Range[Length[Union[L]]]}]]
```

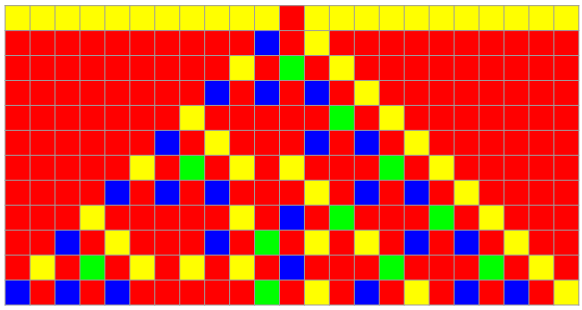
Example

```
ruleM[{1, 11, 3, 9, 15}]
```



```
ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green}
```

```
ArrayPlot[CellularAutomaton[
  ruleM[{1, 11, 3, 9, 15}],
  {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green}, Mesh → True]
```



Step 1

```
ArrayPlot[CellularAutomaton[
  ruleM[{1, 11, 3, 9, 15}],
  {{1}, 0}, 1],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green}, Mesh → True]
```

=

0	1	0
2	1	0

[0, 1, 0] corresponds : [1, 0, 1], hence the next step is [1, 0, 1] -> 0 or [1, 0, 1] -> 1.
Because rule 3 is involved and not rule 8, the next step is 1 and not 0.

```
Mod[ReLabel[{0, 1, 0}], 2]
```

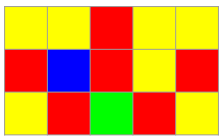
```
{1, 0, 1}
```

Hence,

0	1	0
–	1	–

Step 2

```
ArrayPlot[CellularAutomaton[
  ruleM[{1, 11, 3, 9, 15}],
  {{1}, 0}, 2],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green}, Mesh → True]
```



0	0	1	0	0
□	2	1	0	□
□	□	3	□	□

[2,1,0] corresponds by relabeling to [1,2,0], hence the next step is: [1,2,0] -> 0 or [1,2,0] -> 1 or [1,2,0] -> 2 or [1,2,0] -> 3.
In contrast to a set-based approach, there are no other possibilities involved on the level of morphogrammatics.
Because rule 15 is involved and not rule 5 or rule 10 or rule 14, the next step is 3 and not 0, 1 or 2.

The head of the morphogram

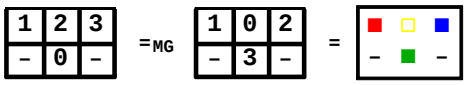
```
Mod[ReLabel[{2, 1, 0}], 3]
```

{1, 2, 0}

The morphogram 15

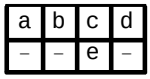
```
Mod[ReLabel[{2, 1, 0, 3}], 4]
```

{1, 2, 3, 0}

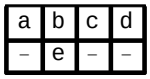


Extensions

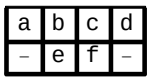
The mechanism for $CA^{(3,2)}$ is naturally *extended* to other architectures.
An example with an asymmetric rule scheme for $CA^{(5,5)}$ is given by the following scheme.

RuleSchemeR: 

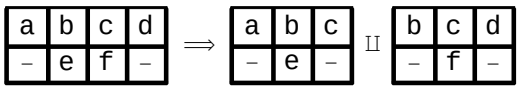
The dual scheme is naturally given by:

RuleSchemeL: 

And there is a double scheme too:

RuleSchemeRL: 

Decomposition of RuleSchemRL



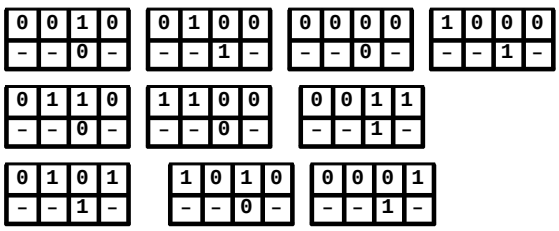
Architectonic complementarity

Additionally to the well known *duality* of classical CA rules, a new and more architectural duality or complementarity is introduced with the right- and left-definition of morphoCA rules.

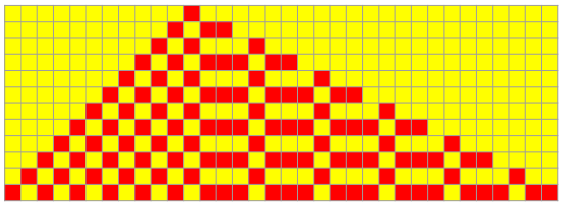
This architectonic complementarity is a basic feature of general morphic CAs.

Examples for Architectonic Complementarity

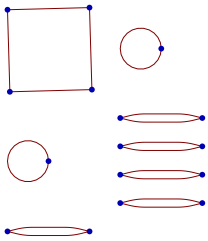
Right - oriented CA



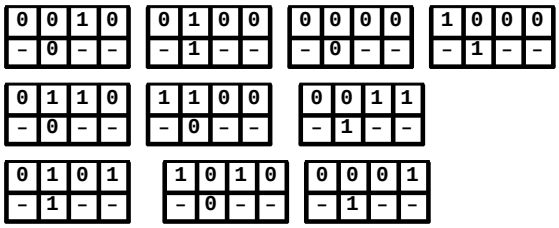
```
ArrayPlot[CellularAutomaton[
  ruleDCKV10[
   {11211, 12112, 11111, 12221, 12211, 11222, 12122, 11222, 12122, 11122}],
  {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 300]
```



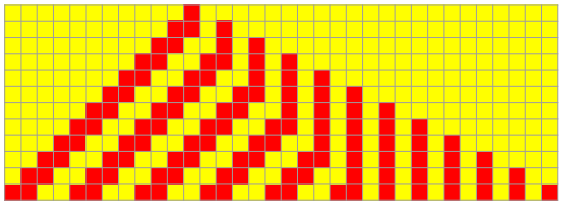
```
GraphPlot[# -> CellularAutomaton[
      ruleDCKV10[{11211, 12112, 11111,
        12221, 12211, 11222, 12122, 11222, 12122, 11122}], #]
   & /@ Tuples[{0, 1}, 4]]
```



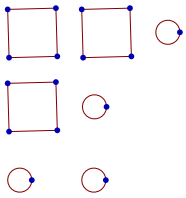
Left - oriented CA



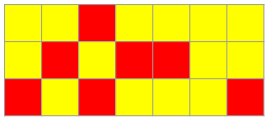

```
ArrayPlot[CellularAutomaton[
  ruleDCKV16[{12111, 12221, 11112, 12121, 12111, 12211, 11222, 12221, 12121,
    12111, 11122, 11122, 12221, 11122, 11212, 11111}] , {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 300]
```



```
GraphPlot[# -> CellularAutomaton[
      ruleDCKV16[{12111, 12221, 11112, 12121, 12111, 12211, 11222, 12221,
        12121, 12111, 11122, 11122, 12221, 11122, 11212, 11111}], #]
   & /@ Tuples[{0, 1}, 4]]
```

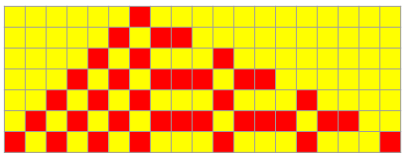


```
{0, 1, 0, 0} → 0
{1, 0, 0, 0} → 1
{0, 0, 0, 0} → 1
```



```
{0, 1, 0, 1} → 0
{1, 0, 1, 1} → 1
{0, 1, 1, 0} → 0
{1, 1, 0, 0} → 0
{1, 0, 0, 0} → 1
```

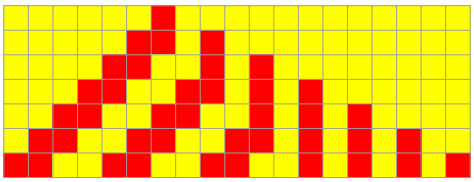
```
ArrayPlot[CellularAutomaton[
  ruleDCKV10[
   {11211, 12112, 11111, 12221, 12211, 11222, 12122, 11222, 12122, 11122}],
  {{1}, 0}, 6],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 300]
```



```
{1, 0, 1, 0} → 1
{1, 0, 1, 1} → 1
{0, 0, 0, 1} → 1
{0, 0, 1, 0} → 1
{1, 0, 0, 0} → 1
{1, 1, 1, 0} → 0
{1, 1, 0, 1} → 0
{1, 1, 1, 1} → 1
```

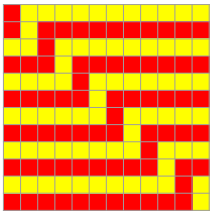
The subset of the rule specification, {11112, 12121,12121,11212 }, is defined by the application of the left-rules scheme and marks the difference between the previous right - oriented CA. Without the information produced by the *application*, both are trivially equal. Until now, the rules are not yet defined accordingly.

```
ArrayPlot[CellularAutomaton[
  ruleDCKV10[
   {12 111, 12 221, 11 112, 12 121, 12 211, 11 222, 12 121, 11 122, 11 212, 11 111}] ,
  {{1}, 0}, 6],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 300]
```



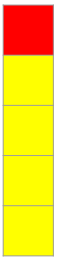
Given the graphic representation, different interpretations are obviously possible.

Classical $CA^{(3,2)}$ interpretation



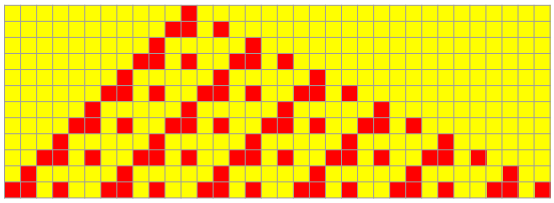
straight – directed morphoCA$^{(3,2)}$

```
ArrayPlot[CellularAutomaton[
  ruleCl[{6, 7, 8, 9}],
  {{1}, 0}, 4],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 100]
```



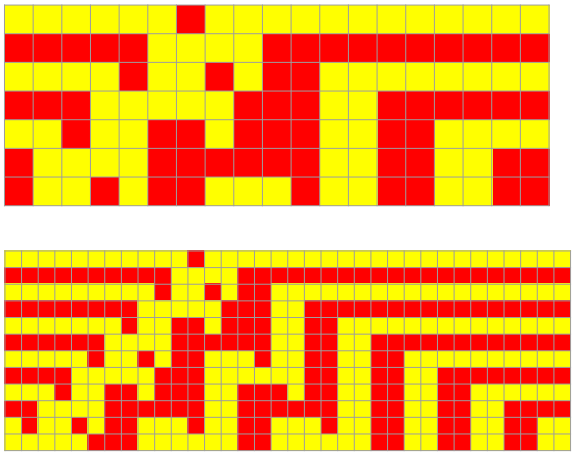
Further example

```
ArrayPlot[CellularAutomaton[
  ruleDCKV8[{11 111, 11 122, 11 212, 12 111, 11 221,  12 122, 12 211,  12 221}],
  {{1}, 0}, {11}], ColorRules -> {1 -> Red, 0 -> Yellow},  Mesh → True]
```

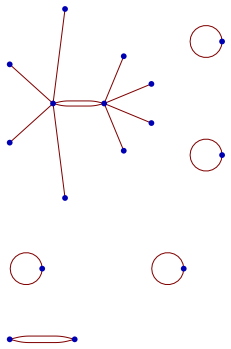


Right – interpretation of the Left – Description by the implemented *ruleDCKVs*.

```
ArrayPlot[CellularAutomaton[
  ruleDCKV10[
   {11 112, 12 222, 12 122, 11 211, 12 212, 11 222, 11 212, 11 212, 11 121, 12 111 }],
  {{1}, 0}, {6}], ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True]
```

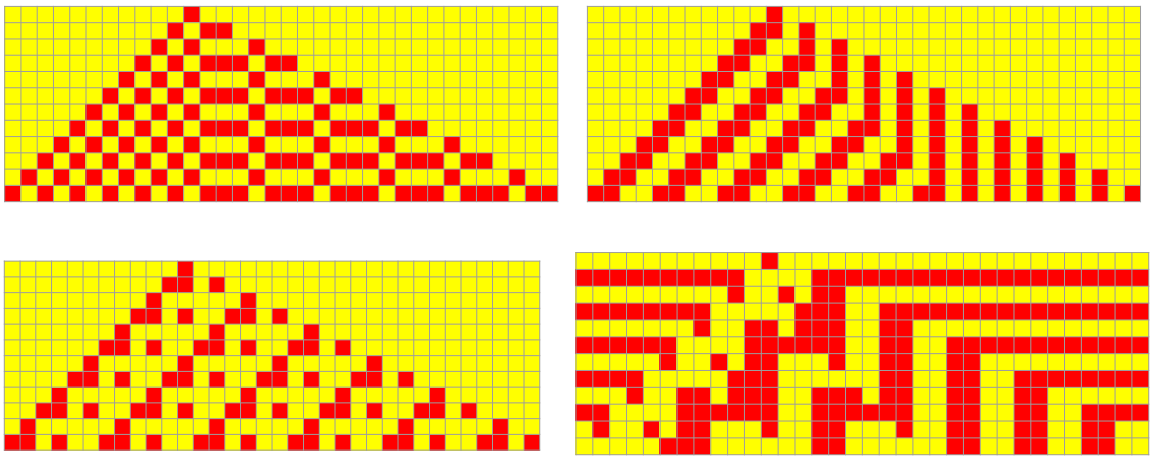


```
GraphPlot[# -> CellularAutomaton[
     ruleDCKV10[{11 112, 12 222, 12 122,
       11 211, 12 212, 11 222, 11 212, 11 212, 11 121, 12 111 }], #]
   & /@ Tuples[{0, 1}, 4]]
```



Comparison : Complementarity of right – and left – oriented CAs

Right-oriented CA **Left - oriented CA**



New morphoCA Rules

RulesSchemes $CA^{(5,4,5)}$

Junctional morphoRules $CA^{(5,2,2)}$

Junctional morphoRules $CA^{(5,2,5)}$

Transjunctional morphoRules $CA^{(5,3,5)}$

Transjunctional morphoRules $CA^{(5,4,5)}$

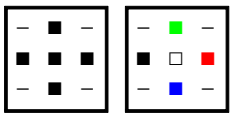
Morphogram tables

Morphograms $MG^{(5,5)}$

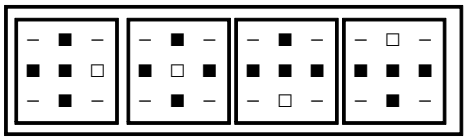
```
-Tcontexture 5;

[[1, 1, 1, 1, 1],
 [1, 1, 1, 2, 2], [1, 1, 2, 1, 2], [1, 1, 2, 2, 1],
 [1, 2, 1, 1, 2], [1, 2, 1, 2, 1], [1, 2, 2, 1, 1], [1, 2, 2, 2, 1],
 [1, 2, 2, 1, 2], [1, 2, 1, 2, 2], [1, 1, 2, 2, 2], [1, 1, 1, 1, 2],
 [1, 1, 1, 2, 1], [1, 1, 2, 1, 1], [1, 2, 1, 1, 1], [1, 2, 2, 2, 2],

 [1, 1, 2, 2, 3], [1, 1, 2, 3, 2], [1, 1, 2, 3, 3], [1, 2, 1, 2, 3],
 [1, 2, 1, 3, 2], [1, 2, 2, 1, 3], [1, 2, 2, 3, 1], [1, 2, 3, 1, 2],
 [1, 2, 3, 2, 1], [1, 2, 1, 3, 3], [1, 2, 3, 1, 3], [1, 2, 3, 3, 1],
 [1, 2, 2, 3, 3], [1, 2, 3, 2, 3], [1, 2, 3, 3, 2], [1, 1, 1, 2, 3],
 [1, 1, 2, 1, 3], [1, 1, 2, 3, 1], [1, 2, 1, 1, 3], [1, 2, 1, 3, 1],
 [1, 2, 3, 1, 1], [1, 2, 2, 2, 3], [1, 2, 2, 3, 2], [1, 2, 3, 2, 2],

 [1, 2, 3, 3, 3], [1, 1, 2, 3, 4], [1, 2, 1, 3, 4], [1, 2, 3, 1, 4],
 [1, 2, 3, 4, 1], [1, 2, 2, 3, 4], [1, 2, 3, 2, 4], [1, 2, 3, 4, 2],
 [1, 2, 3, 3, 4], [1, 2, 3, 4, 3], [1, 2, 3, 4, 4],
 [1, 2, 3, 4, 5]]

Table[StirlingS2[5, m], {m, 5}]
```

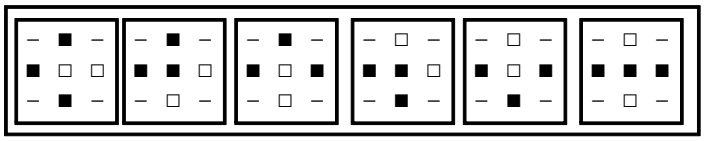
CA(5, 1), CA(5, 5) = 1

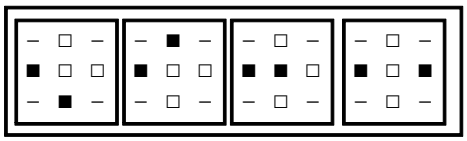
```
[1, 1, 1, 1, 1] [1, 2, 3, 4, 5]
```

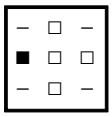


CA(5, 2) = 15 : {■, □}

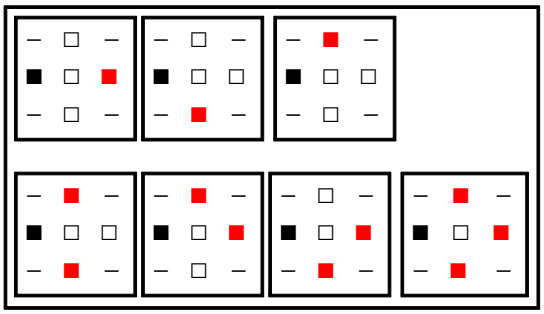


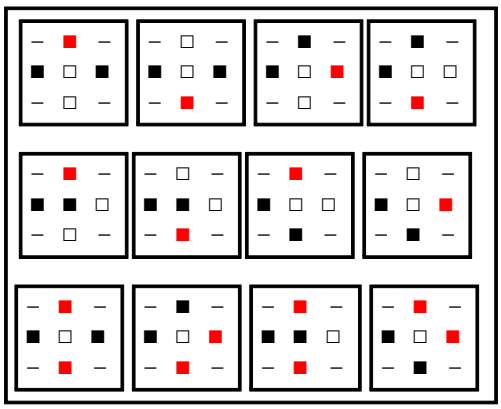


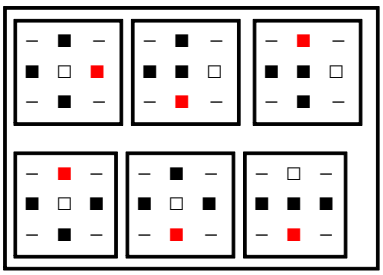




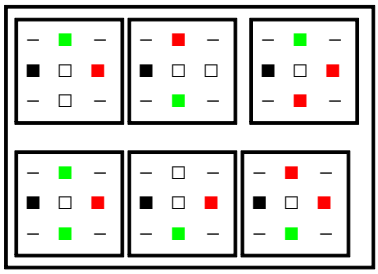
CA $^{(5,3)}$ = 25 : {■, □, ■}

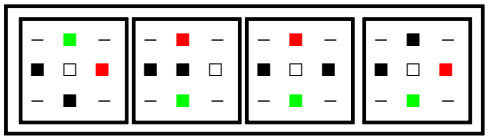






CA $^{(5,4)}$ = 10 : {■, □, ■, ■}





TabView of morphograms MG$^{(6,6)}$

6 | (5,1) | (4,2) | (4,1,1) | (3,3) | (2,2,2) | (2,2,1,1) | (2,1,1,1,1) | (1,1,1,1,1,1)

```
(15=10+4+1):

[1,1,1,1,2,3],[1,1,1,2,1,3],[1,1,1,2,3,1],[1,1,2,1,1,3],[1,1,2,1,3,1],
[1,1,2,3,1,1],[1,2,1,1,1,3],[1,2,1,1,3,1],[1,2,1,3,1,1],[1,2,3,1,1,1];

[1,2,2,2,2,3],[1,2,2,2,3,2],[1,2,2,3,2,2],[1,2,3,2,2,2];

[1,2,3,3,3,3]
```

Refinements of morphograms MG$^{(6,6)}$

No. Partition morphograms

• 6 : 1 [1,1,1,1,1,1]

• (5,1) : 6 = 5+1
[1,1,1,1,1,2],[1,1,1,1,2,1],[1,1,1,2,1,1],[1,1,2,1,1,1],[1,2,1,1,1,1];
[1,2,2,2,2,2].

• (4,2) : 15 = 10+5
[1,1,1,1,2,2],[1,1,1,2,1,2],[1,1,1,2,2,1],[1,1,2,1,1,2],[1,1,2,1,2,1],
[1,1,2,2,1,1],[1,2,1,1,1,2],[1,2,1,1,2,1],[1,2,1,2,1,1],[1,2,2,1,1,1];
[1,2,2,2,2,1],[1,2,2,2,1,2],[1,2,2,1,2,2],[1,2,1,2,2,2],[1,1,2,2,2,2].

• (4,1,1) : 15 = 10+4+1
[1,1,1,1,2,3],[1,1,1,2,1,3],[1,1,1,2,3,1],[1,1,2,1,1,3],[1,1,2,1,3,1],
[1,1,2,3,1,1],[1,2,1,1,1,3],[1,2,1,1,3,1],[1,2,1,3,1,1],[1,2,3,1,1,1];
[1,2,2,2,2,3],[1,2,2,2,3,2],[1,2,2,3,2,2],[1,2,3,2,2,2]; [1,2,3,3,3,3].

• (3,3) : 10
[1,1,1,2,2,2],[1,1,2,1,2,2],[1,1,2,2,1,2],[1,1,2,2,2,1],[1,2,1,1,2,2],
[1,2,1,2,1,2],[1,2,1,2,2,1],[1,2,2,1,1,2],[1,2,2,1,2,1],[1,2,2,2,1,1].

• (3,2,1) : 60 = 30+15+15
[1,1,1,2,2,3],[1,1,1,2,3,2],[1,1,1,2,3,3], (30)
[1,1,2,1,2,3],[1,1,2,1,3,2],[1,1,2,2,1,3],[1,1,2,2,3,1],[1,1,2,3,1,2],
[1,1,2,3,2,1],[1,1,2,1,3,3],[1,1,2,3,1,3],[1,1,2,3,3,1],[1,2,1,1,2,3],
[1,2,1,1,3,2],[1,2,1,2,1,3],[1,2,1,2,3,1],[1,2,1,3,1,2],[1,2,1,3,2,1],

[1,2,2,1,1,3],[1,2,2,1,3,1],[1,2,2,3,1,1],[1,2,3,1,1,2],[1,2,3,1,2,1],
[1,2,3,2,1,1],[1,2,1,1,3,3],[1,2,1,3,1,3],[1,2,1,3,3,1],[1,2,3,1,1,3],
[1,2,3,1,3,1],[1,2,3,3,1,1];
[1,2,2,2,1,3],[1,2,2,2,3,1],[1,2,2,1,2,3], (15)
[1,2,2,1,3,2],[1,2,2,3,2,1],[1,2,2,3,1,2],[1,2,1,2,2,3],[1,2,1,2,3,2],
[1,2,1,3,2,2],[1,2,3,2,2,1],[1,2,3,2,1,2],[1,2,3,1,2,2],[1,1,2,2,2,3],
[1,1,2,2,3,2],[1,1,2,3,2,2];
[1,1,2,3,3,3],[1,2,3,3,3,1],[1,2,3,3,1,3], (15)
[1,2,3,1,3,3],[1,2,1,3,3,3],[1,2,2,2,3,3],[1,2,2,3,2,3],[1,2,2,3,3,2],
[1,2,3,2,2,3],[1,2,3,2,3,2],[1,2,3,3,2,2],[1,2,3,3,3,2],[1,2,3,3,2,3],
[1,2,3,2,3,3],[1,2,2,3,3,3].

• (3,1,1,1) : 20 = 10+6+3+1
[1,1,1,2,3,4],[1,1,2,1,3,4],[1,1,2,3,1,4],[1,1,2,3,4,1],[1,2,1,1,3,4],
[1,2,1,3,1,4],[1,2,1,3,4,1],[1,2,3,1,1,4],[1,2,3,1,4,1],[1,2,3,4,1,1];
[1,2,2,2,3,4],[1,2,2,3,2,4],[1,2,2,3,4,2],[1,2,3,2,2,4],[1,2,3,2,4,2],
[1,2,3,4,2,2];
[1,2,3,3,3,4],[1,2,3,3,4,3],[1,2,3,4,3,3];
[1,2,3,4,4,4],

• (2,2,2) :15
[1,1,2,2,3,3],[1,1,2,3,2,3],[1,1,2,3,3,2],[1,2,1,2,3,3],[1,2,1,3,2,3],
[1,2,1,3,3,2],[1,2,2,1,3,3],[1,2,2,3,1,3],[1,2,2,3,3,1],[1,2,3,1,2,3],
[1,2,3,1,3,2],[1,2,3,2,1,3],[1,2,3,2,3,1],[1,2,3,3,1,2],[1,2,3,3,2,1],

• (2,2,1,1) : 45 = 30+12+3
[1,1,2,2,3,4],[1,1,2,3,2,4],[1,1,2,3,4,2],
[1,1,2,3,3,4],[1,1,2,3,4,3],[1,1,2,3,4,4],[1,2,1,2,3,4],[1,2,1,3,2,4],
[1,2,1,3,4,2],[1,2,2,1,3,4],[1,2,2,3,1,4],[1,2,2,3,4,1],[1,2,3,1,2,4],
[1,2,3,1,4,2],[1,2,3,2,1,4],[1,2,3,2,4,1],[1,2,3,4,1,2],[1,2,3,4,2,1],
[1,2,1,3,3,4],[1,2,1,3,4,3],[1,2,1,3,4,4],[1,2,3,1,3,4],[1,2,3,1,4,3],
[1,2,3,3,1,4],[1,2,3,3,4,1],[1,2,3,4,1,3],[1,2,3,4,3,1],[1,2,3,1,4,4],
[1,2,3,4,1,4],[1,2,3,4,4,1]
[1,2,2,3,3,4],[1,2,2,3,4,3],[1,2,2,3,4,4],[1,2,3,2,3,4],[1,2,3,2,4,3],
[1,2,3,3,2,4],[1,2,3,3,4,2],[1,2,3,4,2,3],[1,2,3,4,3,2],[1,2,3,2,4,4],
[1,2,3,4,2,4],[1,2,3,4,4,2]
[1,2,3,3,4,4],[1,2,3,4,3,4],[1,2,3,4,4,3]

• (2,1,1,1,1) : 15 = 5+4+3+2+1
[1,1,2,3,4,5],[1,2,1,3,4,5],[1,2,3,1,4,5],[1,2,3,4,1,5],[1,2,3,4,5,1];
[1,2,2,3,4,5],[1,2,3,2,4,5],[1,2,3,4,2,5],[1,2,3,4,5,2];
[1,2,3,3,4,5],[1,2,3,4,3,5],[1,2,3,4,5,3];
[1,2,3,4,4,5],[1,2,3,4,5,4];
[1,2,3,4,5,5].

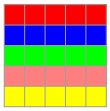
• (1,1,1,1,1,1) : 1
[1,2,3,4,5,6] .

First Refinement	1	6	15	15	10	60	20	15	45	15	1
Second Refinement	1	5+1	10+5	10+4+1	10	30+15+15	10+6+3+1	15	30+12+3	5+4+3+2+1	1
Stirling	1	□	31	□	90	□	65	□	□	15	1

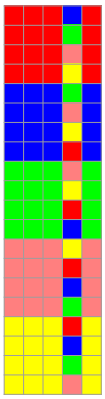
ArrayPlot of Morphograms

ArrayPlot of MorphoRules CA$^{(5,2)}$

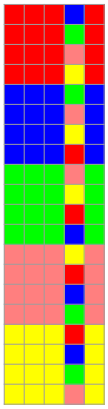
```
ArrayPlot[
 dcl[{11111}],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green, 4 → Pink}, Mesh → True]
```



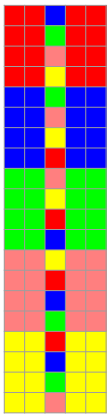
```
ArrayPlot[
 dcl[{11 112}],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green, 4 → Pink}, Mesh → True]
```



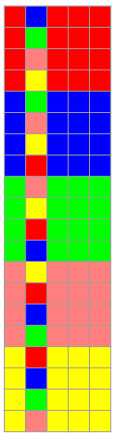
```
ArrayPlot[
 dcl[{11 121}],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green, 4 → Pink}, Mesh → True]
```



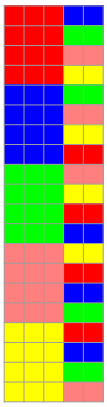
```
ArrayPlot[
 dcl[{11 211}],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green, 4 → Pink}, Mesh → True]
```



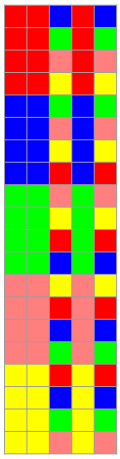
```
ArrayPlot[
 dcl[{12111}],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green, 4 → Pink}, Mesh → True]
```



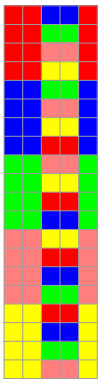
```
ArrayPlot[
 dcl[{11122}],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green, 4 → Pink}, Mesh → True]
```



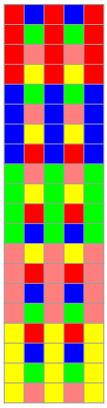
```
ArrayPlot[
 dcl[{11212}],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green, 4 → Pink}, Mesh → True]
```



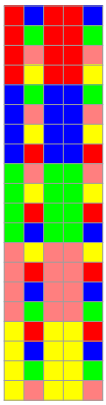
```
ArrayPlot[
 dcl[{11221}],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green, 4 → Pink}, Mesh → True]
```



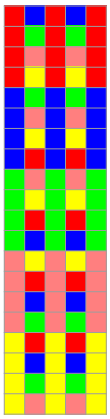
```
ArrayPlot[
 dcl[{12121}],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green, 4 → Pink}, Mesh → True]
```



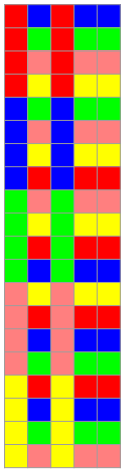
```
ArrayPlot[
 dcl[{12112}],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green, 4 → Pink}, Mesh → True]
```



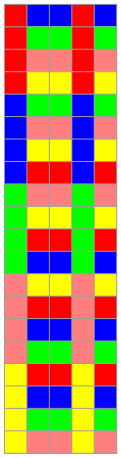
```
ArrayPlot[
 dcl[{12121}],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green, 4 → Pink}, Mesh → True]
```



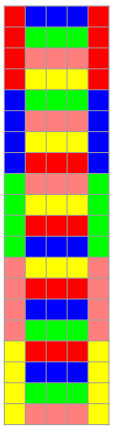
```
ArrayPlot[
 dcl[{12122}],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green, 4 → Pink}, Mesh → True]
```



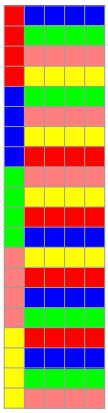
```
ArrayPlot[
 dcl[{12212}],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green, 4 → Pink}, Mesh → True]
```



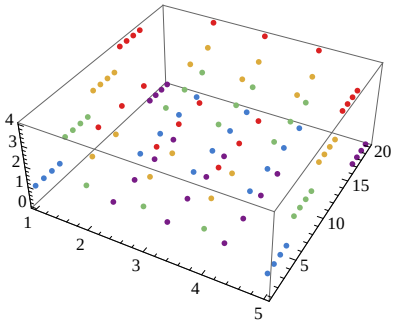
```
ArrayPlot[
 dcl[{12 221}],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green, 4 → Pink}, Mesh → True]
```



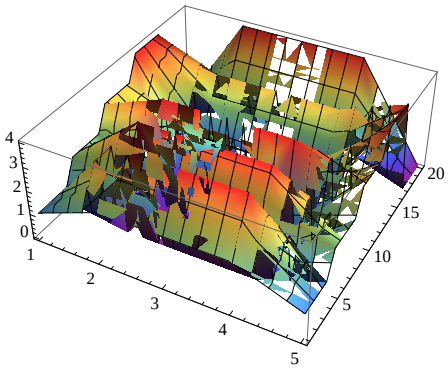
```
ArrayPlot[
 dcl[{12 222}],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green, 4 → Pink}, Mesh → True]
```



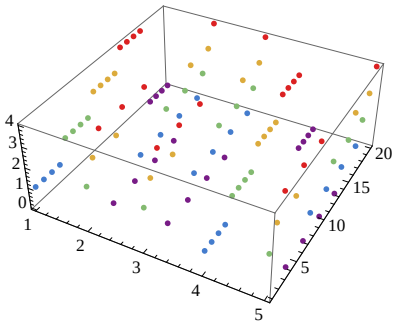
```
ListPointPlot3D[dcl[{12 221}], ColorFunction → "Rainbow"]
```



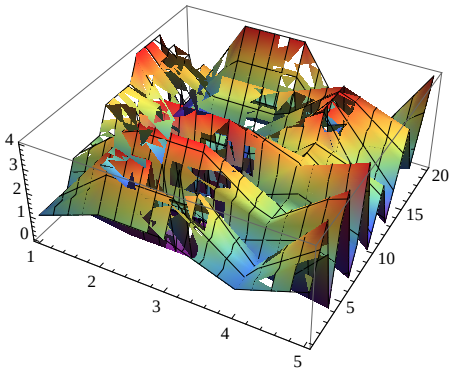
```
ListPlot3D[dcl[{12 221}], ColorFunction → "Rainbow", Mesh → True]
```



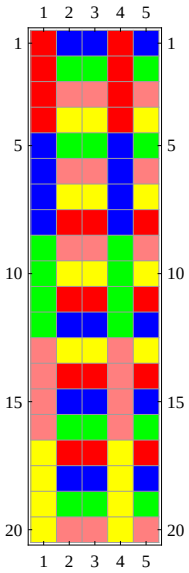

```
ListPointPlot3D[dcl[{12 212}], ColorFunction → "Rainbow"]
```

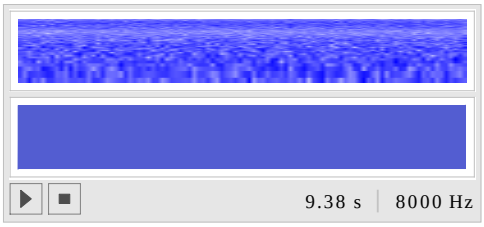



```
ListPlot3D[dcl[{12 212}], ColorFunction → "Rainbow", Mesh → True]
```



```
MatrixPlot[dcl[{12212}],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green, 4 → Pink}, Mesh → True]
```

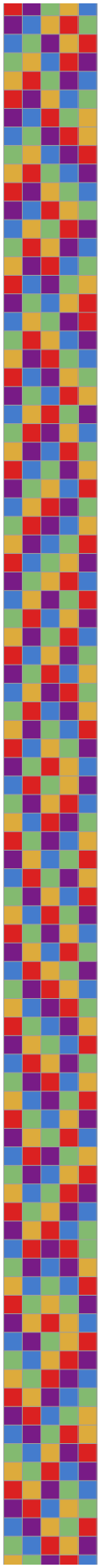




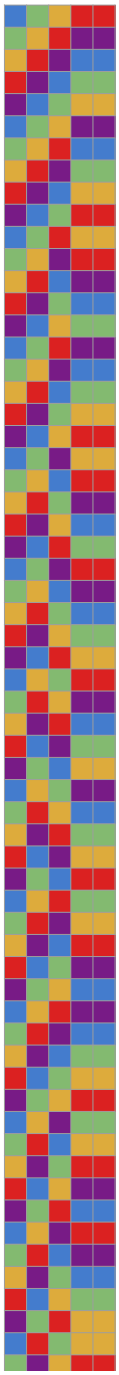
ArrayPlot of MorphoRules CA$^{(5,4)}$

```
[1, 1, 2, 3, 4],
[1, 2, 1, 3, 4],
[1, 2, 3, 4, 1],
[1, 2, 2, 3, 4],
[1, 2, 3, 2, 4],
[1, 2, 3, 1, 4],
[1, 2, 3, 4, 2],
[1, 2, 3, 3, 4],
[1, 2, 3, 4, 3],
[1, 2, 3, 4, 4],
[1, 2, 3, 4, 5]
```

```
ArrayPlot[Map[Flatten, dckv[{12345}] /. Rule → List, 1],
 ColorFunction → "Rainbow", ImageSize → 60, Mesh → True]
```

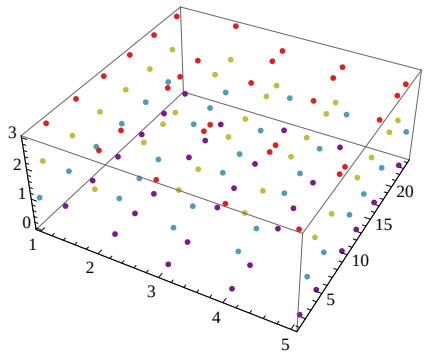


```
ArrayPlot[Map[Flatten, dckv[{12 344}] /. Rule → List, 1],
 ColorFunction → "Rainbow", ImageSize → 60, Mesh → True]
```

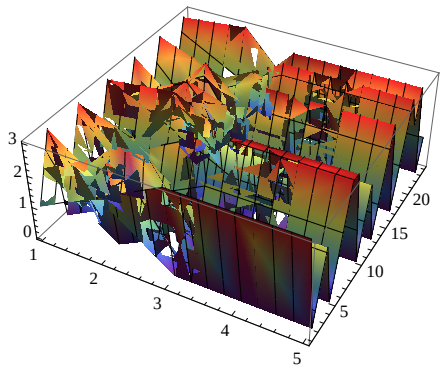


Graphics and Sound examples of morphoCA(5,3)

```
ListPointPlot3D[dckv[{12 333}], ColorFunction → "Rainbow"]
```




```
ListPlot3D[dckv[{12 333}], ColorFunction → "Rainbow", Mesh → True]
```

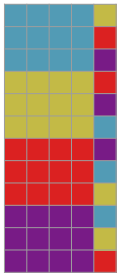


ArrayPlot of MorphoRules CA(5,3)

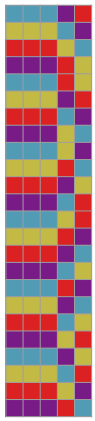
```
dckv[{11 112}]
```

```
Flatten[Table[Mod[{i, i, i, i}, 4] → Mod[ i + j, 4], {i, 4}, {j, 3}], 1]
```

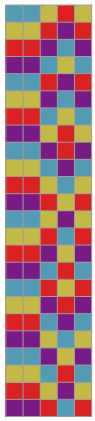
```
ArrayPlot[Map[Flatten, dckv[{11 112}] /. Rule → List, 1],
 ColorFunction → "Rainbow", ImageSize → Small, Mesh → True]
```



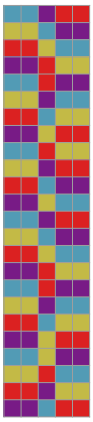
```
ArrayPlot[Map[Flatten, dckv[{11 123}] /. Rule → List, 1],
 ColorFunction → "Rainbow", ImageSize → Small, Mesh → True]
```



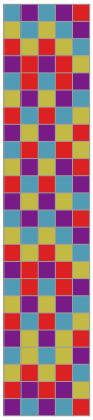
```
ArrayPlot[Map[Flatten, dckv[{11 232}] /. Rule → List, 1],
 ColorFunction → "Rainbow", ImageSize → Small, Mesh → True]
```



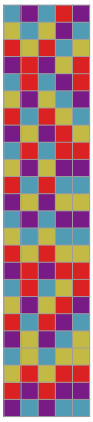
```
ArrayPlot[Map[Flatten, dckv[{11 233}] /. Rule → List, 1],
 ColorFunction → "Rainbow", ImageSize → Small, Mesh → True]
```



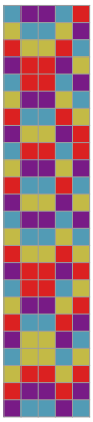
```
ArrayPlot[Map[Flatten, dckv[{12 123}] /. Rule → List, 1],
 ColorFunction → "Rainbow", ImageSize → Small, Mesh → True]
```



```
ArrayPlot[Map[Flatten, dckv[{12 132}] /. Rule → List, 1],
 ColorFunction → "Rainbow", ImageSize → Small, Mesh → True]
```

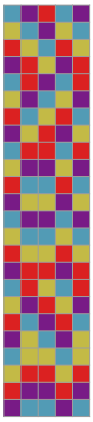


```
ArrayPlot[Map[Flatten, dckv[{12 213}] /. Rule → List, 1],
 ColorFunction → "Rainbow", ImageSize → Small, Mesh → True]
```

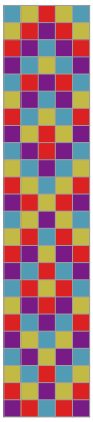


```
ArrayPlot[Map[Flatten, dckv[{12 231}] /. Rule → List, 1],
 ColorFunction → "Rainbow", ImageSize → Small, Mesh → True]
```

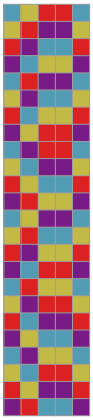
```
ArrayPlot[Map[Flatten, dckv[{12 312}] /. Rule → List, 1],
 ColorFunction → "Rainbow", ImageSize → Small, Mesh → True]
```



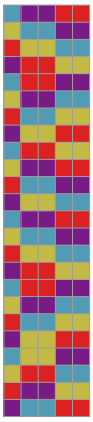
```
ArrayPlot[Map[Flatten, dckv[{12 321}] /. Rule → List, 1],
 ColorFunction → "Rainbow", ImageSize → Small, Mesh → True]
```



```
ArrayPlot[Map[Flatten, dckv[{12 331}] /. Rule → List, 1],
 ColorFunction → "Rainbow", ImageSize → Small, Mesh → True]
```

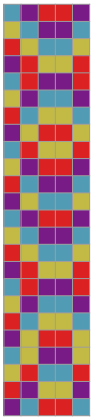


```
ArrayPlot[Map[Flatten, dckv[{12 233}] /. Rule → List, 1],
 ColorFunction → "Rainbow", ImageSize → Small, Mesh → True]
```



```
ArrayPlot[Map[Flatten, dckv[{12 323}] /. Rule → List, 1],
 ColorFunction → "Rainbow", ImageSize → Small, Mesh → True]
```

```
ArrayPlot[Map[Flatten, dckv[{12 332}] /. Rule → List, 1],
 ColorFunction → "Rainbow", ImageSize → Small, Mesh → True]
```



```
ArrayPlot[Map[Flatten, dckv[{11 123}] /. Rule → List, 1],
 ColorFunction → "Rainbow", ImageSize → Small, Mesh → True]
```

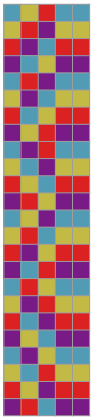
```
ArrayPlot[Map[Flatten, dckv[{11 213}] /. Rule → List, 1],
 ColorFunction → "Rainbow", ImageSize → Small, Mesh → True]
```

```
ArrayPlot[Map[Flatten, dckv[{11 231}] /. Rule → List, 1],
 ColorFunction → "Rainbow", ImageSize → Small, Mesh → True]
```

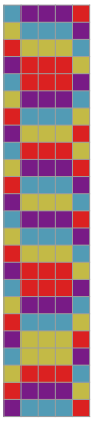
```
ArrayPlot[Map[Flatten, dckv[{12 113}] /. Rule → List, 1],
 ColorFunction → "Rainbow", ImageSize → Small, Mesh → True]
```

```
ArrayPlot[Map[Flatten, dckv[{12 131}] /. Rule → List, 1],
 ColorFunction → "Rainbow", ImageSize → Small, Mesh → True]
```

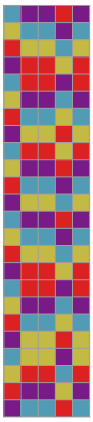
```
ArrayPlot[Map[Flatten, dckv[{12 311}] /. Rule → List, 1],
 ColorFunction → "Rainbow", ImageSize → Small, Mesh → True]
```



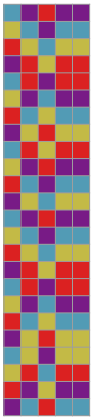
```
ArrayPlot[Map[Flatten, dckv[{12 223}] /. Rule → List, 1],
 ColorFunction → "Rainbow", ImageSize → Small, Mesh → True]
```



```
ArrayPlot[Map[Flatten, dckv[{12 232}] /. Rule → List, 1],
 ColorFunction → "Rainbow", ImageSize → Small, Mesh → True]
```



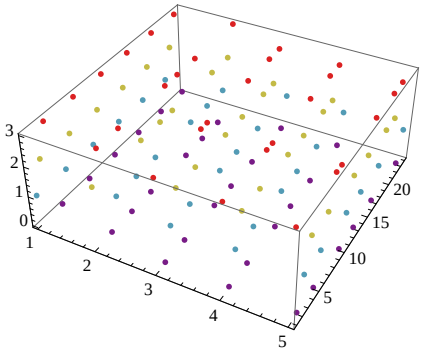
```
ArrayPlot[Map[Flatten, dckv[{12 322}] /. Rule → List, 1],
 ColorFunction → "Rainbow", ImageSize → Small, Mesh → True]
```



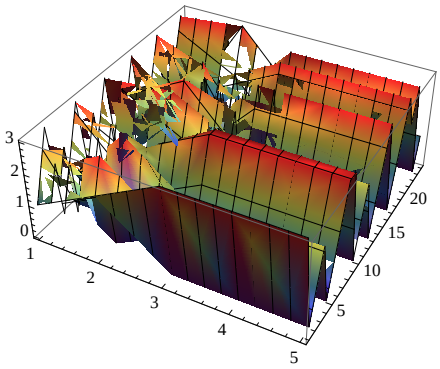
```
ArrayPlot[Map[Flatten, dckv[{12 333}] /. Rule → List, 1],
 ColorFunction → "Rainbow", ImageSize → Small, Mesh → True]
```

Graphics and Sound examples of morphoCA$^{(5,3)}$

```
ListPointPlot3D[dckv[{12 333}], ColorFunction → "Rainbow"]
```




```
ListPlot3D[dckv[{12 333}], ColorFunction → "Rainbow", Mesh → True]
```




```
ListPlay[Map[Flatten, dckv[{11 123}] /. Rule → List, 1], 20 000]
```

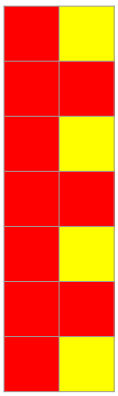
CA-Examples, right-oriented

Catalog of CA$^{(5,2)}$ Examples

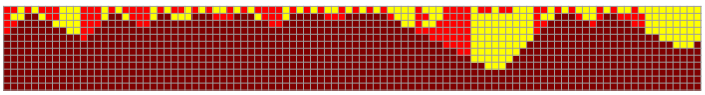
Morphogram sequences for morphoCA$^{(5,2)}$

```
{1, 0, 0, 0, 0}, {1, 0, 0, 0, 1}, {1, 0, 0, 1, 0},
{1, 0, 0, 1, 1}, {1, 0, 1, 0, 0}, {1, 0, 1, 0, 1},
{1, 0, 1, 1, 0}, {1, 0, 1, 1, 1}, {1, 1, 0, 0, 0},
{1, 1, 0, 0, 1}, {1, 1, 0, 1, 0}, {1, 1, 0, 1, 1},
{1, 1, 1, 0, 0}, {1, 1, 1, 0, 1}, {1, 1, 1, 1, 0}, {1, 1, 1, 1, 1}
```

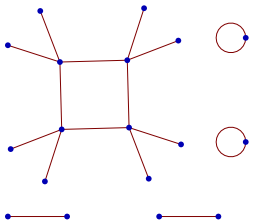
```
ArrayPlot[CellularAutomaton[ruleDCKV10[{11111, 11121, 11122,
    11222, 12211, 12212, 11112, 12112, 11212, 12222}], {{1}, 0}, 6],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 300]
```



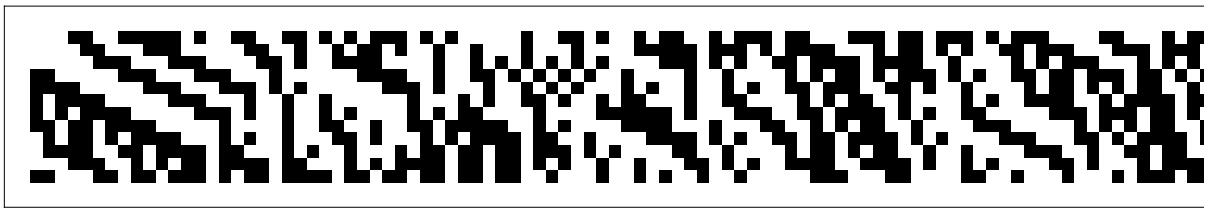
```
ArrayPlot[
 CellularAutomaton[ruleDCKV10[{11111, 11121, 11122, 11222, 12211, 12212,
    11112, 12112, 11212, 12222}] , RandomInteger[1, 100], 11],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green},
 Mesh → True, ImageSize → 300]
```



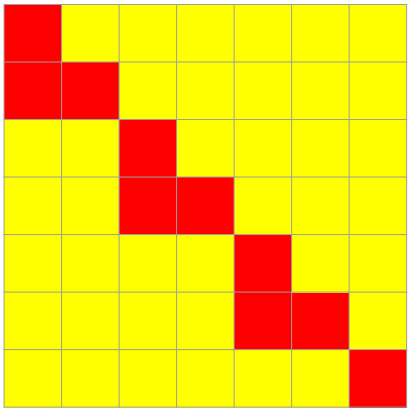
```
GraphPlot[
 # -> CellularAutomaton[ruleDCKV10[{11111, 11121, 11122, 11222, 12211, 12212,
       11112, 12112, 11212, 12222}], #] & /@ Tuples[{0, 1}, 4]]
```



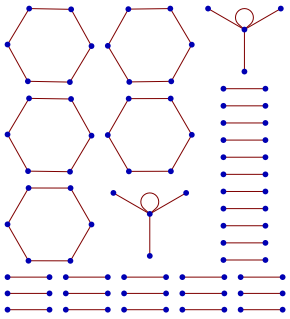
```
ArrayPlot[CellularAutomaton[
  ruleDCKV14[{11111, 11221, 12212, 12111, 11212, 12122, 12121, 12221, 11222,
    11121, 12122, 12221, 11222, 12212}], RandomInteger[1, 100], 11],
 Mesh → False, ImageSize → 100]
```



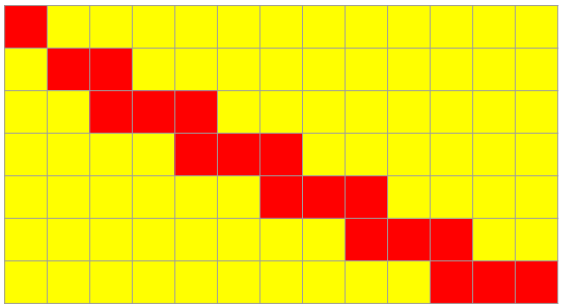
```
ArrayPlot[CellularAutomaton[
  ruleDCKV10[
   {11111, 11121, 11122, 11221, 12211, 12212, 11112, 12112, 11212, 12222}],
  {{1}, 0}, 6],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 300]
```



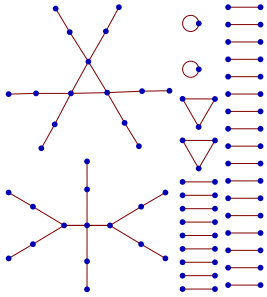
```
GraphPlot[# -> CellularAutomaton[
      ruleDCKV10[{11 111, 11 121, 11 122,
        11 221, 12 211, 12 212, 11 112, 12 112, 11 212, 12 222}], #]
   & /@ Tuples[{0, 1}, 6]]
```



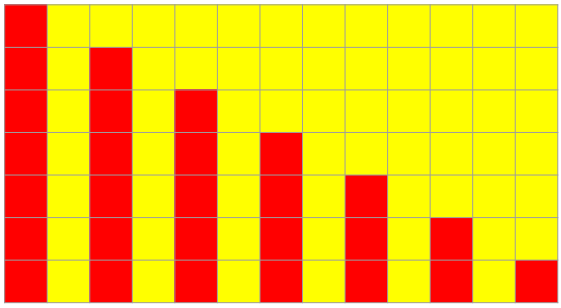
```
ArrayPlot[CellularAutomaton[ruleDCKV8[
   {11 111, 11 221, 11 121, 12 221,  12 112, 12 212, 11 112, 11 211}] , {{1}, 0}, 6],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 300]
```



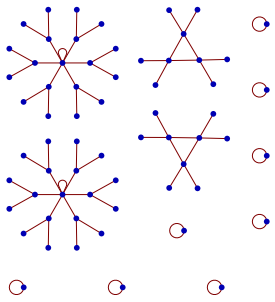
```
GraphPlot[# -> CellularAutomaton[
      ruleDCKV8[{11 111, 11 221, 11 121, 12 221, 12 112, 12 212, 11 112, 11 211}], #]
   & /@ Tuples[{0, 1}, 6]]
```



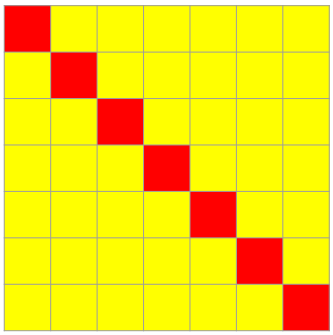
```
ArrayPlot[CellularAutomaton[ruleDCKV10[{11 111, 11 221, 11 121,
    12 111, 11 212, 12 221, 12 212, 11 112, 11 122, 12 121}], {{1}, 0}, 6],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 300]
```



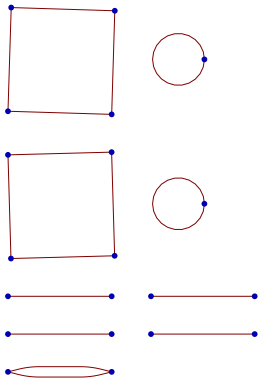
```
GraphPlot[# -> CellularAutomaton[
      ruleDCKV10[{11 111, 11 221, 11 121,
        12 111, 11 212, 12 221, 12 212, 11 112, 11 122, 12 121}] , #]
   & /@ Tuples[{0, 1}, 6]]
```



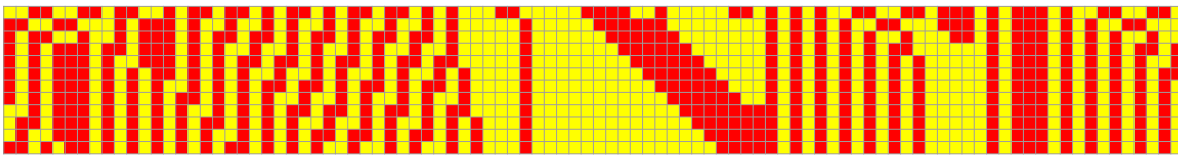
```
ArrayPlot[CellularAutomaton[
  ruleDCKV6[{11111, 11121, 12222, 12122, 11211, 12112}],
  {{1}, 0}, 6],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → {400, 300}]
```



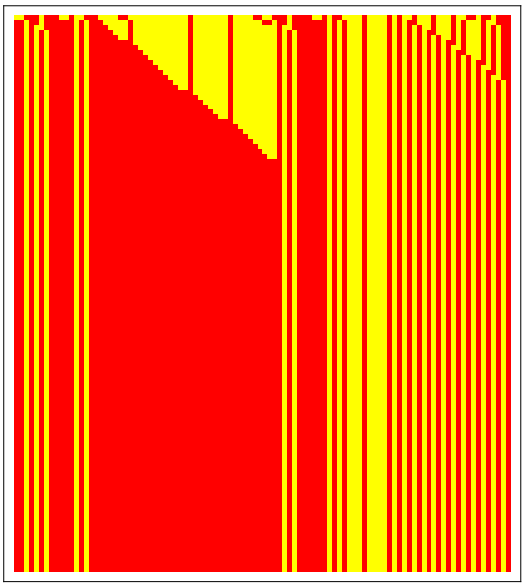
```
GraphPlot[# -> CellularAutomaton[
     ruleDCKV6[{11111, 11121, 12222, 12122, 11211, 12112}] , #]
   & /@ Tuples[{0, 1}, 4]]
```



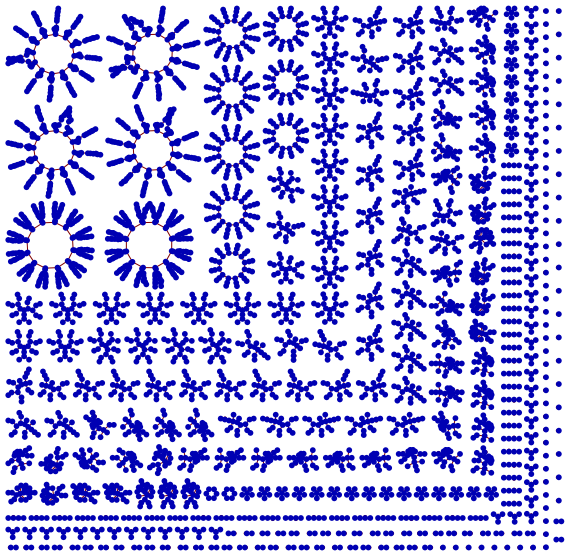
```
ArrayPlot[
 CellularAutomaton[ruleDCKV12[{11111, 11121, 11122, 11221, 12211, 12212, 11112,
    12111, 11212, 12121, 12212, 12222}], RandomInteger[1, 100], {11, All}],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 100]
```



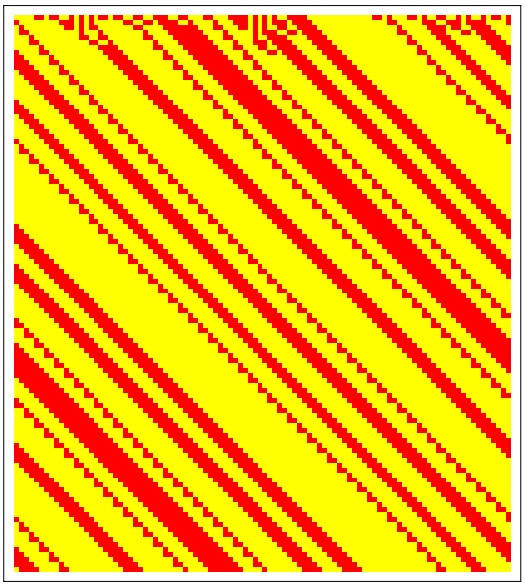
```
ArrayPlot[
 CellularAutomaton[ruleDCKV12[{11111, 11121, 11122, 11221, 12211, 12212, 11112,
    12111, 11212, 12121, 12212, 12222}], RandomInteger[1, 100], {111, All}],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green},
 Mesh → False, ImageSize → {400, 300}]
```



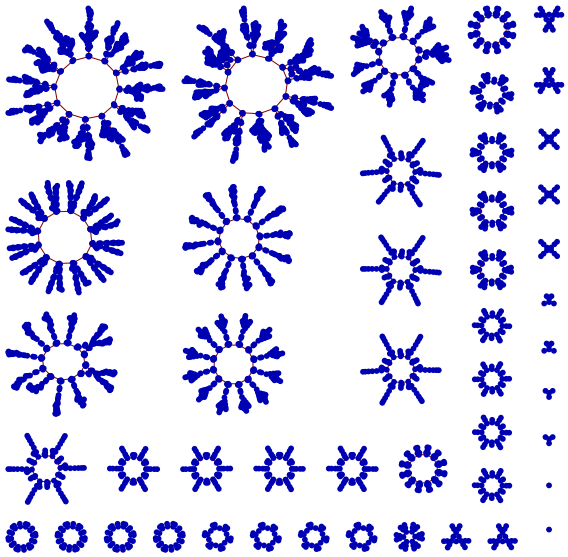
```
GraphPlot[
 # -> CellularAutomaton[ruleDCKV12[{11111, 11121, 11122, 11221, 12211, 12212,
       11112, 12111, 11212, 12121, 12212, 12222}], #] & /@ Tuples[{0, 1}, 12]]
```



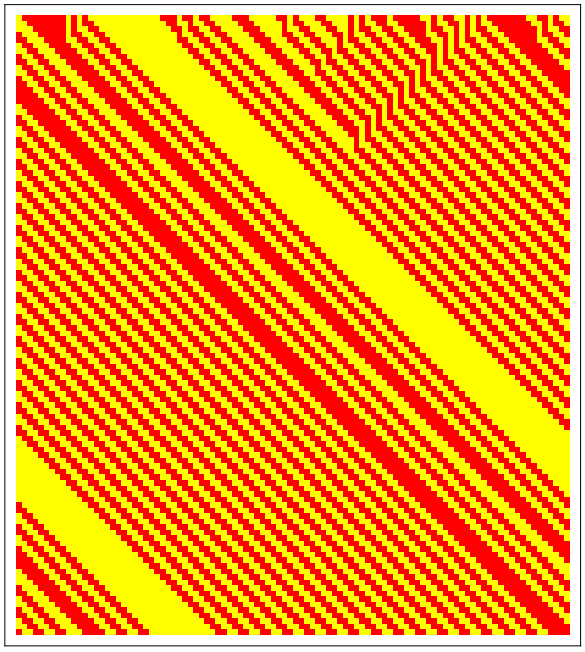
```
ArrayPlot[
 CellularAutomaton[ruleDCKV12[{11 111, 11 121, 11 122, 11 221, 12 211, 12 212, 11 112,
    12 112, 11 212, 12 121, 12 212, 12 222}], RandomInteger[1, 100], {111, All}],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green},
 Mesh → False, ImageSize → 300]
```



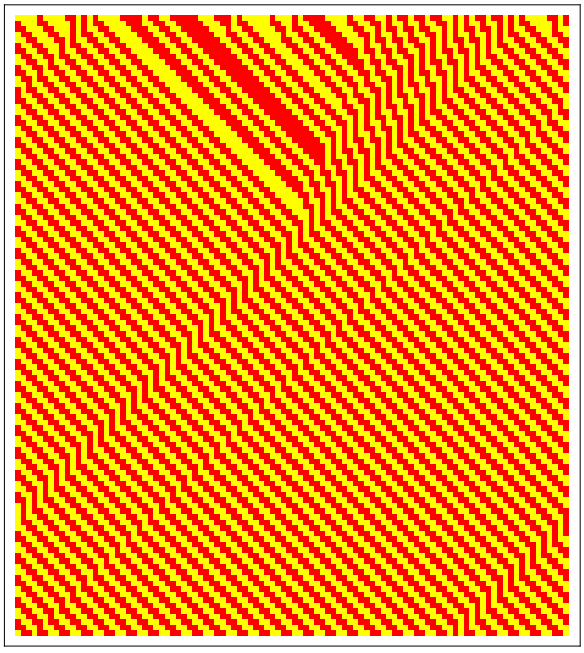
```
GraphPlot[
 # -> CellularAutomaton[ruleDCKV12[{11 111, 11 121, 11 122, 11 221, 12 211, 12 212,
       11 112, 12 112, 11 212, 12 121, 12 212, 12 222}], #] & /@ Tuples[{0, 1}, 12]]
```



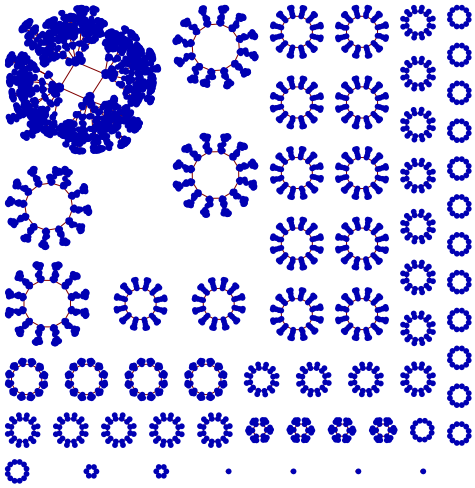
```
ArrayPlot[
 CellularAutomaton[ruleDCKV12[{11 111, 11 121, 11 122, 11 221, 12 212, 12 212, 11 112,
    12 112, 11 212, 12 121, 12 212, 12 222}], RandomInteger[1, 100], {111, All}],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green},
 Mesh → False, ImageSize → 300]
```

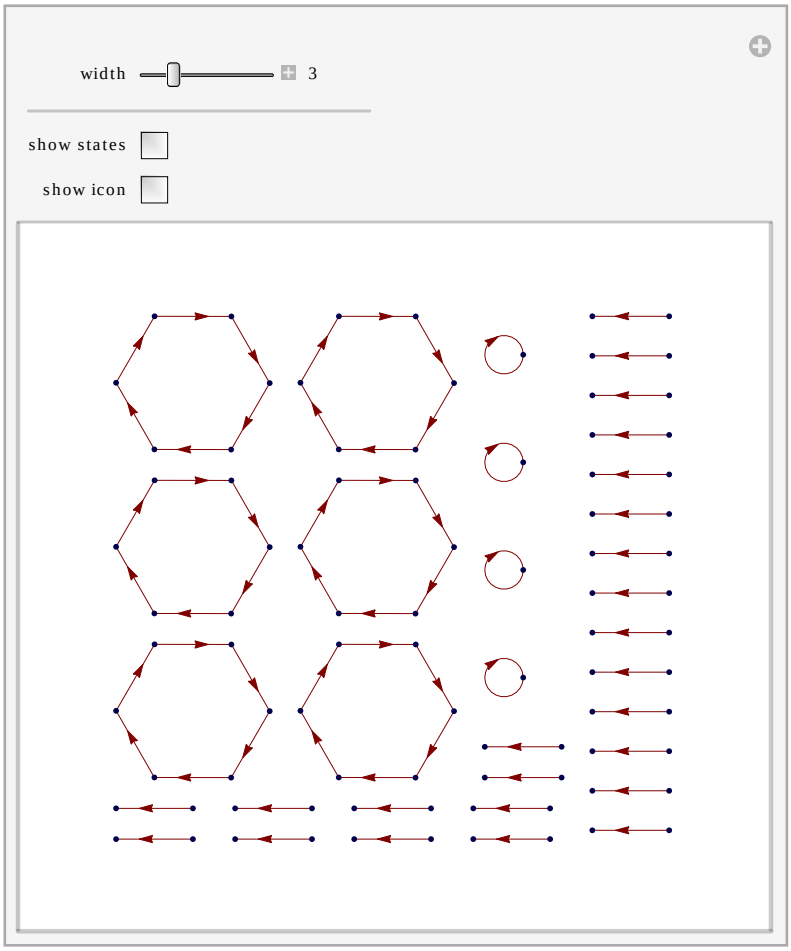


```
ArrayPlot[
 CellularAutomaton[ruleDCKV12[{11 111, 11 121, 11 122, 11 221, 12 212, 12 112, 11 112,
    12 112, 11 212, 12 121, 12 212, 12 222}], RandomInteger[1, 100], {111, All}],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green},
 Mesh → False, ImageSize → 300]
```

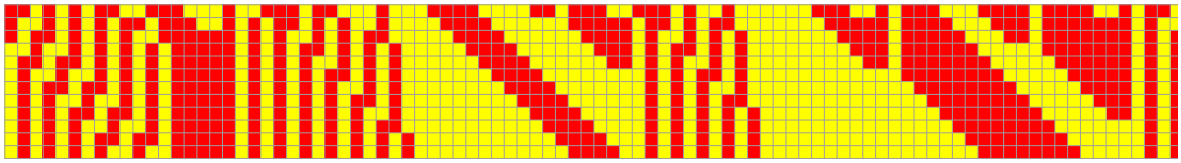


```
GraphPlot[
 # -> CellularAutomaton[ruleDCKV12[{11 111, 11 121, 11 122, 11 221, 12 212, 12 112,
       11 112, 12 112, 11 212, 12 121, 12 212, 12 222}], #] & /@ Tuples[{0, 1}, 12]]
```

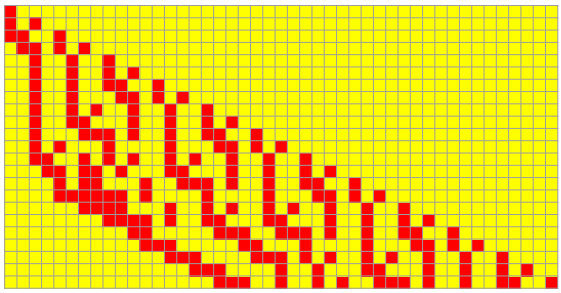




```
ArrayPlot[
 CellularAutomaton[ruleDCKV12[{11 111, 11 121, 11 122, 11 221, 12 211, 12 212, 11 112,
    12 111, 11 212, 12 121, 12 212, 12 222}], RandomInteger[1, 100], {11, All}],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 100]
```

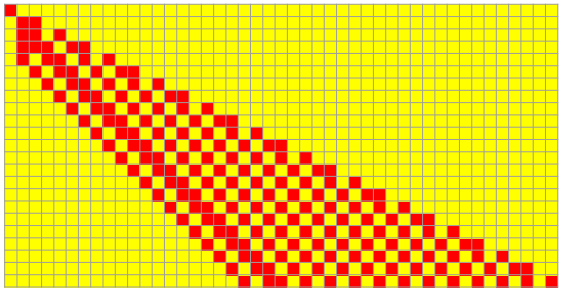


```
ArrayPlot[
 CellularAutomaton[ruleDCKV14[{11 111, 11 221, 12 212, 12 111, 11 212, 12 122, 12 121,
    12 221, 11 222, 11 121, 12 122, 12 221, 11 222, 12 212}] , {{1}, 0}, 22],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 300]
```

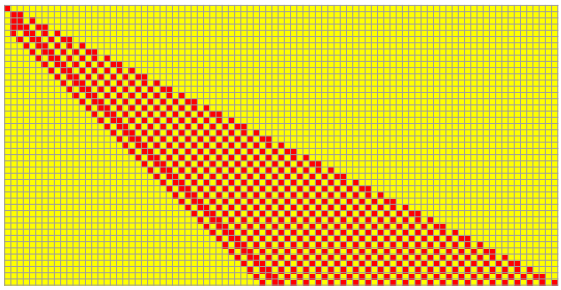


```
ArrayPlot[CellularAutomaton[
  ruleDCKV14[{11 111, 11 221, 12 212, 12 111, 11 212, 12 122, 12 121, 12 221, 11 222,
    11 121, 12 122, 12 221, 11 222, 12 212}], RandomInteger[{1, 100}], 22],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 300]
```

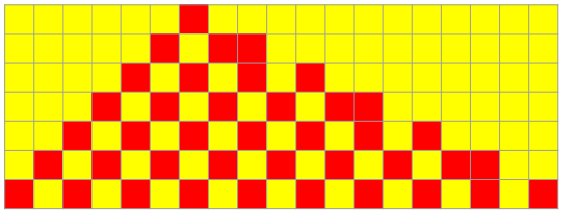
```
ArrayPlot[
 CellularAutomaton[ruleDCKV16[{11 111, 11 211, 12 212, 12 112, 11 212, 12 122, 12 121,
    12 221, 11 222, 11 121, 12 222, 12 221, 11 112, 12 211, 12 212, 12 221}],
  {{1}, 0}, 22],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 300]
```



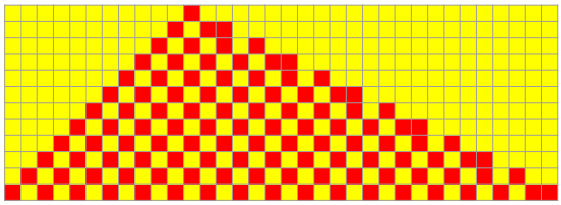
```
ArrayPlot[
 CellularAutomaton[ruleDCKV16[{11 111, 11 211, 12 212, 12 112, 11 212, 12 122, 12 121,
    12 221, 11 222, 11 121, 12 222, 12 221, 11 112, 12 211, 12 212, 12 221}],
  {{1}, 0}, 44],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 300]
```



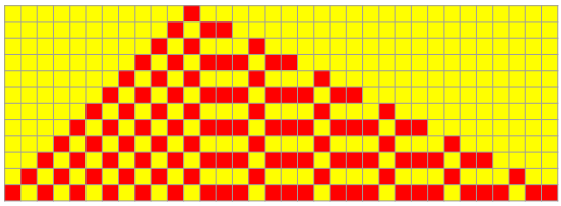
```
ArrayPlot[
 CellularAutomaton[ruleDCKV16[{11 111, 11 211, 12 212, 12 112, 11 212, 12 122, 12 121,
    12 221, 11 222, 11 122, 12 222, 12 221, 11 112, 12 211, 12 112, 12 221}],
  {{1}, 0}, 6],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 300]
```



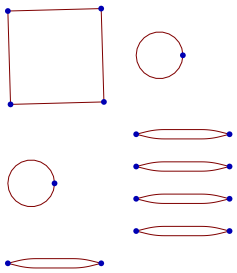
```
ArrayPlot[CellularAutomaton[
  ruleDCKV16[{11 111, 11 211, 12 212, 12 112, 11 212, 12 122, 12 121, 12 221, 11 222,
    11 122, 12 222, 12 221, 11 112, 12 211, 12 112, 12 221}] , {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 300]
```



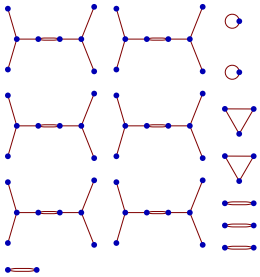
```
ArrayPlot[CellularAutomaton[ruleDCKV10[{11 211, 12 112, 11 111,
    12 221, 12 211, 11 222, 12 122, 11 222, 12 122, 11 122}], {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 300]
```



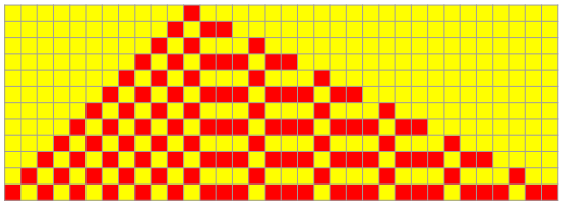
```
GraphPlot[# -> CellularAutomaton[
      ruleDCKV10[{11211, 12112, 11111,
        12221, 12211, 11222, 12122, 11222, 12122, 11122}] , #]
   & /@ Tuples[{0, 1}, 4]]
```



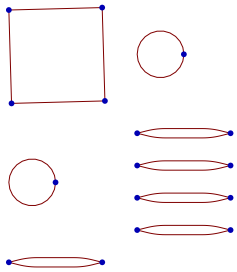
```
GraphPlot[# -> CellularAutomaton[
      ruleDCKV10[{11211, 12112, 11111,
        12221, 12211, 11222, 12122, 11222, 12122, 11122}] , #]
   & /@ Tuples[{0, 1}, 6]]
```



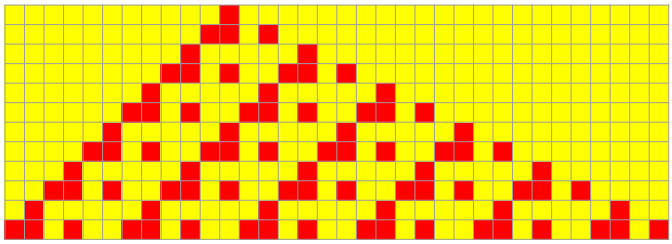
```
ArrayPlot[CellularAutomaton[
  ruleDCKV10[
   {11211, 12112, 11111, 12221, 12211, 11222, 12122, 11222, 12122, 11122}],
  {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 300]
```



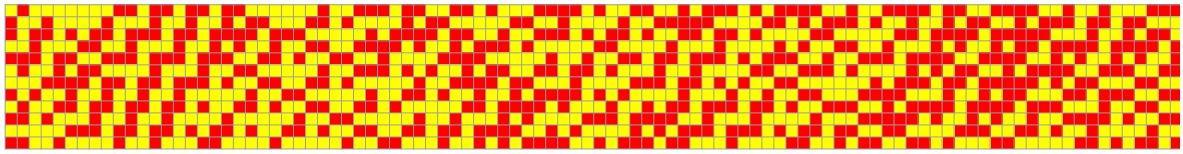
```
GraphPlot[# -> CellularAutomaton[
      ruleDCKV10[{11 211, 12 112, 11 111,
        12 221, 12 211, 11 222, 12 122, 11 222, 12 122, 11 122}] , #]
   & /@ Tuples[{0, 1}, 4]]
```



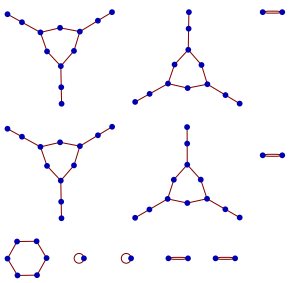
```
ArrayPlot[CellularAutomaton[
  ruleDCKV8[{11 111, 11 122, 11 212, 12 111, 11 221, 12 122, 12 211, 12 221}],
  {{1}, 0}, {11}],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True]
```



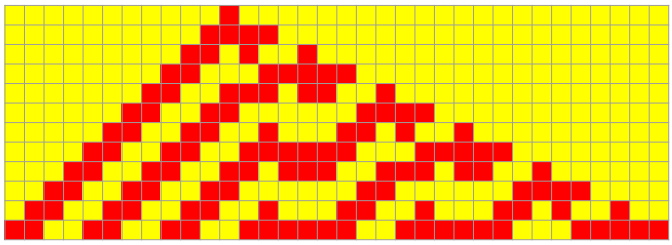
```
ArrayPlot[CellularAutomaton[
  ruleDCKV8[{11 111, 11 122, 11 212, 12 111, 11 221, 12 122, 12 211, 12 221}],
  RandomInteger[1, 100], {11}], ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True]
```



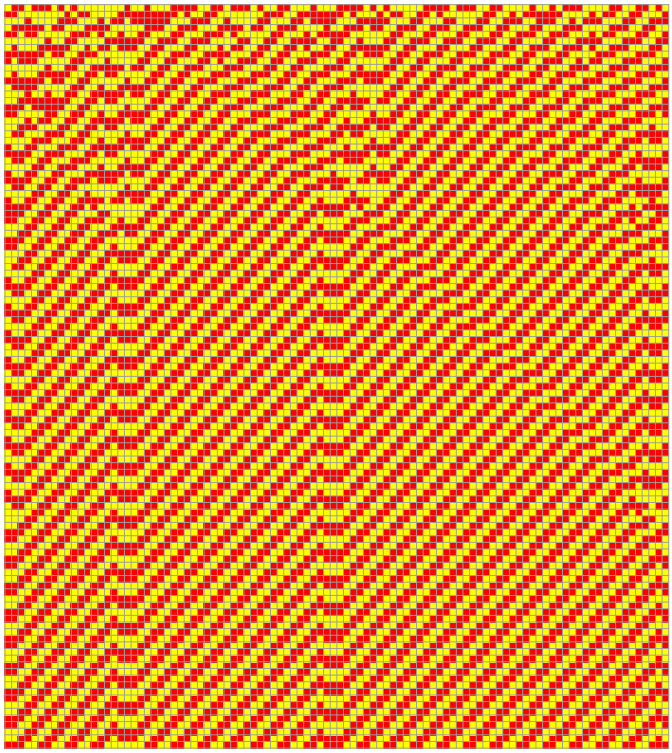
```
GraphPlot[# -> CellularAutomaton[
      ruleDCKV8[{11 111, 11 122, 11 212, 12 111, 11 221, 12 122, 12 211, 12 221}], #]
   & /@ Tuples[{0, 1}, 6]]
```



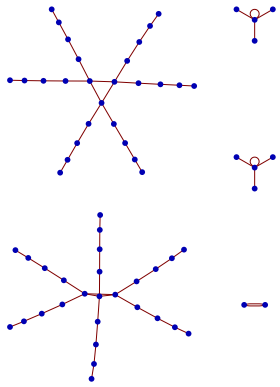
```
ArrayPlot[CellularAutomaton[
  ruleDCKV10[
   {11111, 11122, 11212, 11222, 11221, 12122, 12211, 12221, 12112, 12121}],
  {{1}, 0}, {11}], ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True]
```



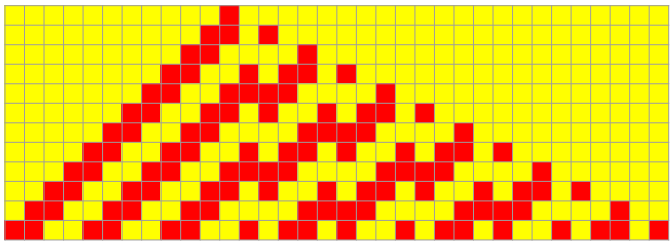
```
ArrayPlot[CellularAutomaton[
  ruleDCKV10[
   {11111, 11122, 11212, 11222, 11221, 12122, 12211, 12221, 12112, 12121}],
  RandomInteger[1, 100], {111}], ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True]
```



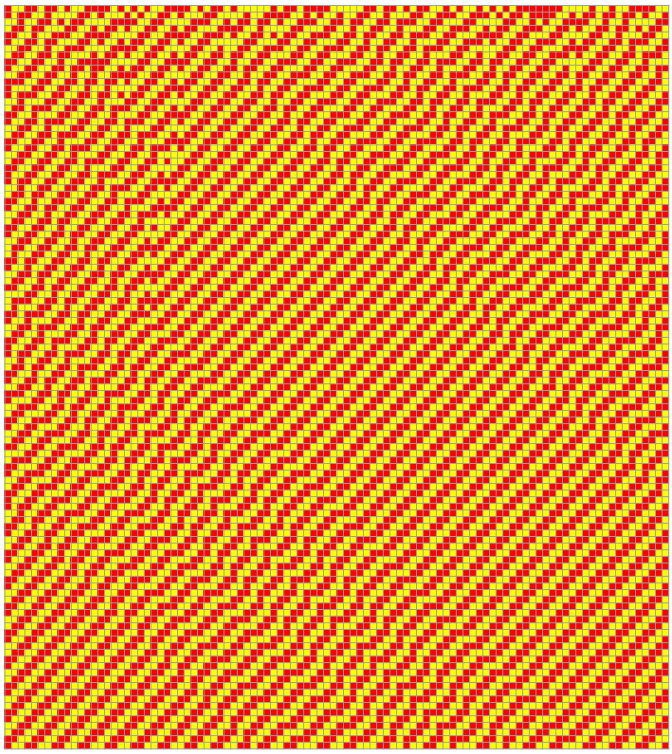
```
GraphPlot[# -> CellularAutomaton[
      ruleDCKV10[{11 111, 11 122, 11 212,
        11 222, 11 221, 12 112, 12 211, 12 221, 12 112, 12 122}], #]
   & /@ Tuples[{0, 1}, 6]]
```



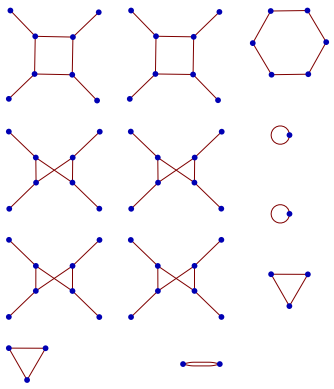
```
ArrayPlot[CellularAutomaton[
  ruleDCKV12[{11 111, 11 122, 11 212, 11 222,
    11 221, 12 221, 12 111, 12 211, 12 221, 12 122, 12 222, 11 211}],
  {{1}, 0}, {11}], ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True]
```



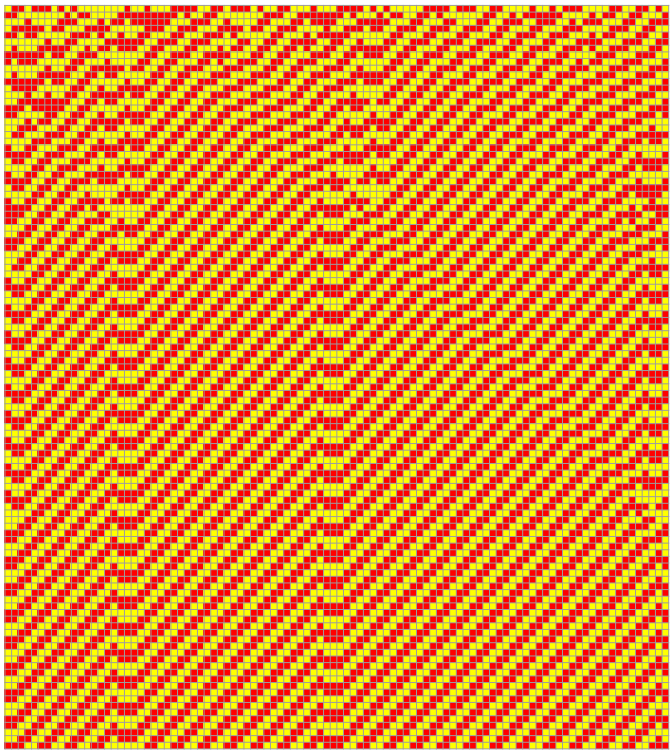
```
ArrayPlot[CellularAutomaton[
  ruleDCKV12[{11 111, 11 122, 11 212, 11 222,
    11 221, 12 221, 12 111, 12 211, 12 221, 12 122, 12 222, 11 211}],
  RandomInteger[1, 100], {111}], ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True]
```



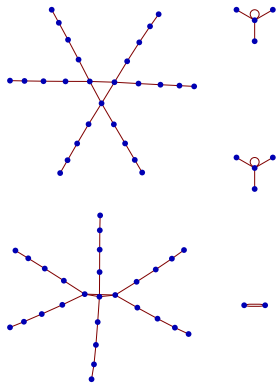
```
GraphPlot[
 # -> CellularAutomaton[ruleDCKV12[{11 111, 11 122, 11 212, 11 222, 11 221, 12 221,
       12 111, 12 211, 12 221, 12 122, 12 222, 11 211}], #] & /@ Tuples[{0, 1}, 6]]
```



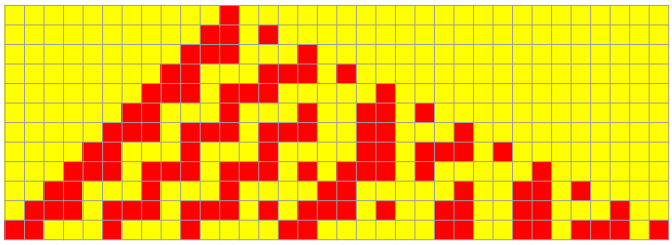
```
ArrayPlot[CellularAutomaton[
  ruleDCKV10[
   {11 111, 11 122, 11 212, 11 222, 11 221, 12 122, 12 211, 12 221, 12 112, 12 121}],
  RandomInteger[1, 100], {111}], ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True]
```



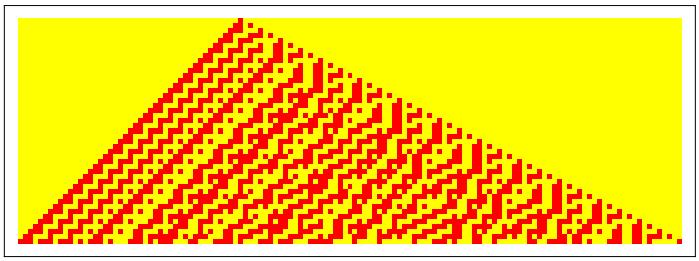
```
GraphPlot[
 # -> CellularAutomaton[ruleDCKV10[{11 111, 11 122, 11 212, 11 222, 11 221, 12 112,
       12 211, 12 221, 12 112, 12 122}], #]
   & /@ Tuples[{0, 1}, 6]]
```



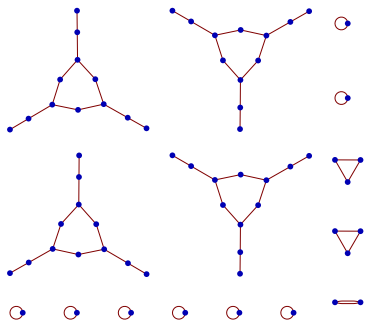
```
ArrayPlot[CellularAutomaton[
  ruleDCKV12[{11111, 11122, 11212, 11222,
    11221, 12221, 12111, 12212, 12221, 12122, 12222, 11211}],
  {{1}, 0}, {11}], ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True]
```



```
ArrayPlot[CellularAutomaton[
  ruleDCKV12[{11111, 11122, 11212, 11222,
    11221, 12221, 12111, 12212, 12221, 12122, 12222, 11211}],
  {{1}, 0}, {44}], ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → False]
```

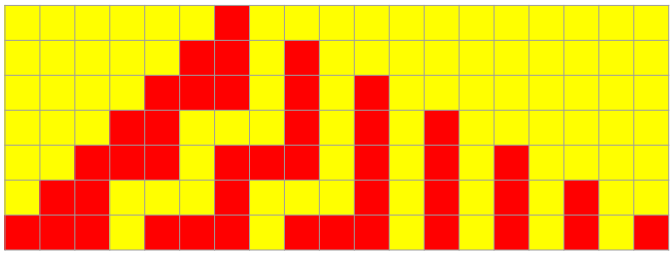


```
GraphPlot[# -> CellularAutomaton[
     ruleDCKV12[{11111, 11122, 11212, 11222, 11221,
       12221, 12111, 12212, 12221, 12122, 12222, 11211}], #]
   & /@ Tuples[{0, 1}, 6]]
```

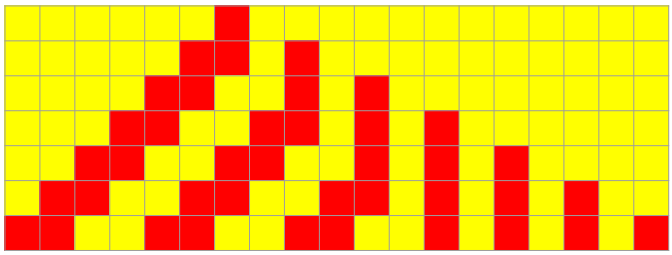


```
ArrayPlot[CellularAutomaton[
  ruleDCKV12[{11111, 11122, 11112, 11222,
    11221, 12221, 12111, 12212, 12221, 12121, 12222, 11211}],
  {{1}, 0}, {22}], ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True]
```

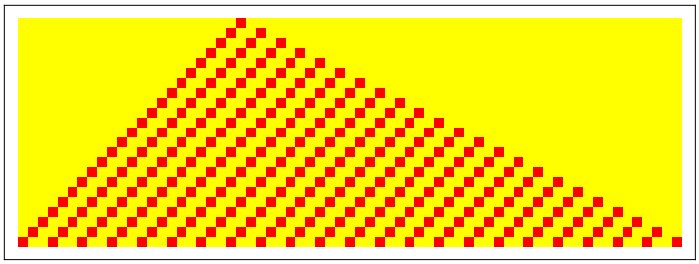
```
ArrayPlot[CellularAutomaton[
  ruleDCKV12[{11111, 11122, 11212, 11222,
    11221, 12221, 12111, 12212, 12221, 12121, 12222, 11211}],
  {{1}, 0}, {6}], ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True]
```



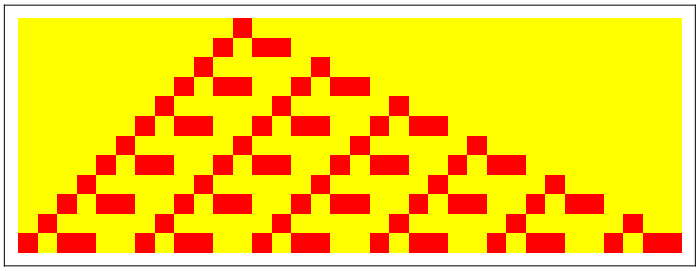
```
ArrayPlot[CellularAutomaton[
  ruleDCKV12[{11111, 11122, 11212, 11222,
    11212, 12221, 12111, 12211, 12221, 12121, 12222, 11211}],
  {{1}, 0}, {6}], ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True]
```



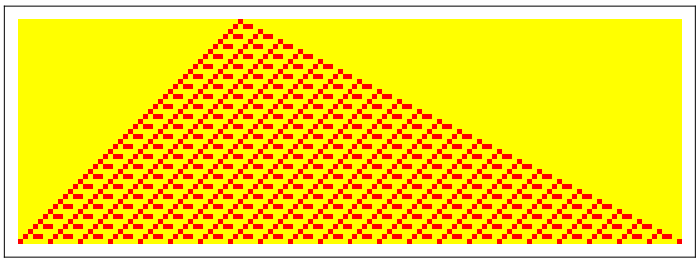
```
ArrayPlot[CellularAutomaton[
  ruleDCKV12[{11111, 11122, 11112, 11222,
    11221, 12221, 12111, 12211, 12221, 12121, 12222, 11211}],
  {{1}, 0}, {22}], ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → False]
```



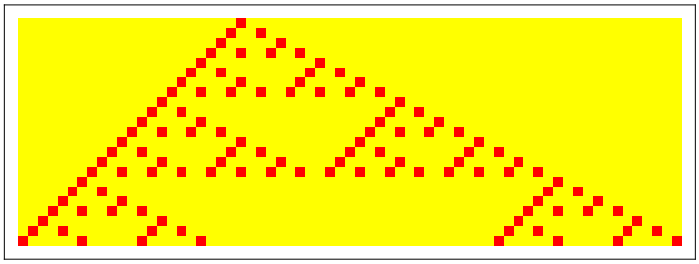
```
ArrayPlot[CellularAutomaton[
  ruleDCKV8[{11111, 11122, 11211, 12112, 11222, 12121, 12211, 12221}],
  {{1}, 0}, {11}], ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → False]
```

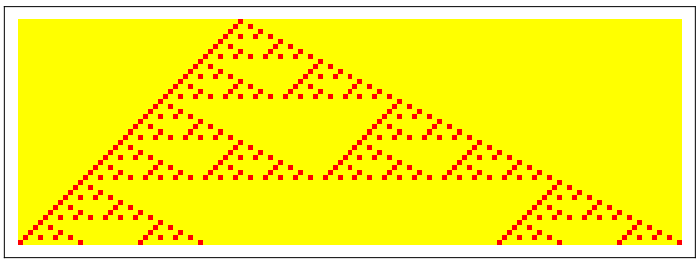


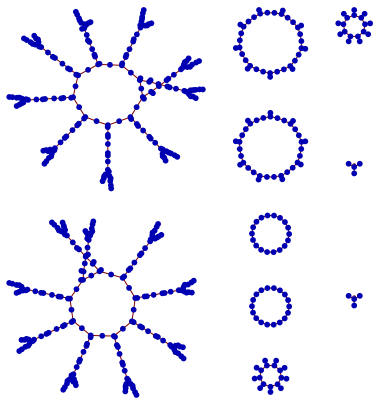
```
ArrayPlot[CellularAutomaton[
  ruleDCKV8[{11111, 11122, 11211, 12112, 11222, 12121, 12211, 12221}],
  {{1}, 0}, {44}], ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → False]
```



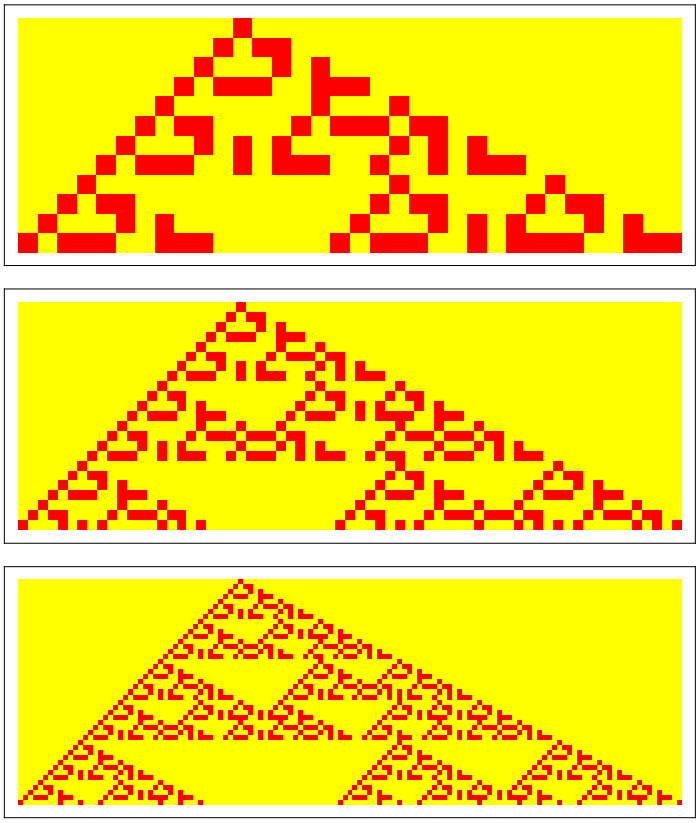
```
ArrayPlot[CellularAutomaton[
  ruleDCKV12[{11111, 11122, 11112, 11222,
    11221, 12221, 12111, 12212, 12221, 12121, 12222, 11211}],
  {{1}, 0}, {22}], ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → False]
```



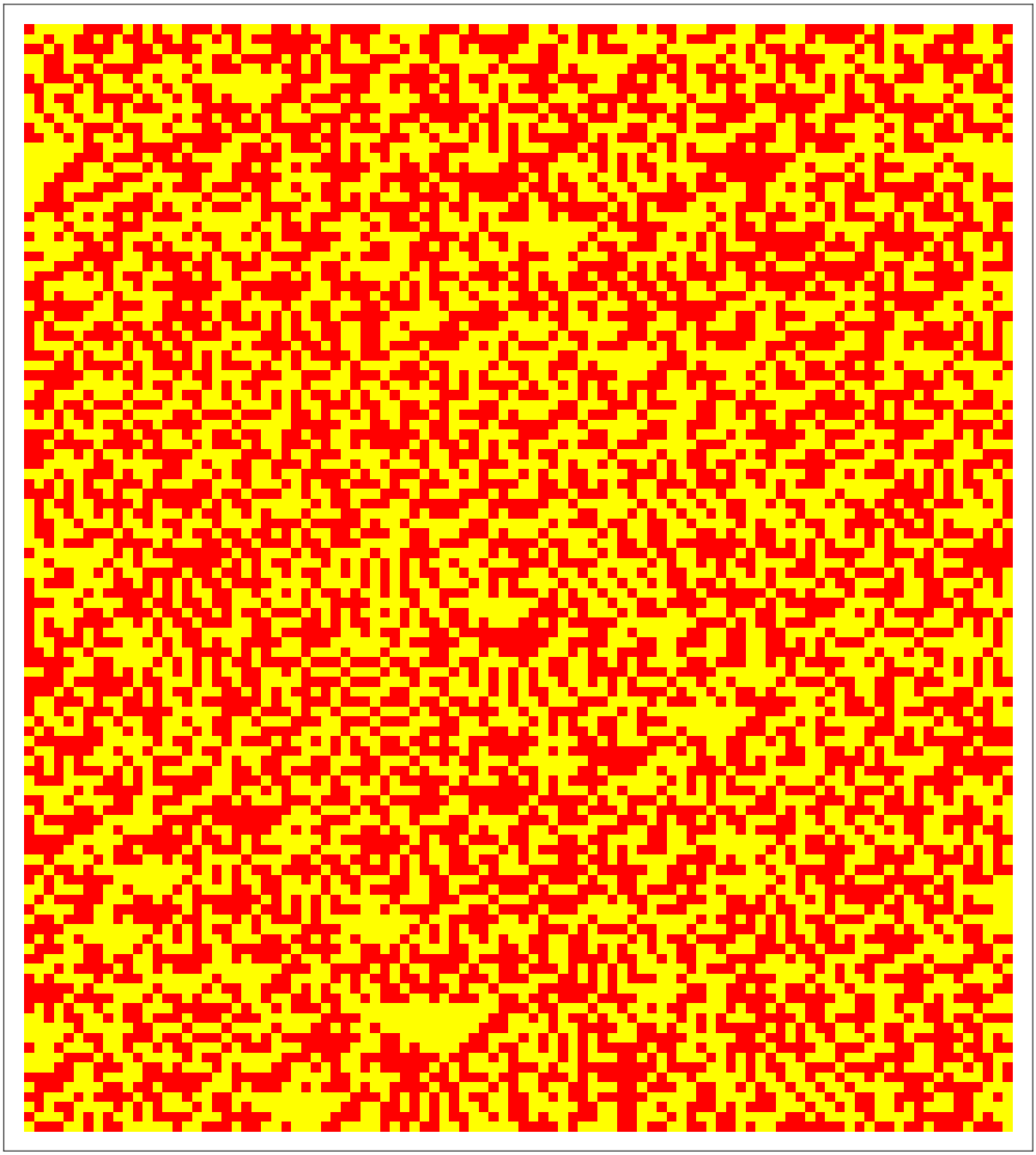


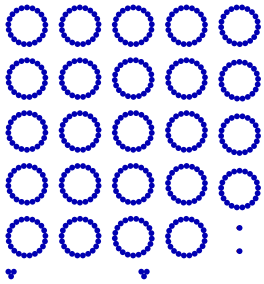


```
ArrayPlot[CellularAutomaton[
  ruleDCKV12[
   {
    11111, 11122,
    11112, 11222,
    11221, 12221,
    12112, 12212,
    12221, 12121,
    12222, 11211
   }],
  {{1}, 0}, {11}], ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → False]
```

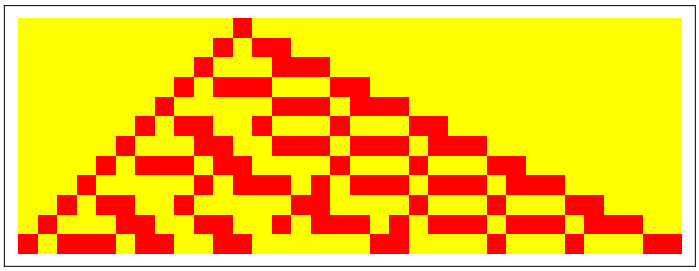


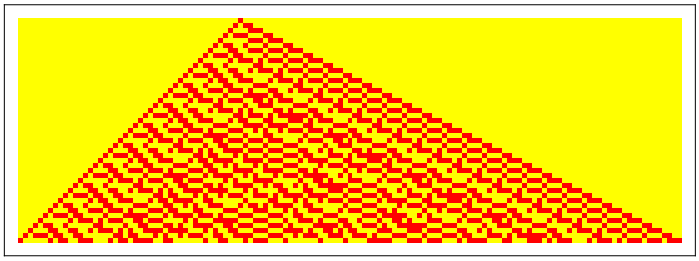
```
ArrayPlot[CellularAutomaton[
  ruleDCKV12[
   {
    11111, 11122,
    11112, 11222,
    11221, 12221,
    12112, 12212,
    12221, 12121,
    12222, 11211
   }],
  RandomInteger[1, 100], {111}],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → False]
```



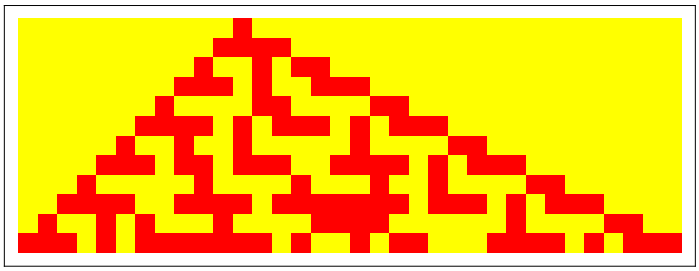


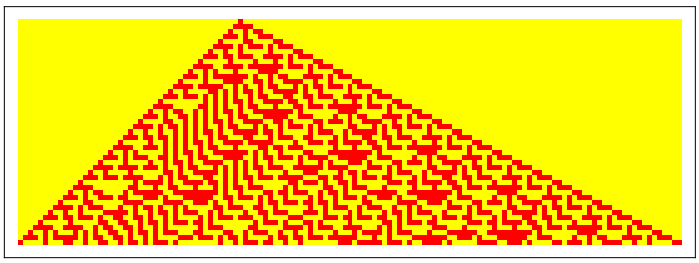
```
ArrayPlot[CellularAutomaton[
  ruleDCKV12[{11 111, 11 122, 11 112, 11 221,
    11 221, 12 221, 12 112, 12 212, 12 221, 12 121, 12 222, 11 211}],
  {{1}, 0}, {11}], ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → False]
```



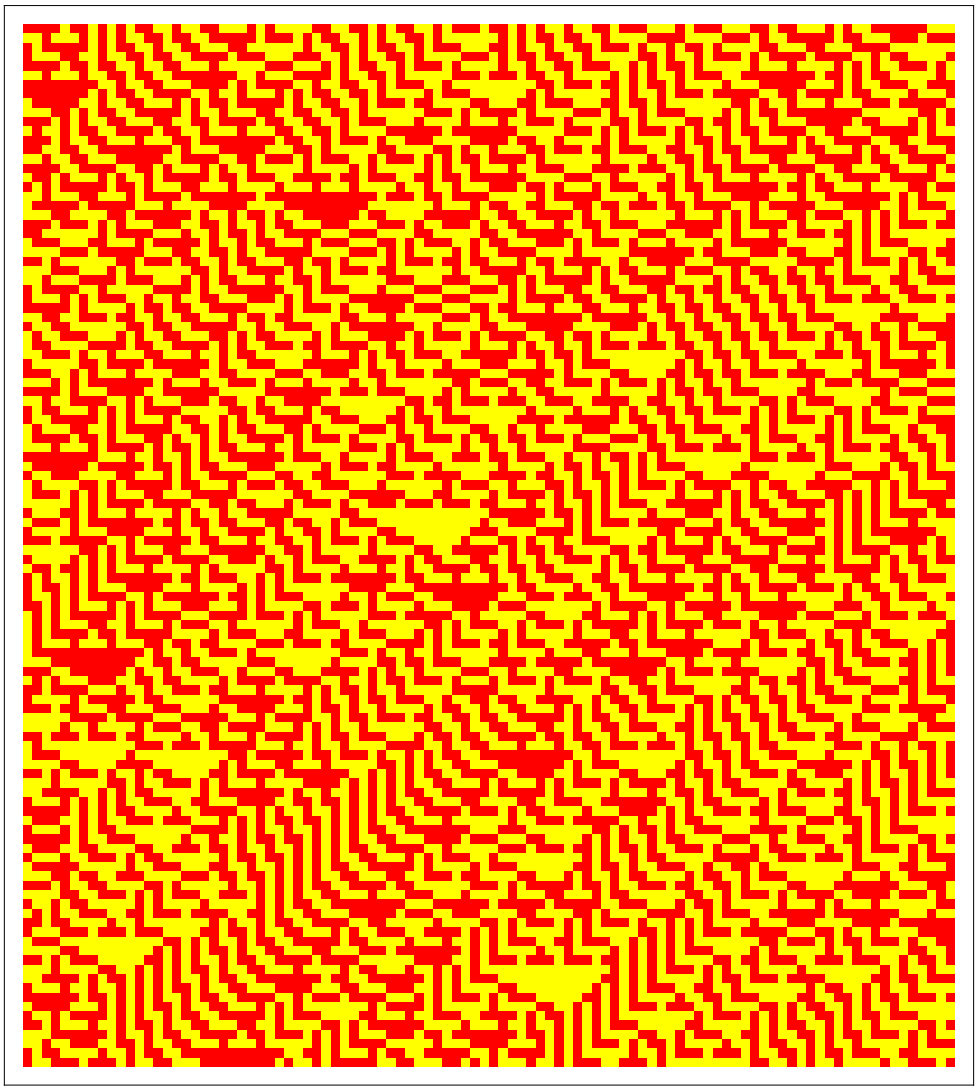


```
ArrayPlot[CellularAutomaton[
  ruleDCKV10[
   {11 111, 11 122, 11 212, 11 221, 12 112, 12 212, 12 221, 12 121, 12 222, 11 211}],
  {{1}, 0}, {11}], ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → False]
```

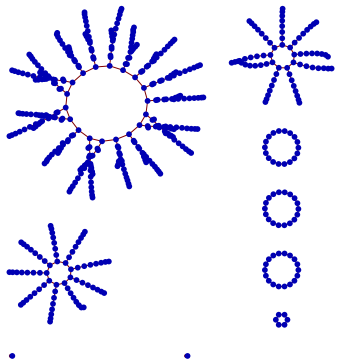




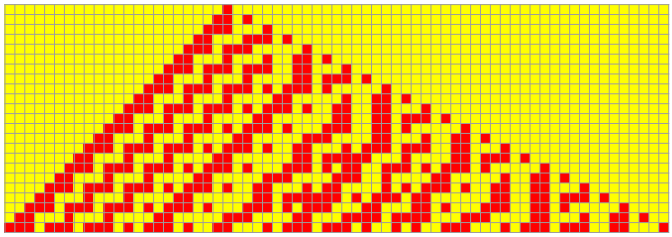
```
ArrayPlot[CellularAutomaton[
  ruleDCKV10[
   {11 111, 11 122, 11 212, 11 221, 12 112, 12 212, 12 221, 12 121, 12 222, 11 211}],
  RandomInteger[1, 100], {111}],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → False]
```



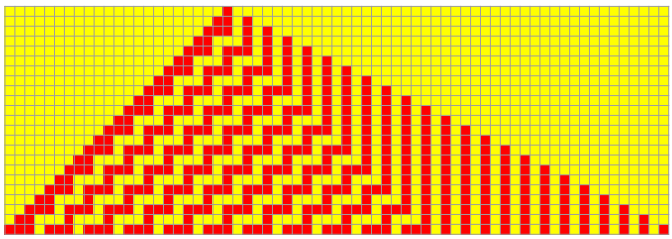
```
GraphPlot[# -> CellularAutomaton[
      ruleDCKV10[{11111, 11122, 11212,
        11221, 12112, 12212, 12221, 12121, 12222, 11211}], #]
   & /@ Tuples[{0, 1}, 9]]
```

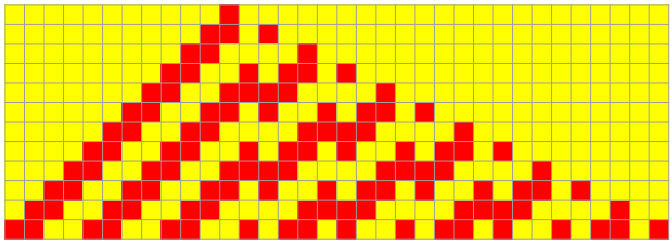


```
ArrayPlot[CellularAutomaton[
  ruleDCKV12[{11111, 11122, 11112, 11222,
    11221, 12221, 12111, 12212, 12221, 12122, 12222, 11212}],
  {{1}, 0}, {22}], ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True]
```

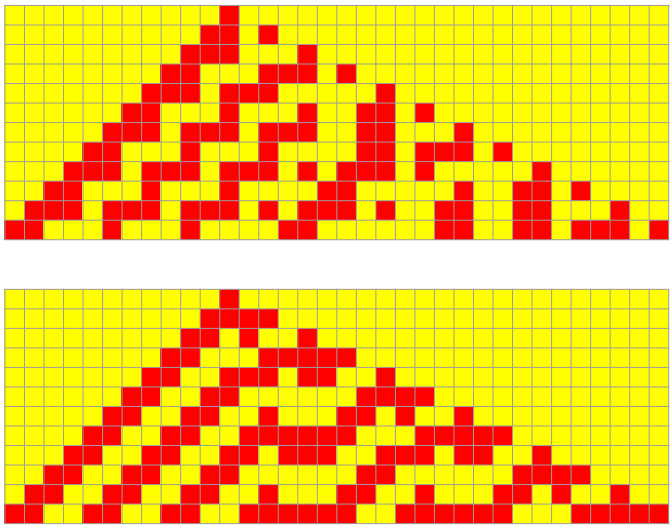


```
ArrayPlot[CellularAutomaton[
  ruleDCKV12[{11111, 11122, 11112, 11222,
    11221, 12221, 12111, 12212, 12221, 12121, 12222, 11212}],
  {{1}, 0}, {22}], ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True]
```

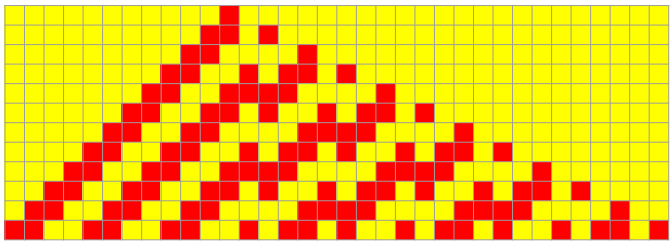




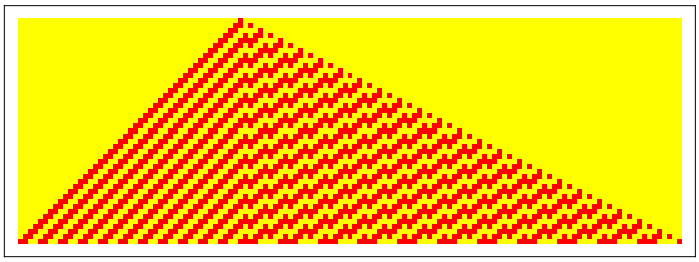
```
ArrayPlot[CellularAutomaton[
  ruleDCKV12[{11111, 11122, 11212, 11222,
    11221, 12221, 12111, 12212, 12221, 12122, 12222, 11211}],
  {{1}, 0}, {11}], ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True]
```



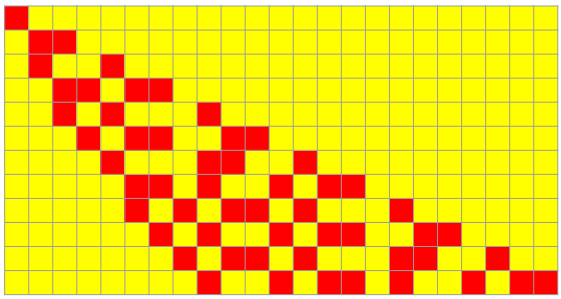
```
ArrayPlot[CellularAutomaton[
  ruleDCKV12[{11111, 11122, 11212, 11222,
    11221, 12221, 12111, 12211, 12221, 12122, 12222, 11211}],
  {{1}, 0}, {11}], ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True]
```



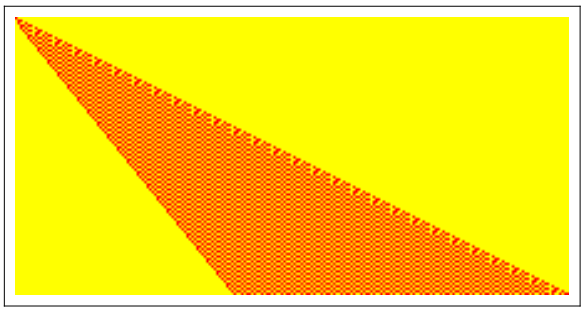
```
ArrayPlot[CellularAutomaton[
  ruleDCKV12[{11111, 11122, 11212, 11222,
    11221, 12221, 12111, 12211, 12221, 12122, 12222, 11211}],
  {{1}, 0}, {44}], ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → False]
```



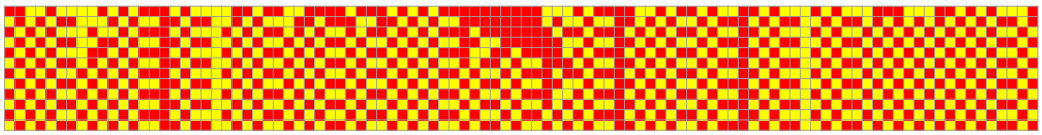
```
ArrayPlot[CellularAutomaton[ruleDCKV10[{11 211, 12 112, 11 111,
    12 221, 12 211, 11 222, 12 122, 11 222, 12 122, 11 121}], {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 300]
```



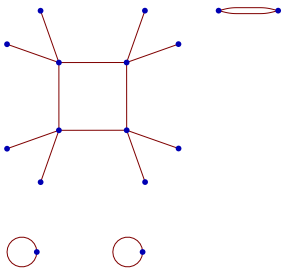
```
ArrayPlot[CellularAutomaton[ruleDCKV10[{11 211, 12 112, 11 111, 12 221,
    12 211, 11 222, 12 122, 11 222, 12 122, 11 121}] , {{1}, 0}, 111],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green, 4 → Pink},
 Mesh → False, ImageSize → 300]
```



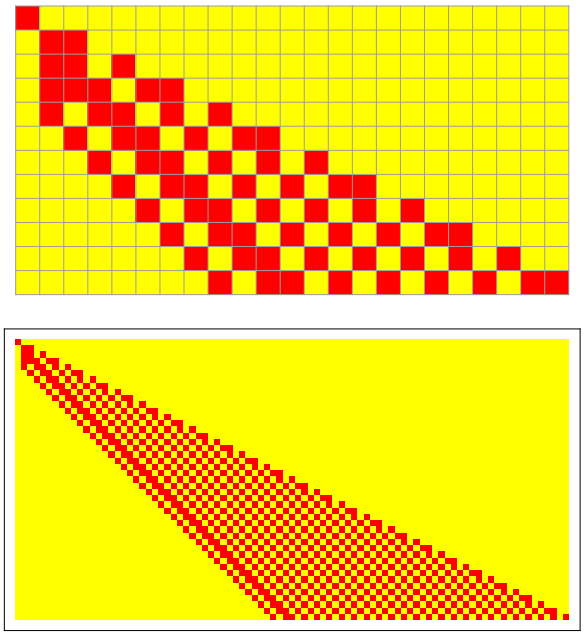
```
ArrayPlot[
 CellularAutomaton[ruleDCKV10[{11 211, 12 112, 11 111, 12 221, 12 211, 11 222,
    12 122, 11 222, 12 122, 11 121}] , RandomInteger[1, 100], 11],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 300]
```



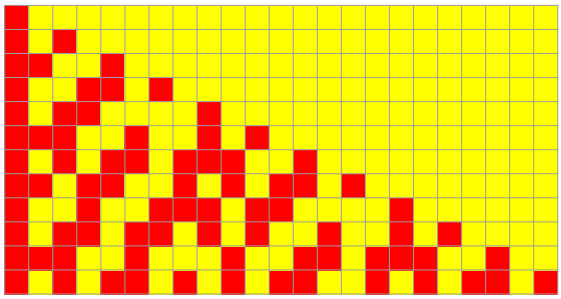
```
GraphPlot[
 # -> CellularAutomaton[ruleDCKV10[{11 211, 12 112, 11 111, 12 221, 12 211, 11 222,
       12 122, 11 222, 12 122, 11 121}] , #]
   & /@ Tuples[{0, 1}, 4]]
```

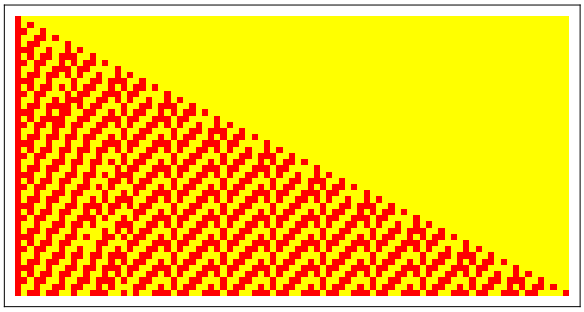


```
ArrayPlot[CellularAutomaton[ruleDCKV10[{11211, 12112, 11111,
    12221, 12212, 11222, 12122, 11221, 12122, 11121}], {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 300]
```

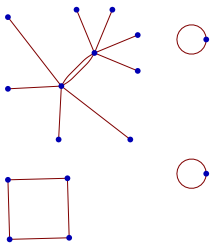


```
ArrayPlot[CellularAutomaton[ruleDCKV10[{11121, 12111, 11111,
    12221, 12211, 11222, 12122, 11212, 12122, 11122}] , {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green, 4 → Pink},
 Mesh → True, ImageSize → 300]
```

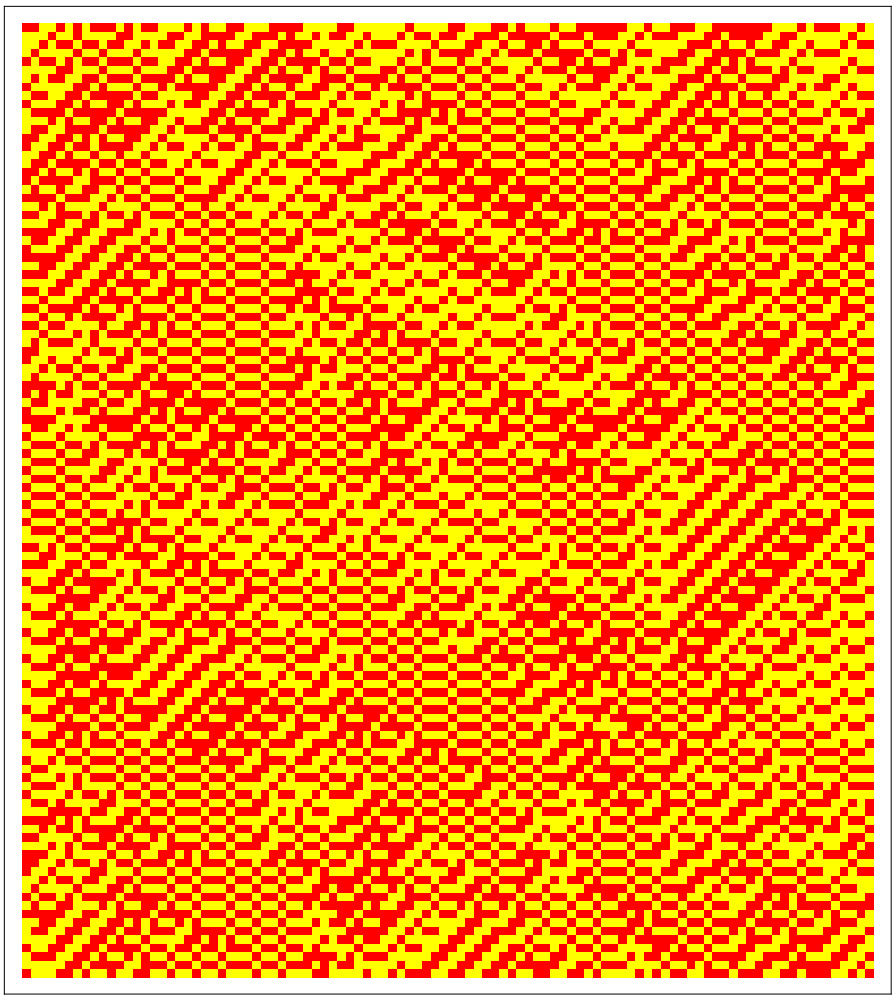




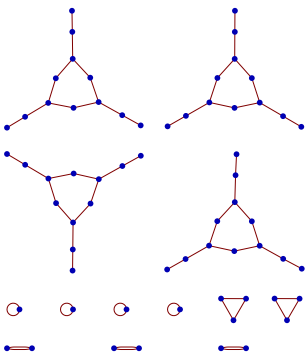
```
GraphPlot[
 # -> CellularAutomaton[ruleDCKV10[{11121, 12111, 11111, 12221, 12211, 11222,
       12122, 11212, 12122, 11122}] , #] & /@ Tuples[{0, 1}, 4]]
```



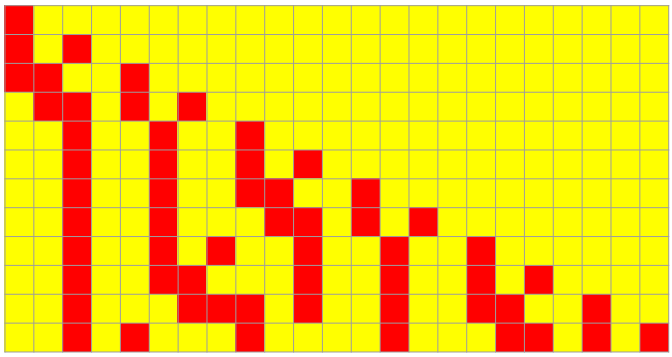
```
ArrayPlot[CellularAutomaton[
  ruleDCKV8[{11 111, 11 122, 11 211, 12 112, 11 222, 12 121, 12 211, 12 221}],
  RandomInteger[1, 100], 111], ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → False]
```

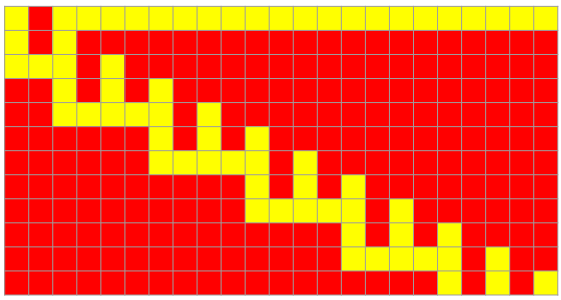


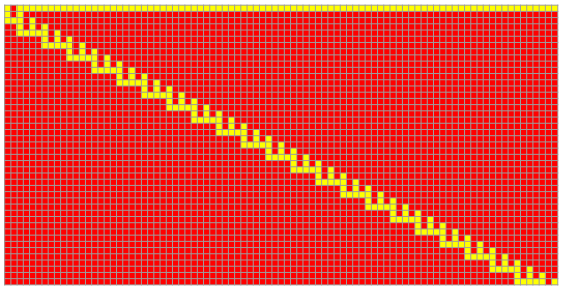
```
GraphPlot[# -> CellularAutomaton[
      ruleDCKV8[{11 111, 11 122, 11 211, 12 112, 11 222, 12 121, 12 211, 12 221}], #]
   & /@ Tuples[{0, 1}, 6]]
```

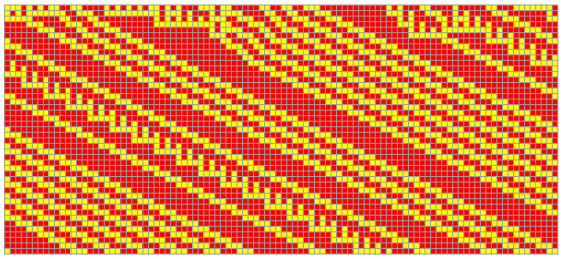


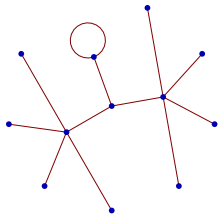
```
ArrayPlot[CellularAutomaton[
  ruleDCKV18[{11111, 11112, 11121, 11221, 12212, 12212, 12111, 12111,
    11212, 11211, 12122, 12121, 12221, 12221, 11121, 12122, 12122}],
  {{1}, 0}, 11], ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True]
```

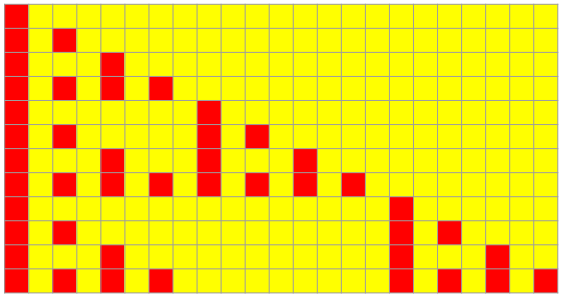


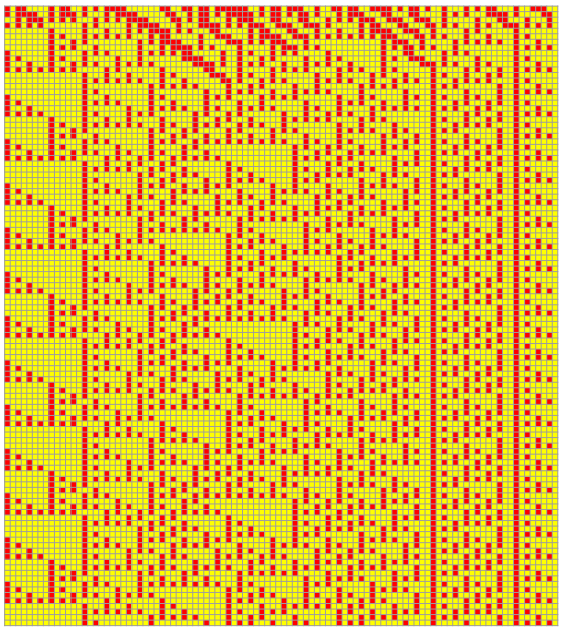




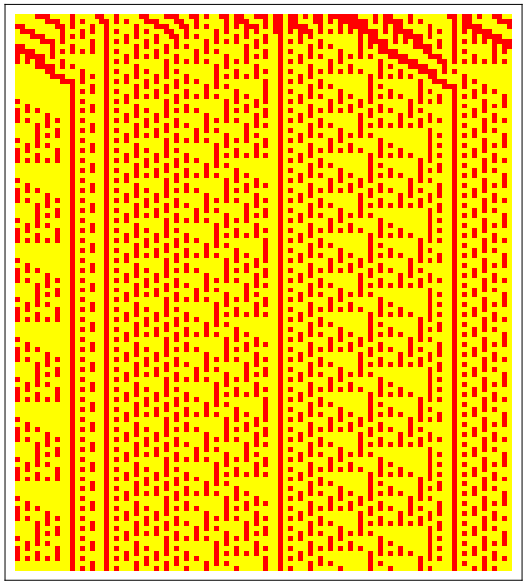


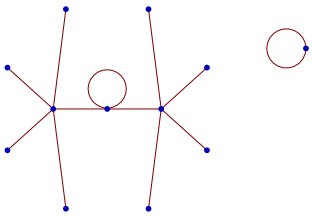


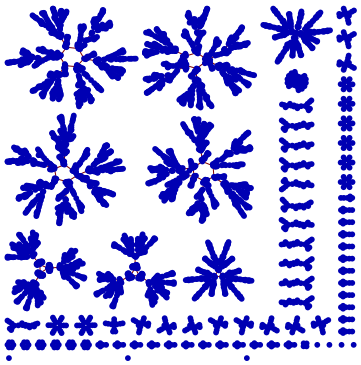




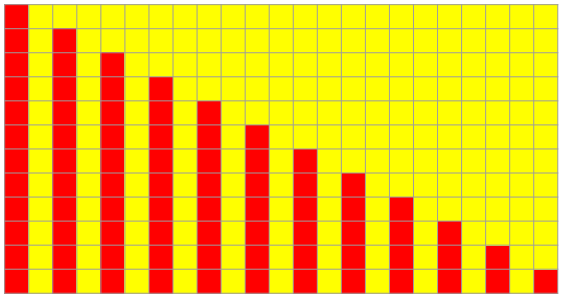
```
ArrayPlot[CellularAutomaton[
  {{1, 1, 1, 1} → 1,
   {0, 0, 0, 0} → 0,
   {0, 0, 0, 1} → 0, {1, 1, 0, 0} → 1, {0, 0, 1, 1} → 0, {1, 0, 0, 1} → 0,
   {0, 1, 1, 0} → 1, {1, 0, 1, 1} → 1, {0, 1, 0, 0} → 0, {1, 1, 0, 1} → 0,
   {0, 0, 1, 0} → 1, {1, 0, 1, 0} → 0, {0, 1, 0, 1} → 0, {1, 0, 0, 0} → 1,
   {0, 1, 1, 1} → 0, {1, 1, 1, 0} → 1, {1, 0, 1, 0} → 0, {0, 1, 0, 1} → 1},
  RandomInteger[1, 100], {111}],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → False, ImageSize → {400, 300}]
```



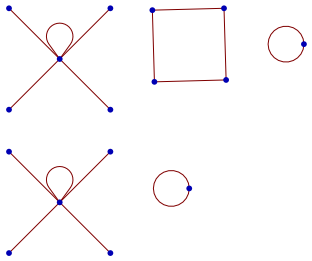




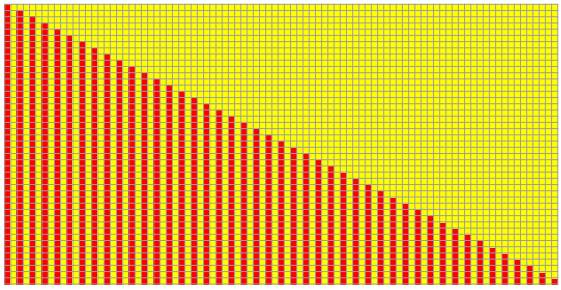
```
ArrayPlot[CellularAutomaton[
  ruleDCKV10[
   {11121, 12111, 11111, 12221, 12211, 11222, 12121, 11212, 12122, 11122}] ,
  {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 300]
```



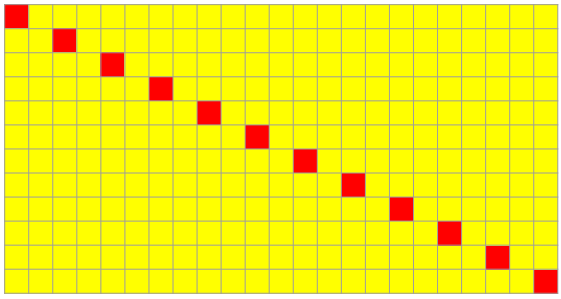
```
GraphPlot[# -> CellularAutomaton[
      ruleDCKV10[{11121, 12111, 11111,
        12221, 12211, 11222, 12121, 11212, 12122, 11122}] , #]
   & /@ Tuples[{0, 1}, 4]]
```



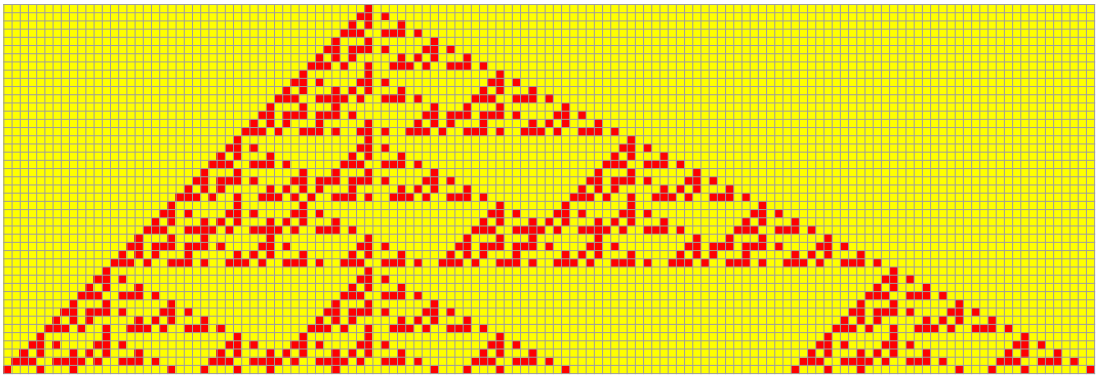
```
ArrayPlot[CellularAutomaton[
  ruleDCKV10[
   {11111, 11221, 11121, 12111,  11212, 12221, 12212, 11112, 11122, 12121}],
  {{1}, 0}, 44],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 300]
```



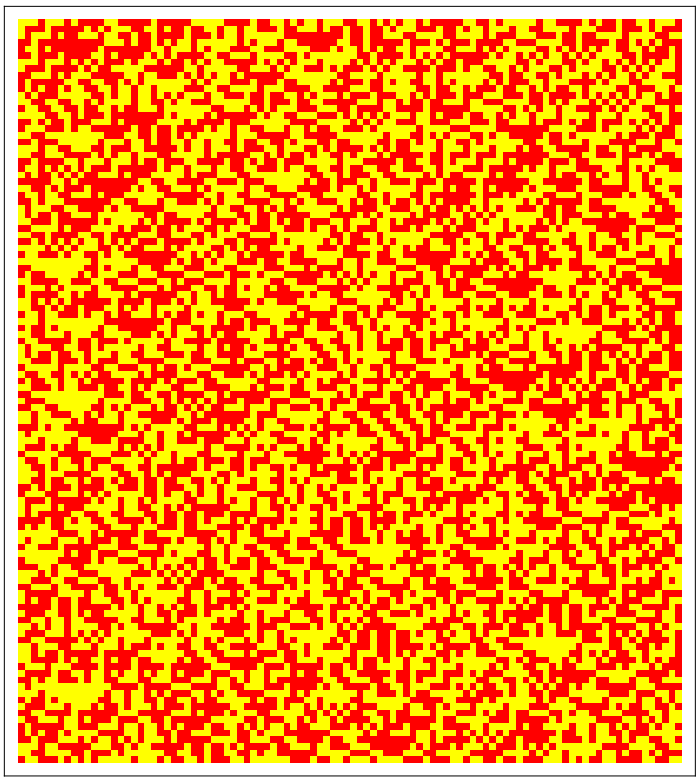
```
ArrayPlot[CellularAutomaton[
  ruleDCKV10[
   {11121, 12111, 11111, 12221, 12211, 11211, 12121, 11212, 12122, 11122}] ,
  {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 300]
```



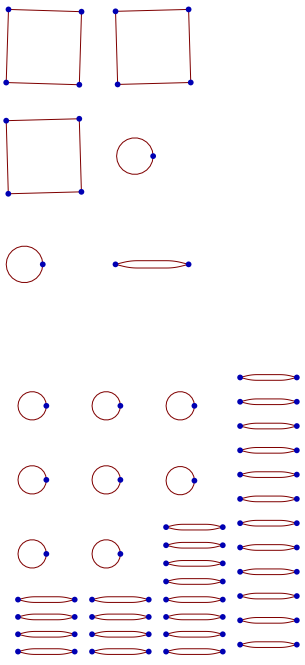
```
ArrayPlot[CellularAutomaton[
  ruleDCKV8[{11111, 11122, 11212, 12111, 11221, 12122, 12212, 12221}],
  {{1}, 0}, 44],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True]
```



```
ArrayPlot[CellularAutomaton[
  ruleDCKV8[{11 111, 11 122, 11 212, 12 111, 11 221, 12 122, 12 212, 12 221}],
  RandomInteger[1, 100], {111}],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → False]
```



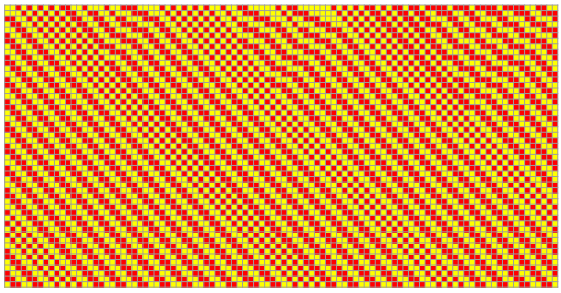
```
GraphPlot[# -> CellularAutomaton[
     ruleDCKV8[{11 111, 11 122, 11 212, 12 111, 11 221, 12 122, 12 212, 12 221}], #]
   & /@ Tuples[{0, 1}, 4]]
```



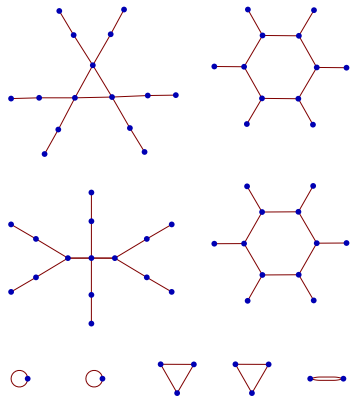
```
ArrayPlot[CellularAutomaton[
  ruleDCKV10[
   {11121, 12111, 11111, 12221, 12211, 11211, 12121, 11212, 12122, 11122}],
  RandomInteger[1, 100], 50],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 300]
```

```
$Aborted
```

```
ArrayPlot[
 CellularAutomaton[ruleDCKV16[{11111, 11211, 12212, 12112, 11212, 12122, 12333,
    12221, 11221, 11121, 12222, 12221, 11112, 12211, 12113, 12223}],
  RandomInteger[1, 100], 50],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 300]
```



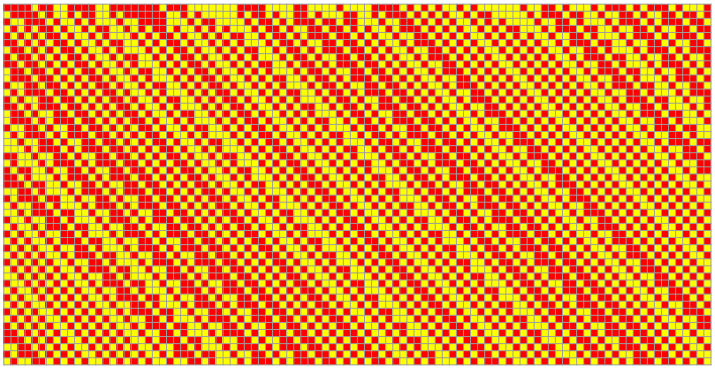
```
GraphPlot[# -> CellularAutomaton[
     ruleDCKV16[{11111, 11211, 12212, 12112, 11212, 12122, 12333, 12221,
       11221, 11121, 12222, 12221, 11112, 12211, 12113, 12223}] , #]
   & /@ Tuples[{0, 1}, 6]]
```



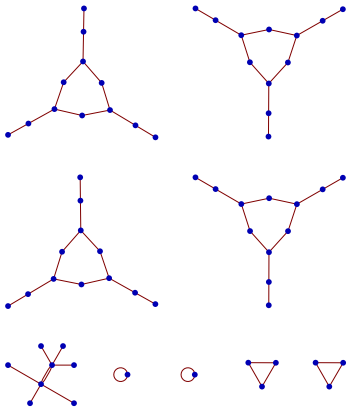
```
ruleDCKV16a :=
 ruleDCKV16[{11111, 11211, 12212, 12112, 11212, 12122, 12121,
   12221, 11222, 11121, 12222, 12221, 11112, 12211, 12212, 12221}]
```

```
ruleDCKV16a
```

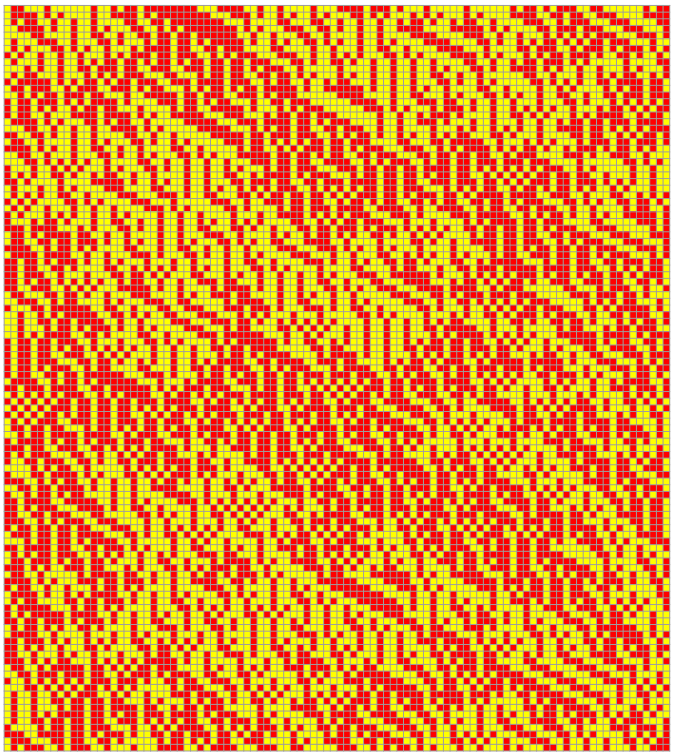
```
ArrayPlot[CellularAutomaton[ruleDCKV16a , RandomInteger[1, 100], 50],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → 300]
```



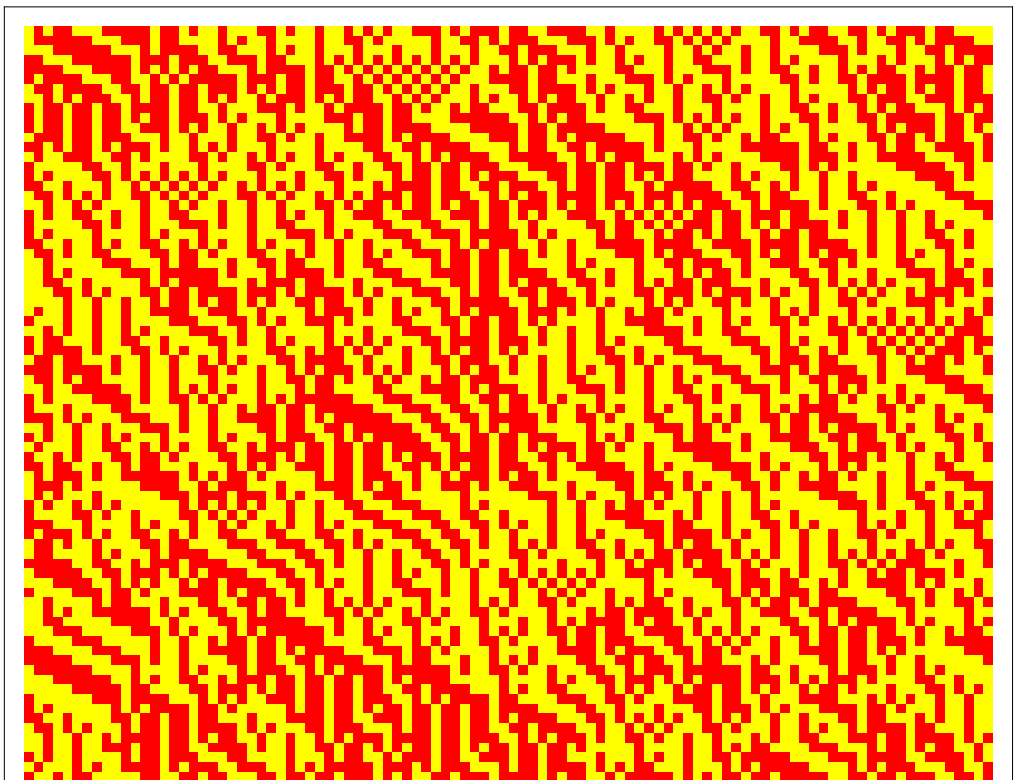
```
GraphPlot[# -> CellularAutomaton[ruleDCKV16a , #] & /@ Tuples[{0, 1}, 6]]
```

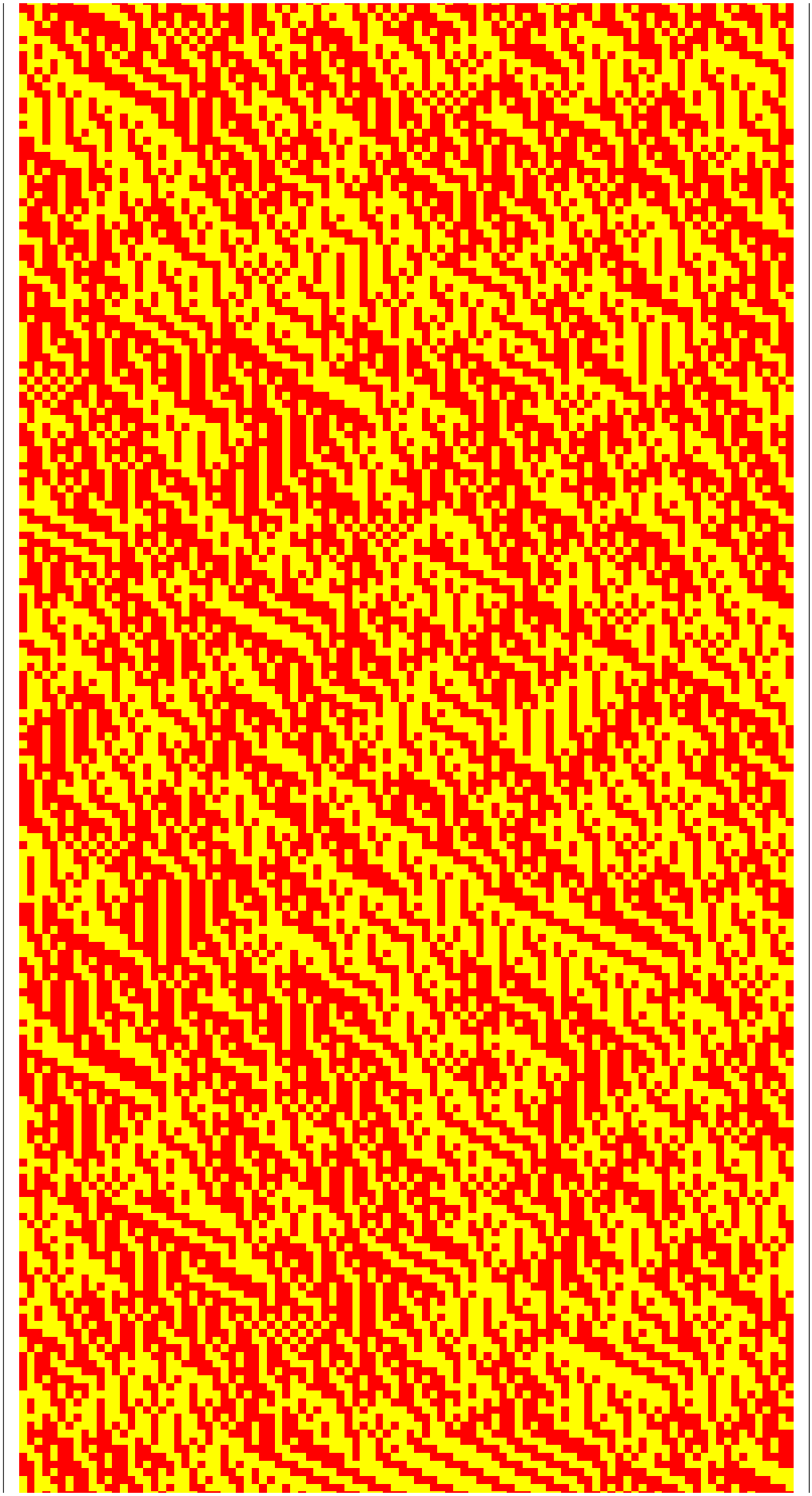


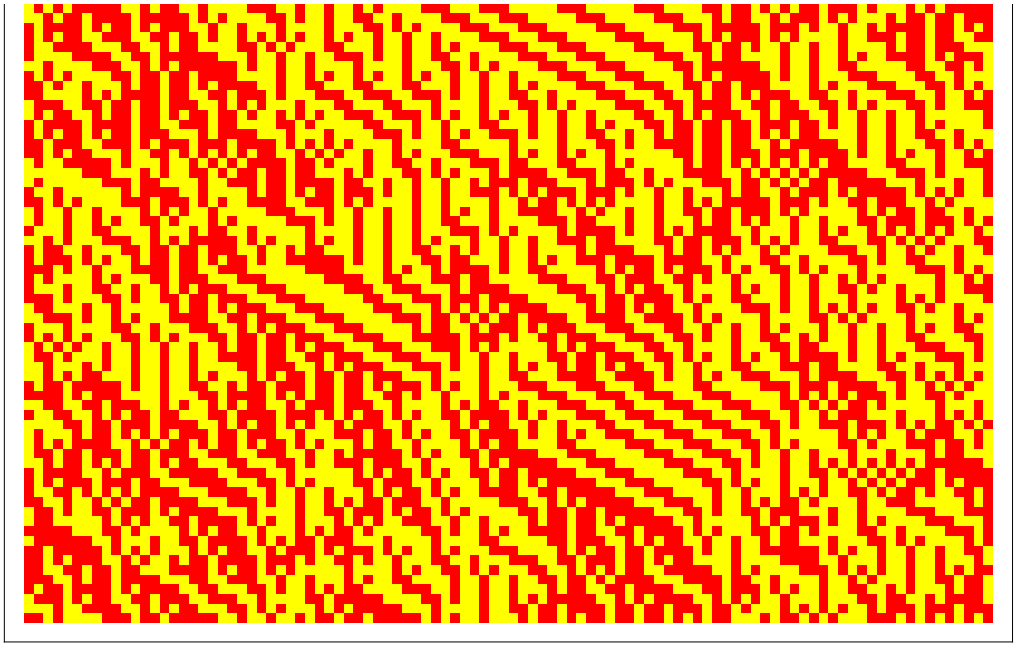
```
ArrayPlot[CellularAutomaton[
  ruleDCKV14[{11 111, 11 221, 12 212, 12 111, 11 212, 12 122, 12 121, 12 221, 11 222,
    11 121, 12 122, 12 221, 11 222, 12 212}], RandomInteger[1, 100], 111],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → True, ImageSize → {400, 300}]
```



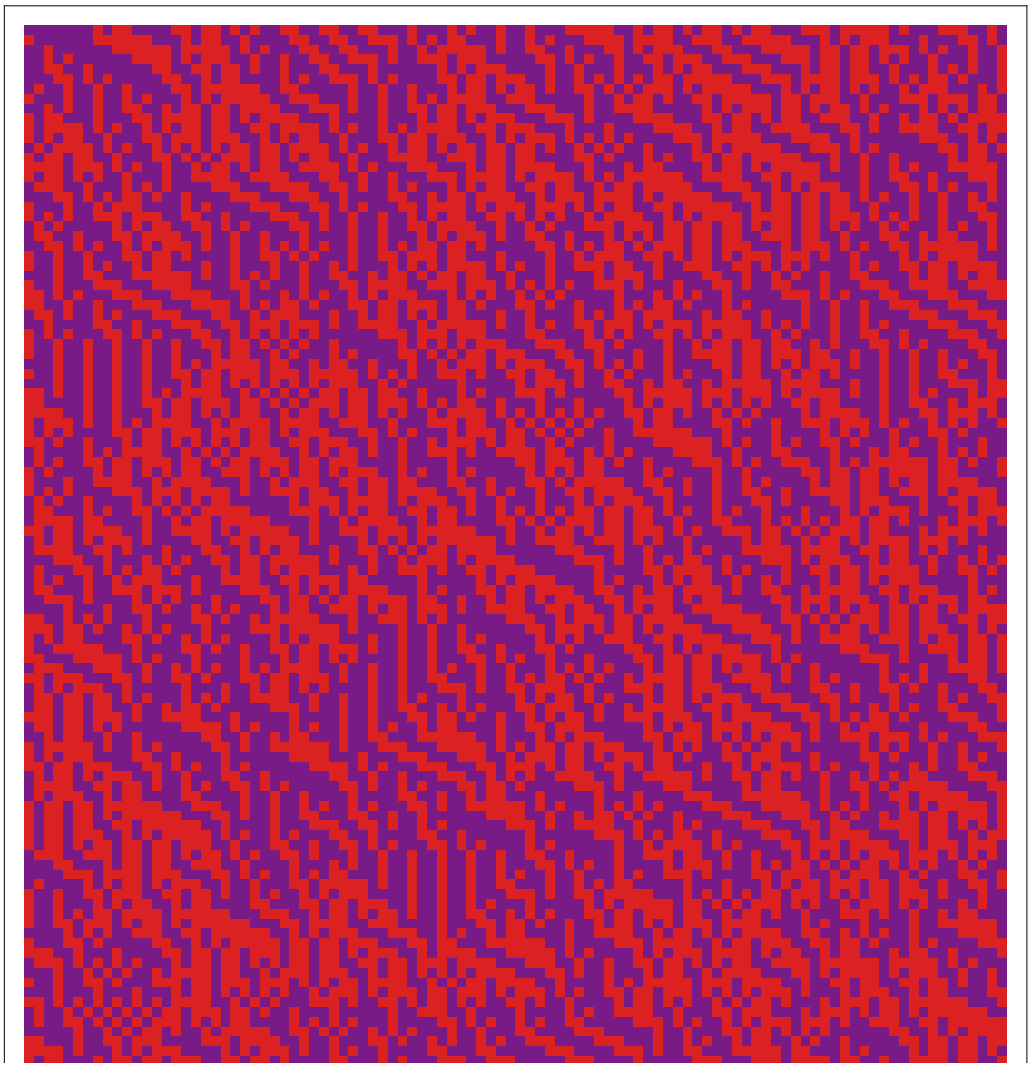
```
ArrayPlot[CellularAutomaton[
  ruleDCKV14[{11 111, 11 221, 12 212, 12 111, 11 212, 12 122, 12 121, 12 221, 11 222,
    11 121, 12 122, 12 221, 11 222, 12 212}], RandomInteger[1, 100], {333, All}],
 ColorRules -> {1 -> Red, 0 -> Yellow}, Mesh → False, ImageSize → 100]
```

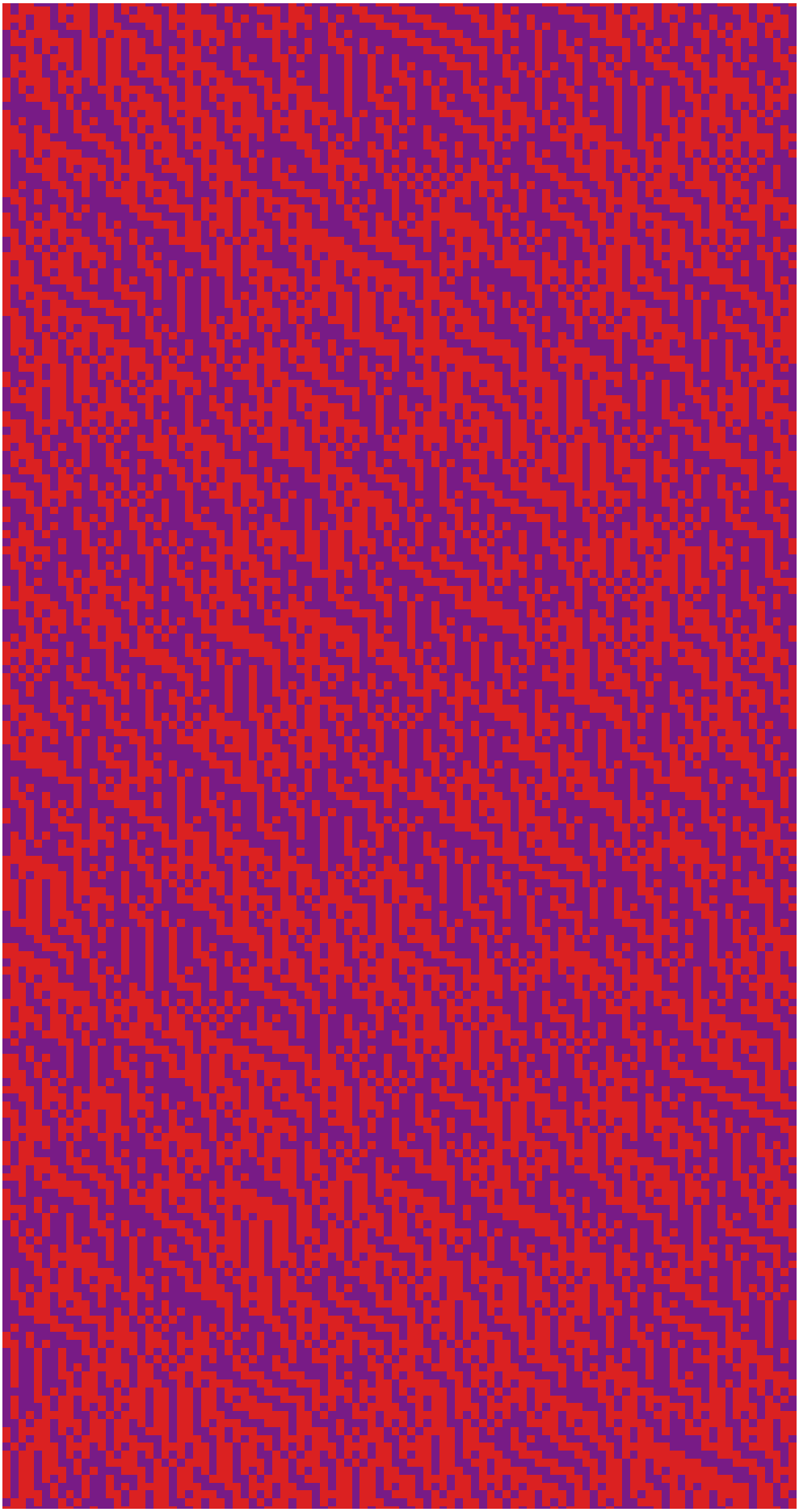


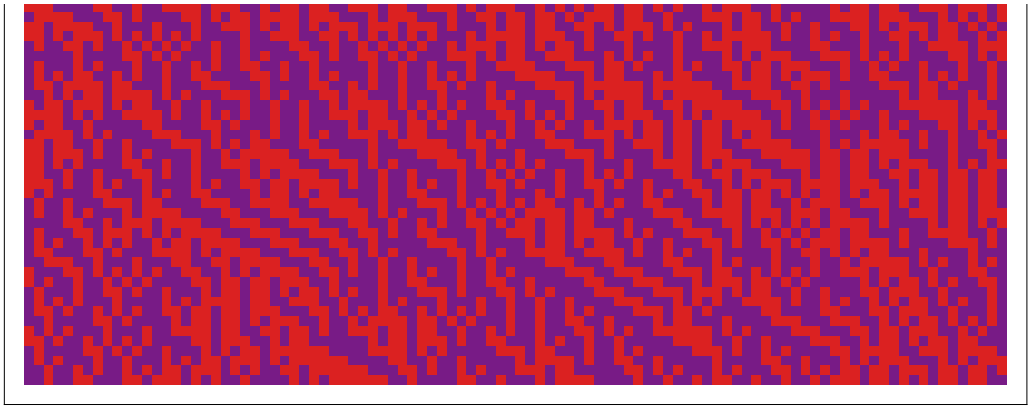




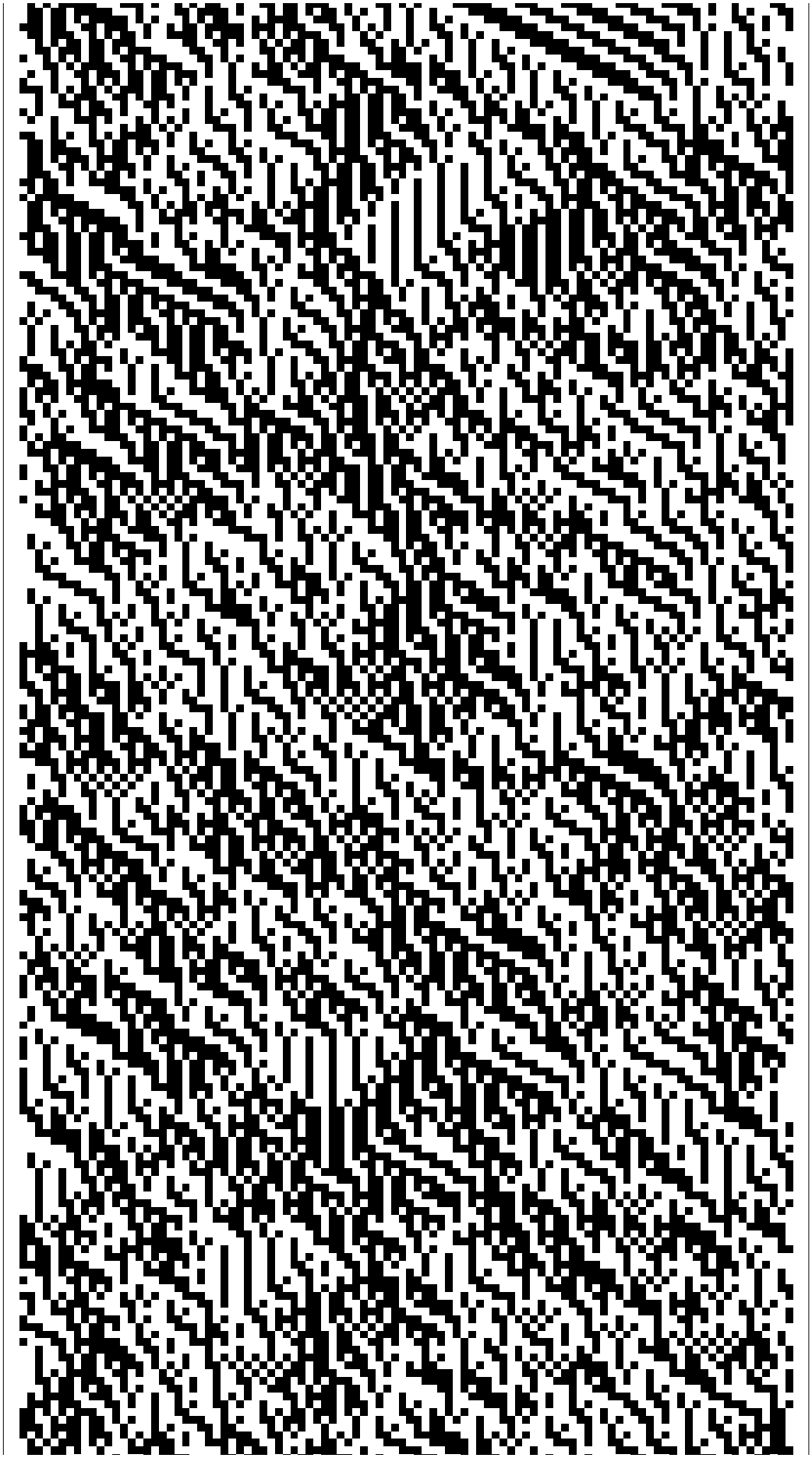
```
ArrayPlot[CellularAutomaton[
  ruleDCKV14[{11 111, 11 221, 12 212, 12 111, 11 212, 12 122, 12 121, 12 221, 11 222,
    11 121, 12 122, 12 221, 11 222, 12 212}] , RandomInteger[1, 100], {333, All}],
 ColorFunction → "Rainbow", Mesh → False, ImageSize → 100]
```

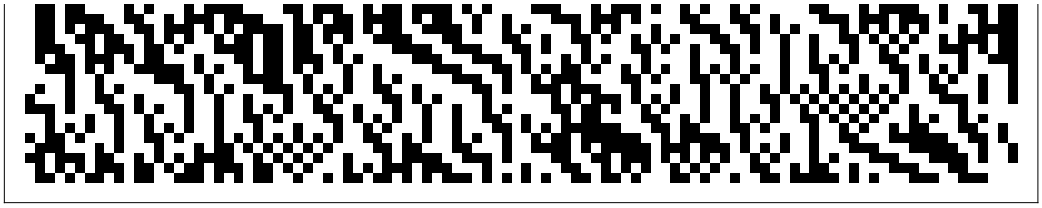




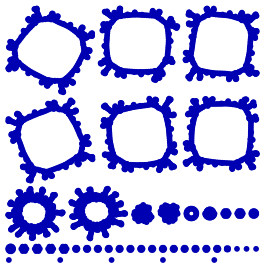


```
ArrayPlot[CellularAutomaton[
  ruleDCKV14[{11 111, 11 221, 12 212, 12 111, 11 212, 12 122, 12 121, 12 221, 11 222,
    11 121, 12 122, 12 221, 11 222, 12 212}], RandomInteger[1, 100], {333, All}],
 Mesh → False, ImageSize → 100]
```

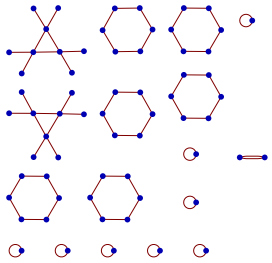




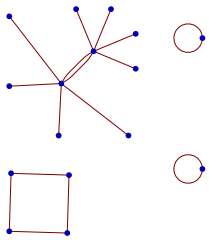
```
GraphPlot[# -> CellularAutomaton[
      ruleDCKV14[{11111, 11221, 12212, 12111, 11212, 12122, 12121, 12221, 11222,
        11121, 12122, 12221, 11222, 12212}], #] & /@ Tuples[{0, 1}, 12]]
```



```
GraphPlot[# -> CellularAutomaton[
      ruleDCKV14[{11111, 11221, 12212, 12111, 11212, 12122, 12121, 12221, 11222,
        11121, 12122, 12221, 11222, 12212}], #] & /@ Tuples[{0, 1}, 6]]
```

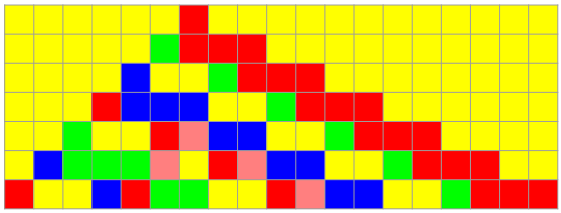


```
GraphPlot[# -> CellularAutomaton[
      ruleDCKV14[{11111, 11221, 12212, 12111, 11212, 12122, 12121, 12221, 11222,
        11121, 12122, 12221, 11222, 12212}], #] & /@ Tuples[{0, 1}, 4]]
```

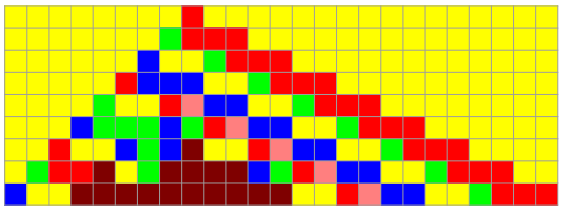


Catalog of $CA^{(5,3)}$ Examples

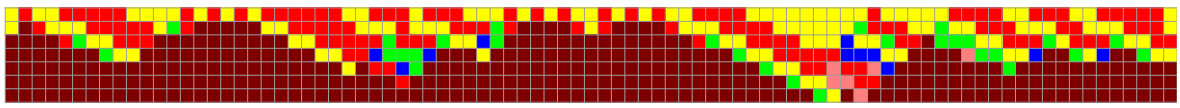
```
ArrayPlot[CellularAutomaton[
  ruleDCKV18[{11111, 11123, 11121, 11221, 12221, 12231, 11112, 12112, 11231,
    12221, 12312, 12223, 12212, 11212, 12133, 12331, 12341, 11233}],
  {{1}, 0}, 6],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green, 4 → Pink},
 Mesh → True, ImageSize → 300]
```



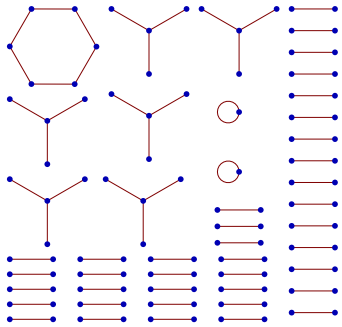
```
ArrayPlot[
 CellularAutomaton[ruleDCKV22[{11111, 11123, 11121, 11221, 12221, 12231,
    11112, 12112, 11231, 12221, 12312, 12223, 12212, 11212,
    12133, 12331, 12323, 12332, 12345, 12314, 11234, 11234}],
  {{1}, 0}, 8],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green, 4 → Pink},
 Mesh → True, ImageSize → 300]
```



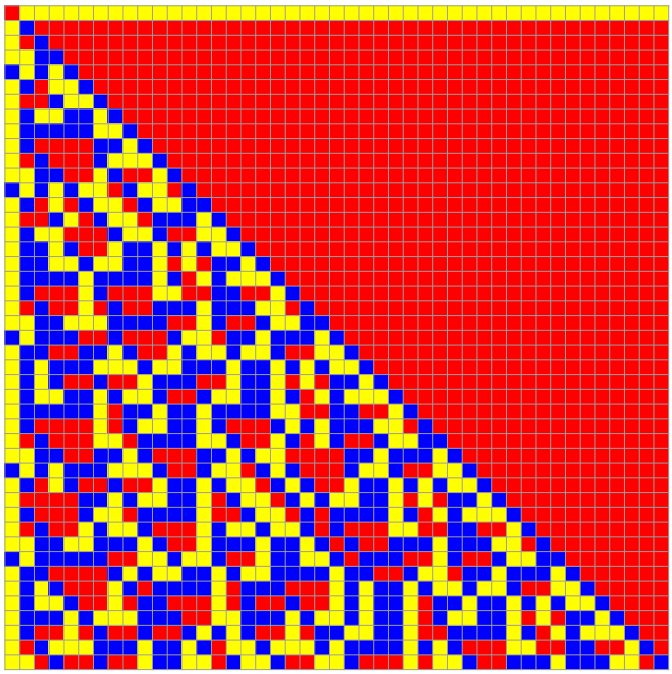
```
ArrayPlot[CellularAutomaton[
  ruleDCKV18[{11111, 11123, 11121, 11221, 12221, 12231, 11112, 12112, 11231,
    12221, 12312, 12223, 12212, 11212, 12133, 12331, 12341, 11233}],
  RandomInteger[1, 100], 6],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green, 4 → Pink},
 Mesh → True, ImageSize → 300]
```

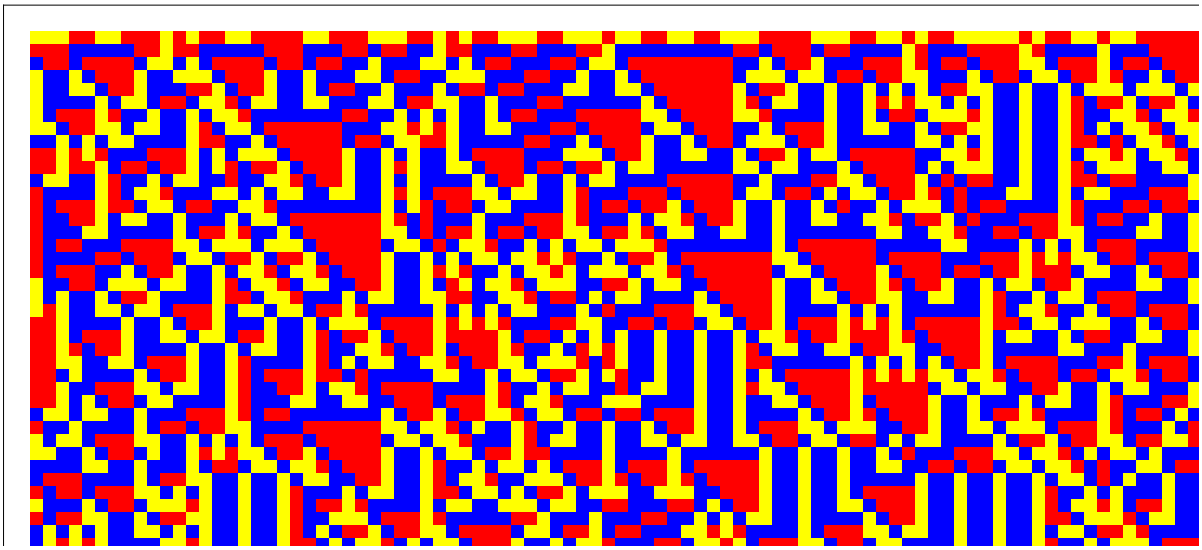


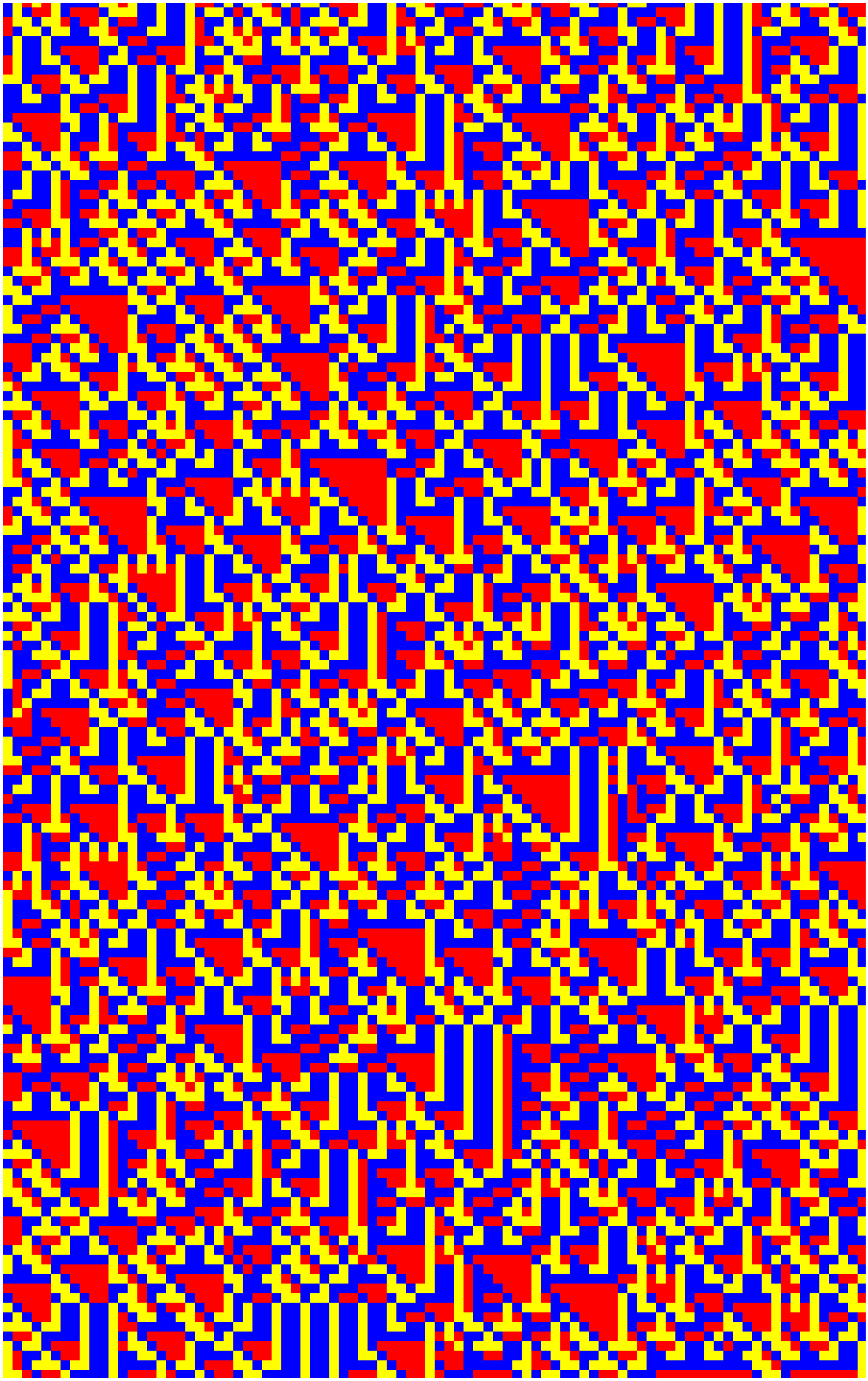
```
GraphPlot[# -> CellularAutomaton[
      ruleDCKV18[{11 111, 11 123, 11 121, 11 221, 12 221, 12 231, 11 112, 12 112, 11 231,
        12 221, 12 312, 12 223, 12 212, 11 212, 12 133, 12 331, 12 341, 11 233}], #]
   & /@ Tuples[{0, 1}, 6]]
```

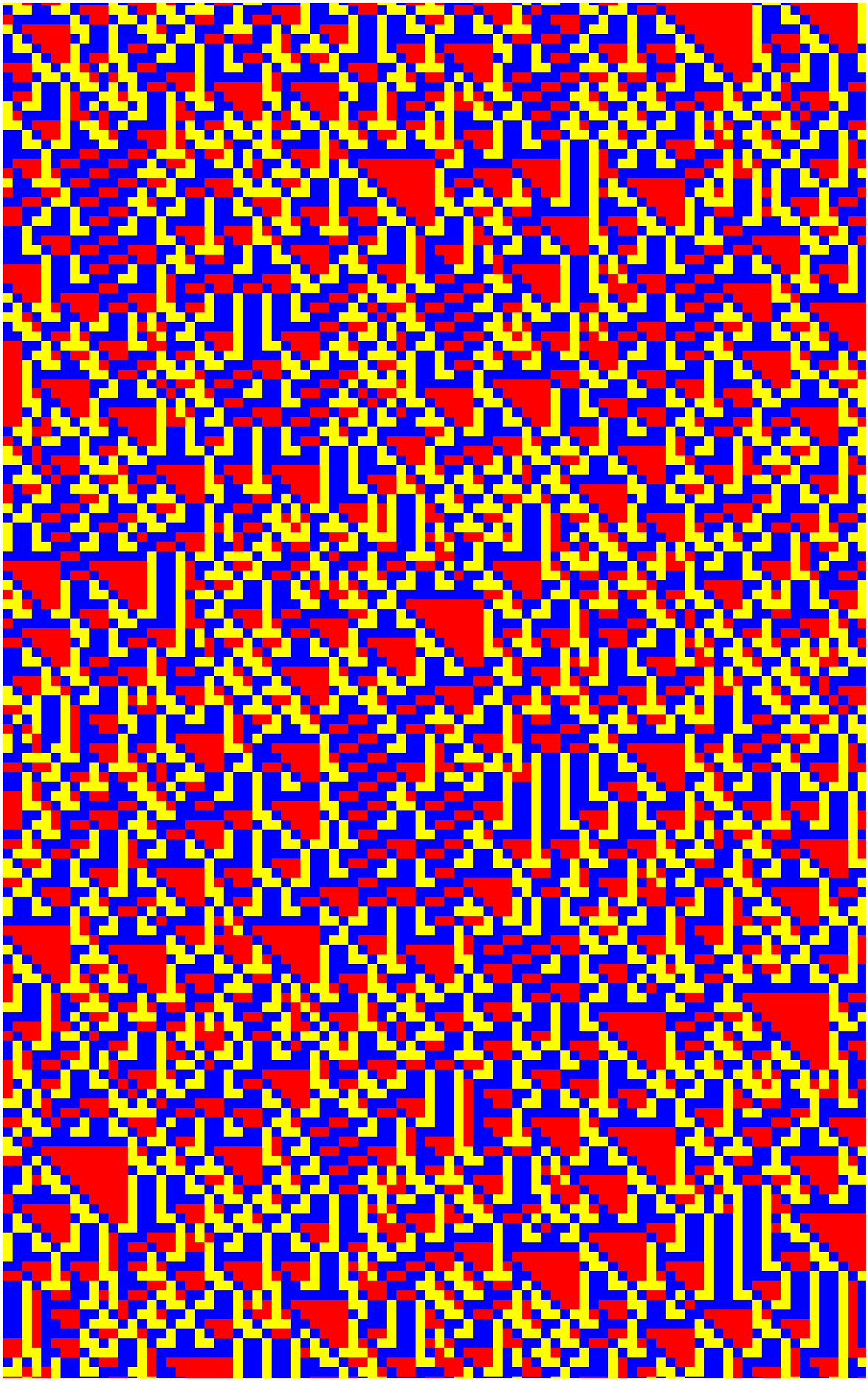


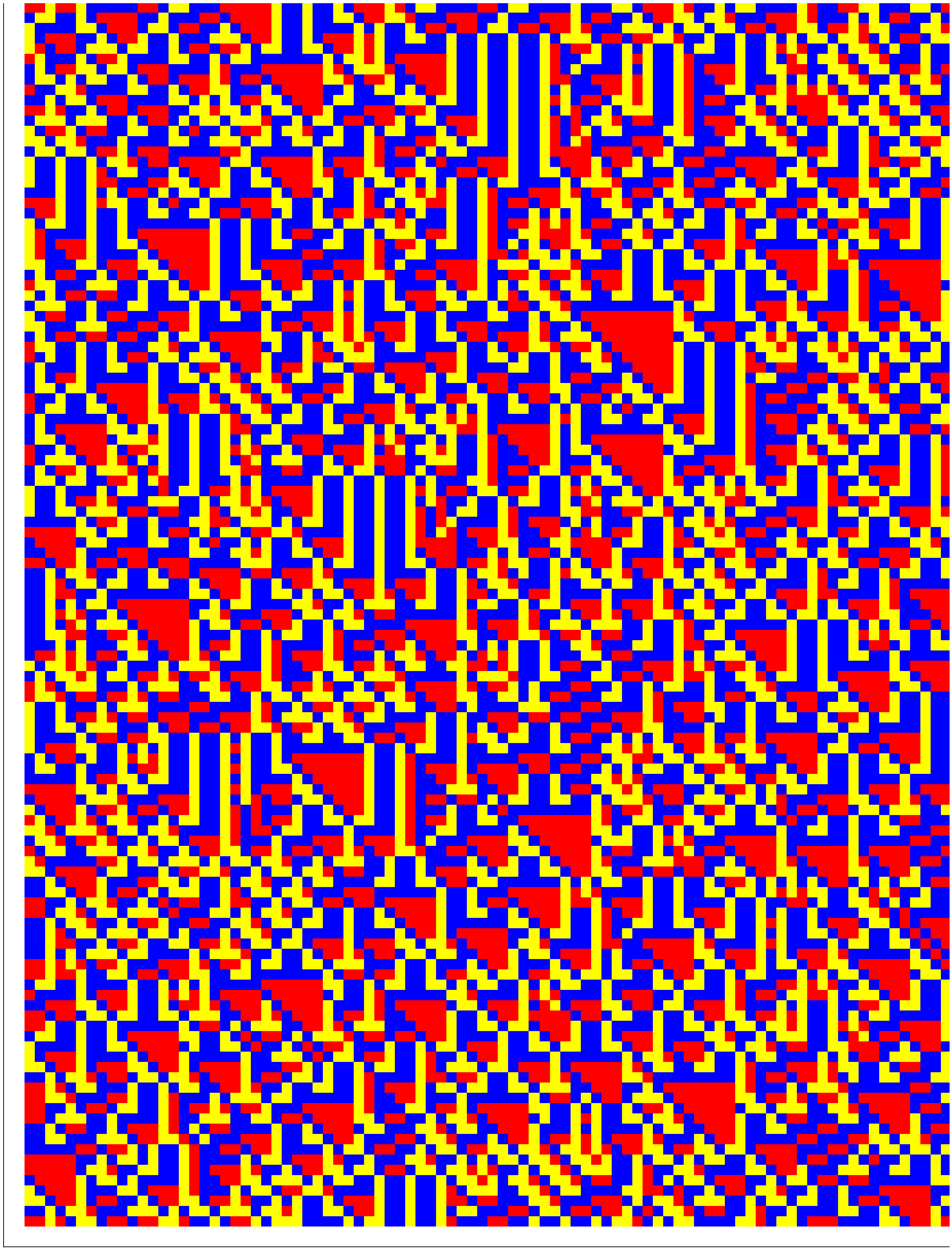
static

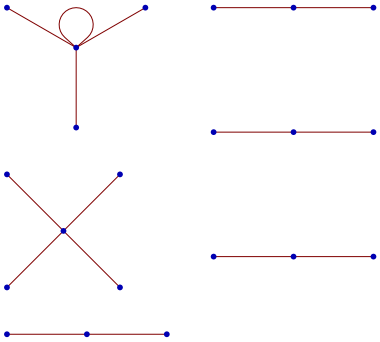




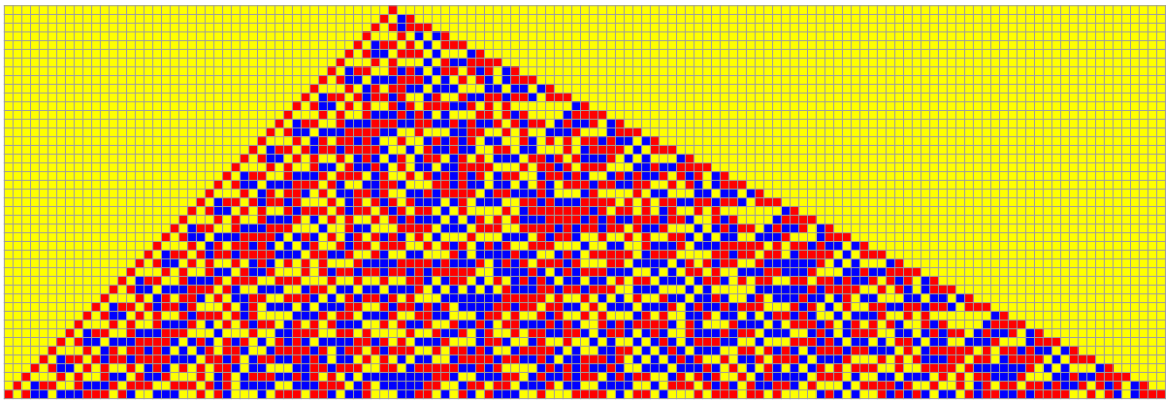


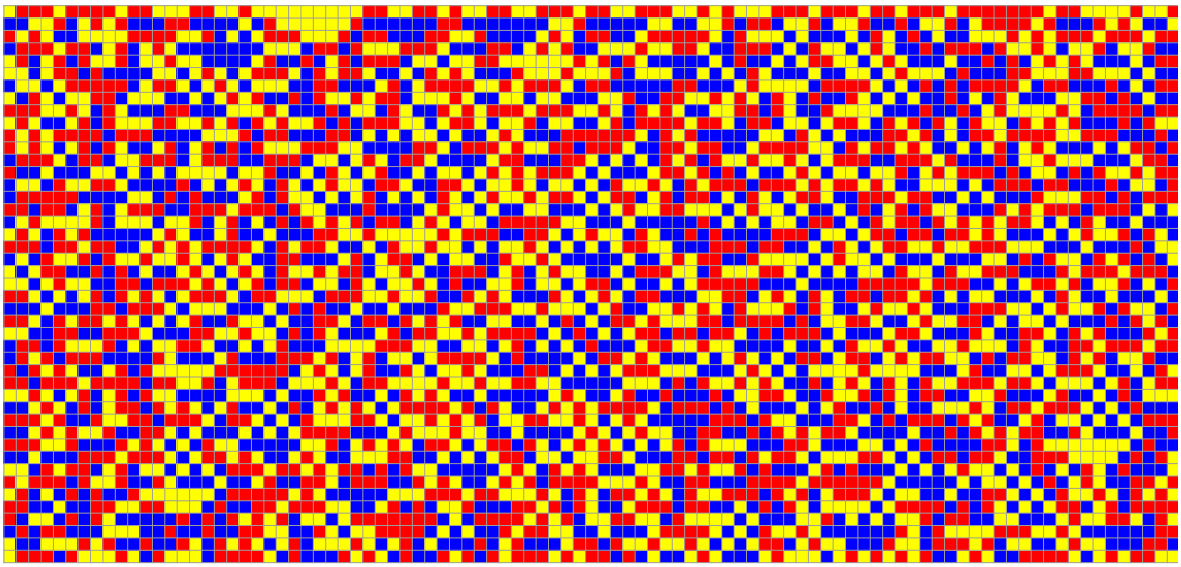


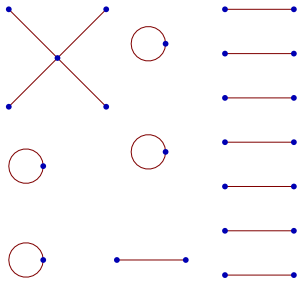




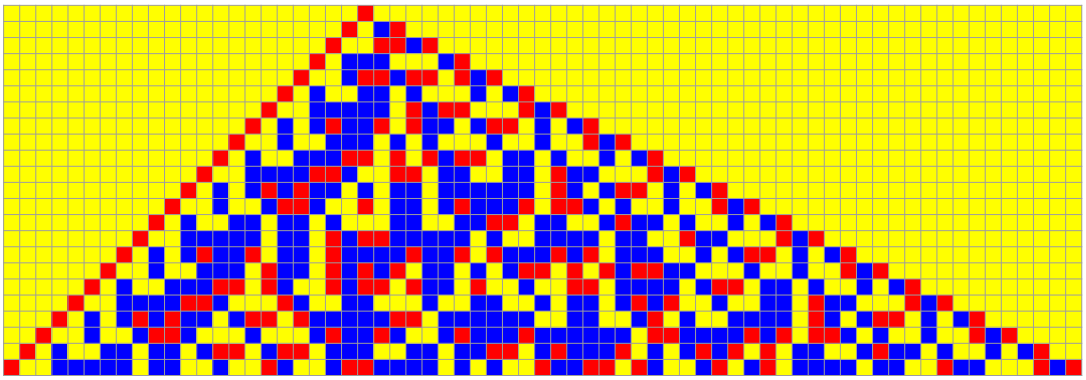
dynamic

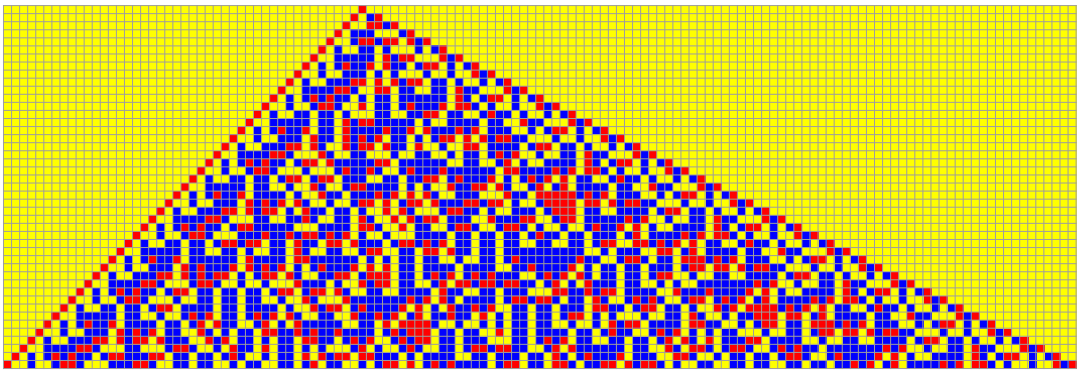


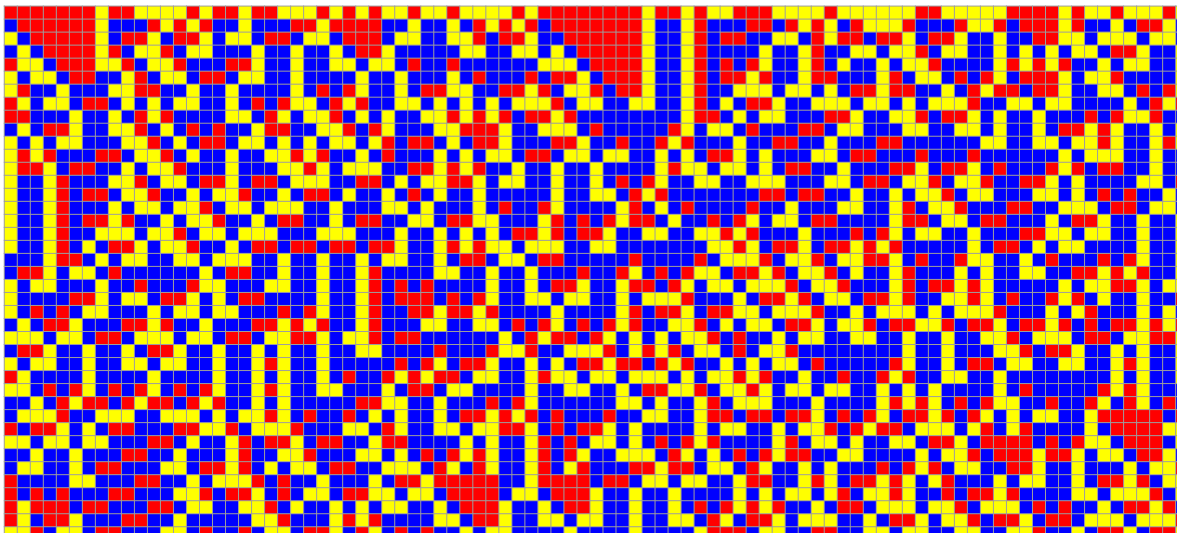


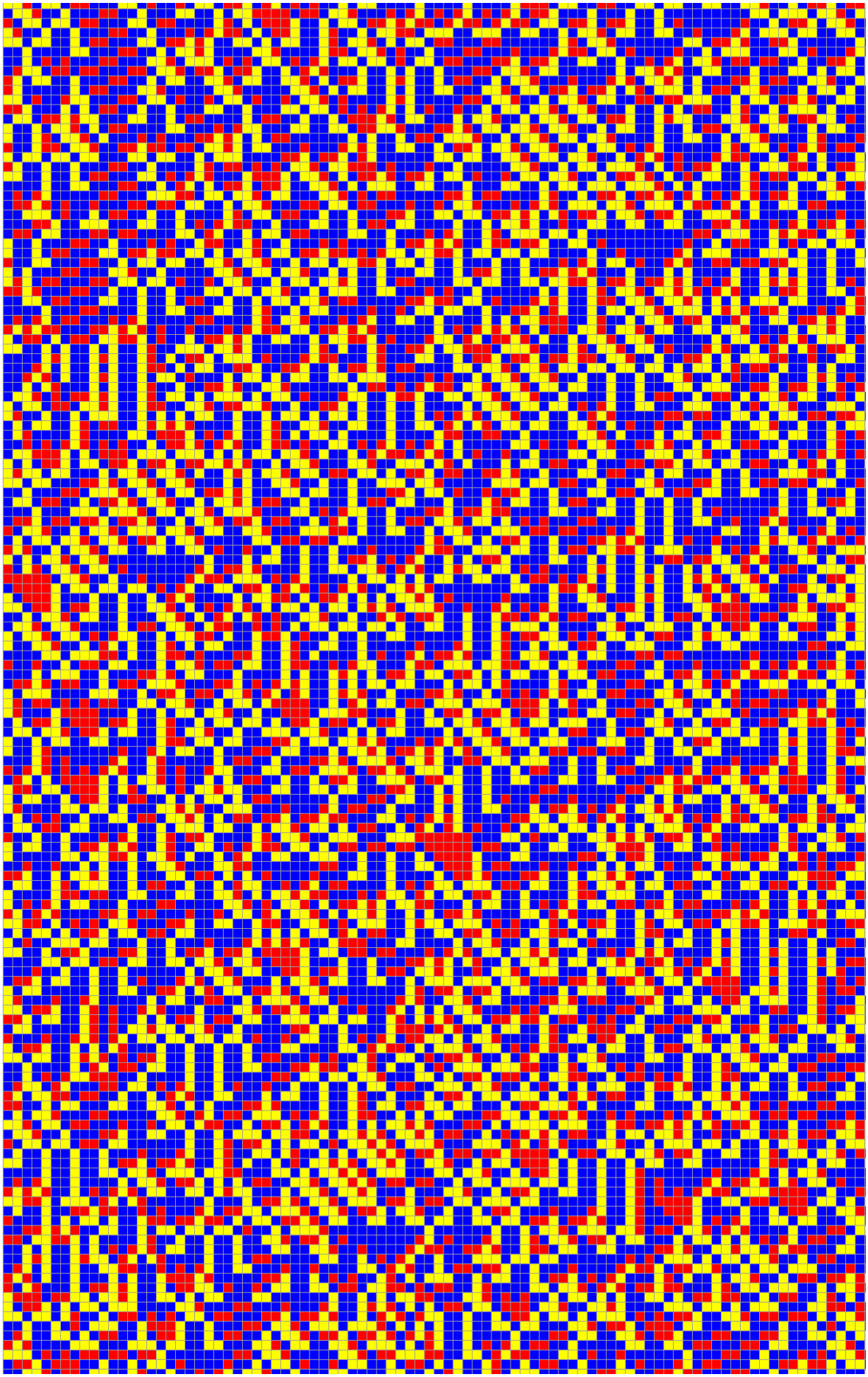


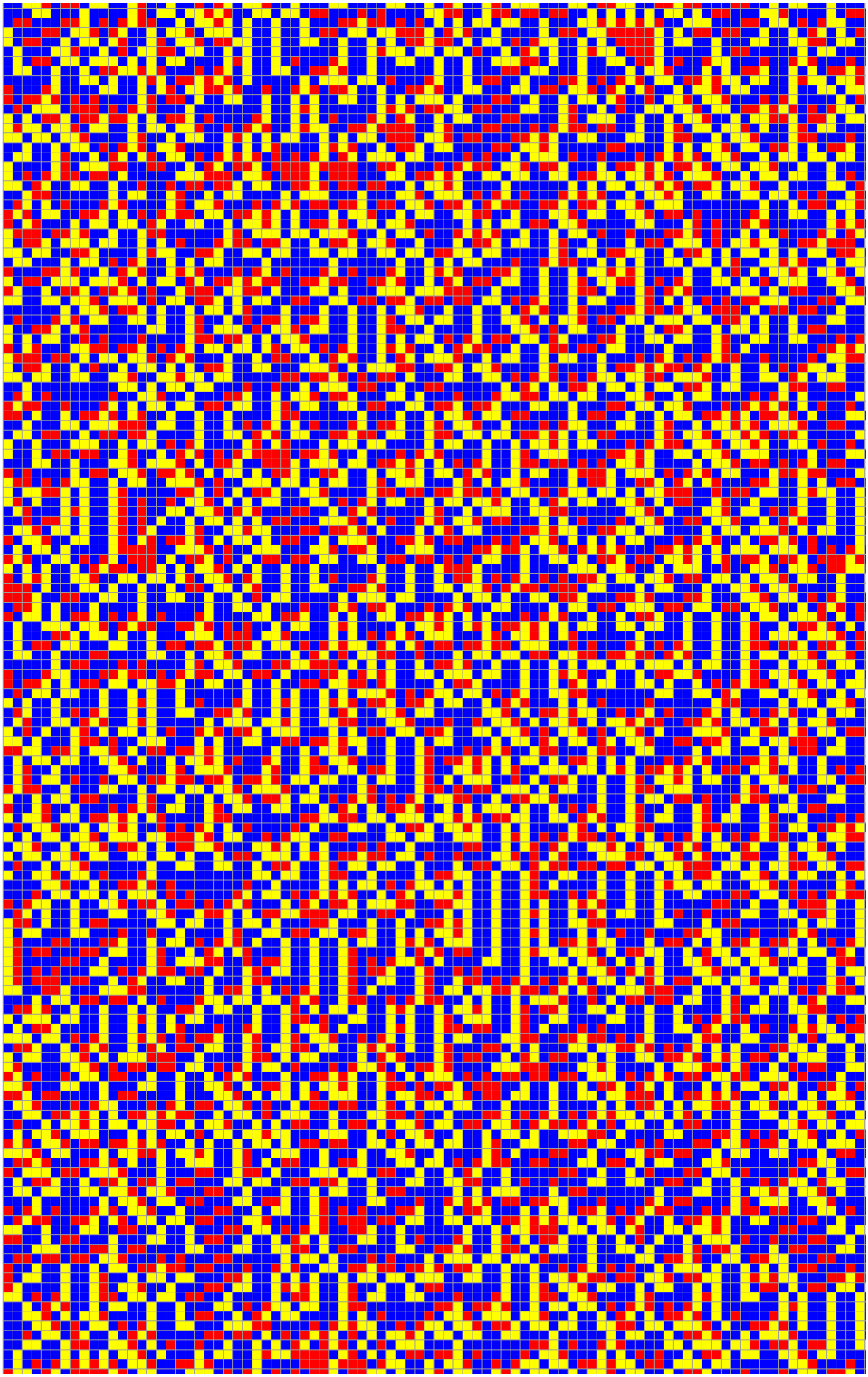
Static

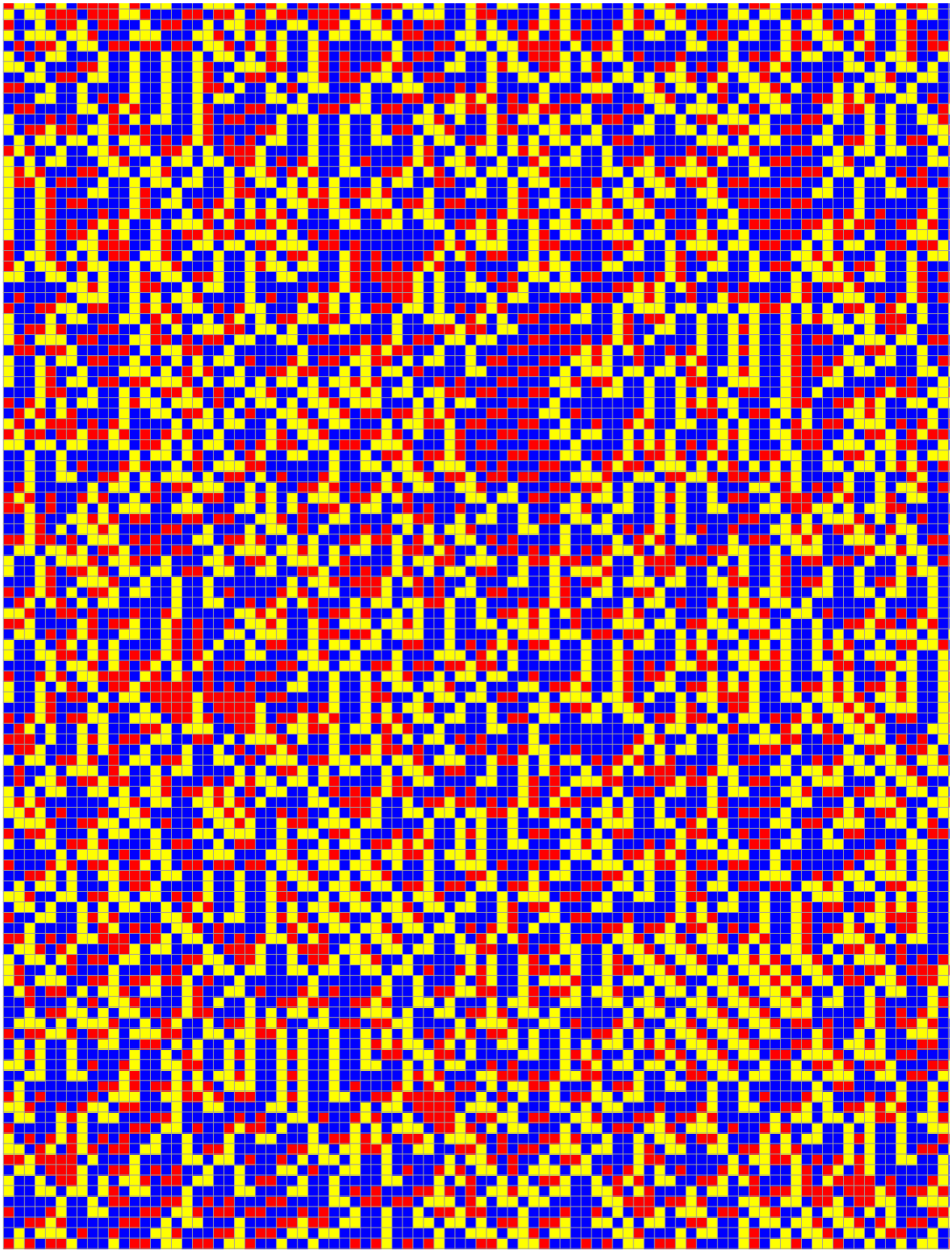


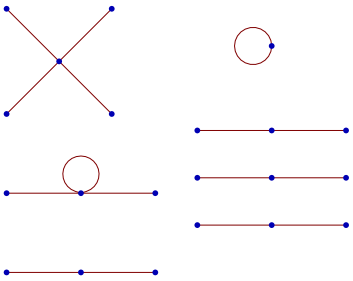




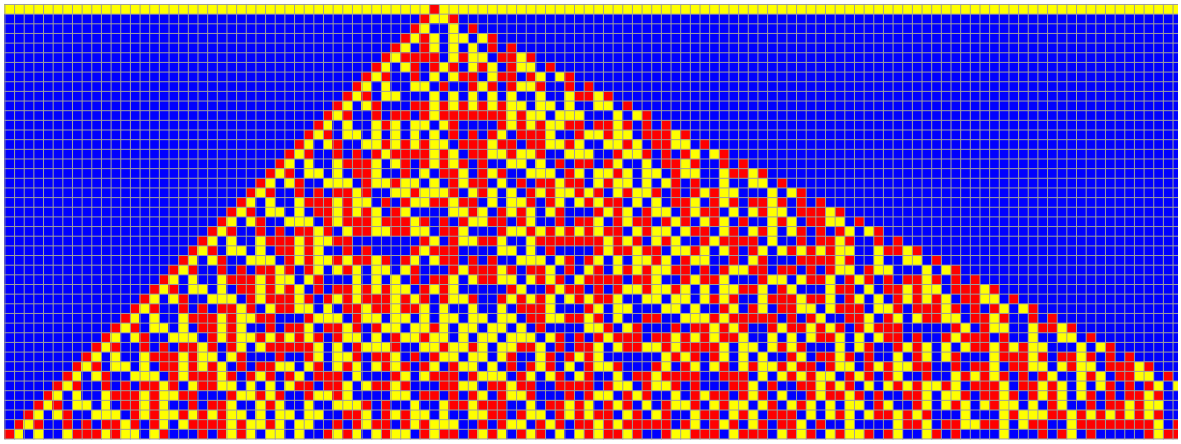


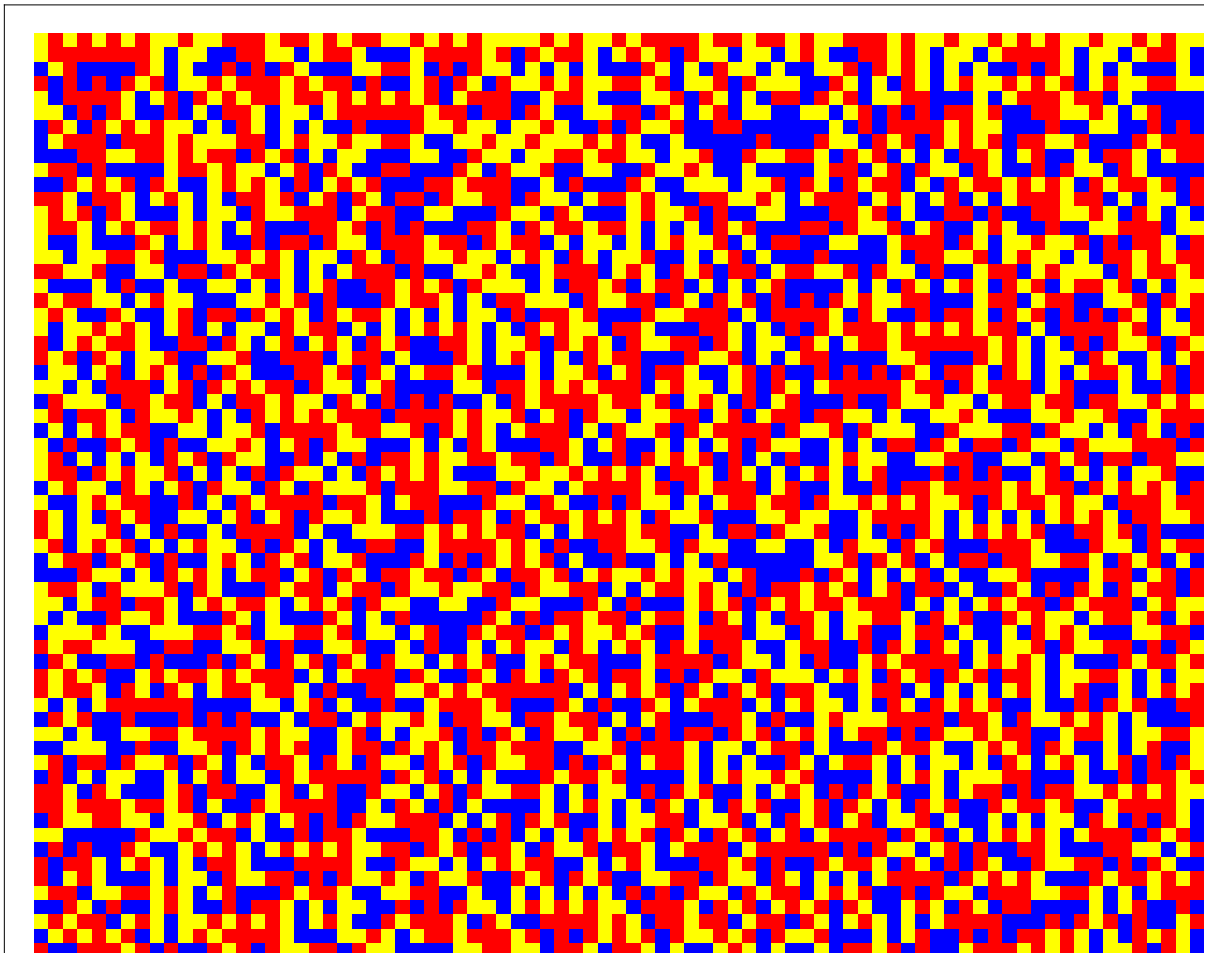


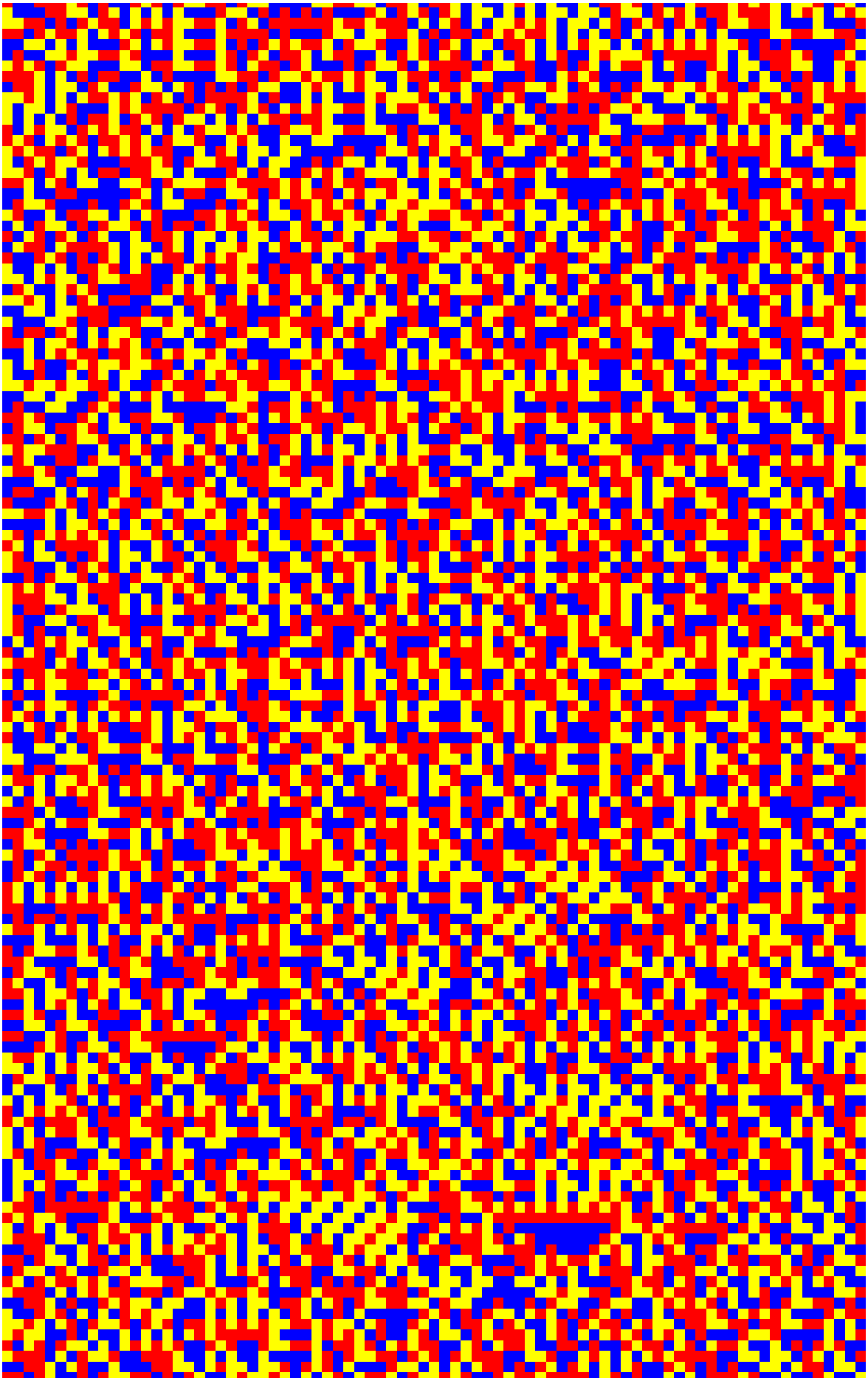


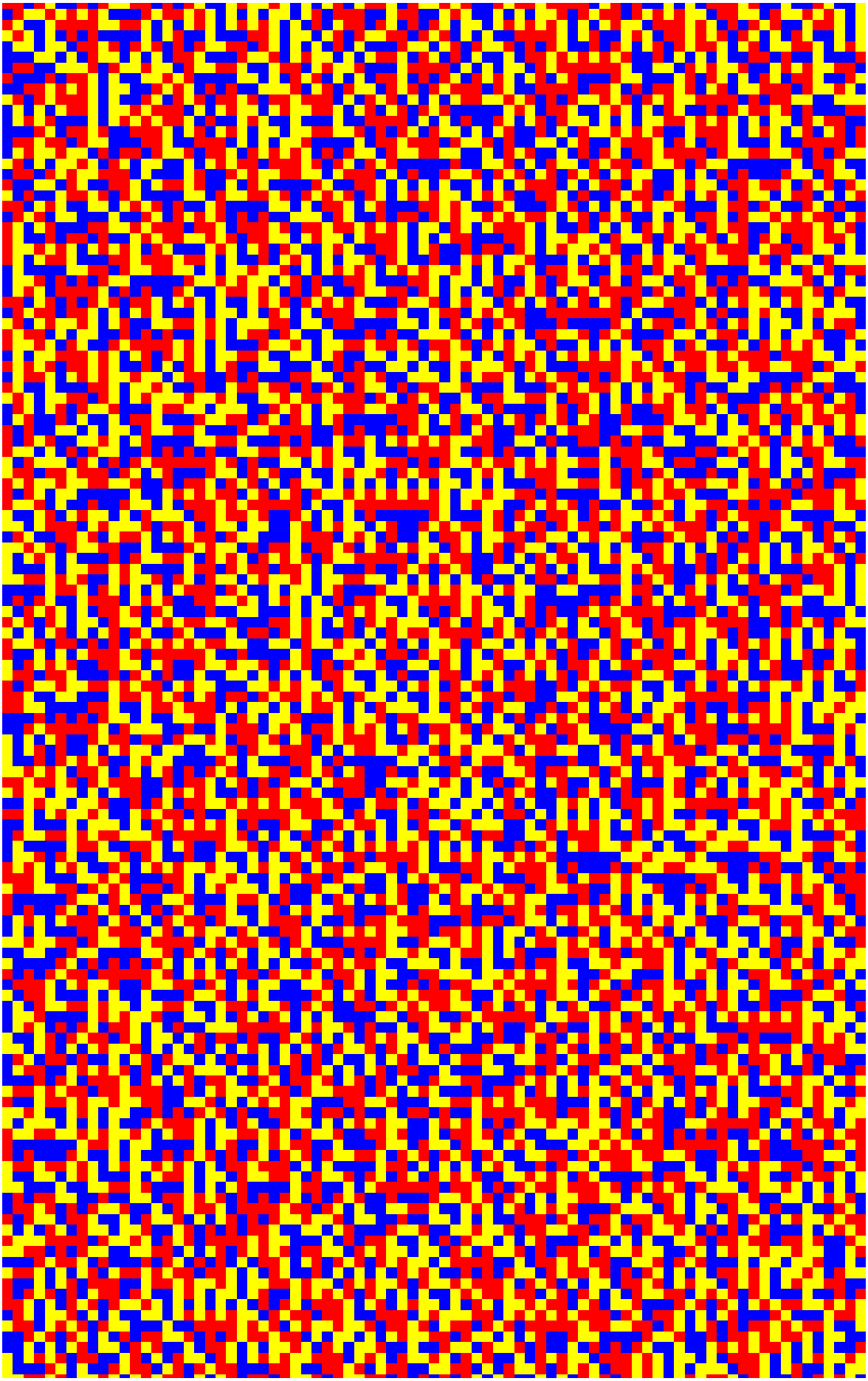


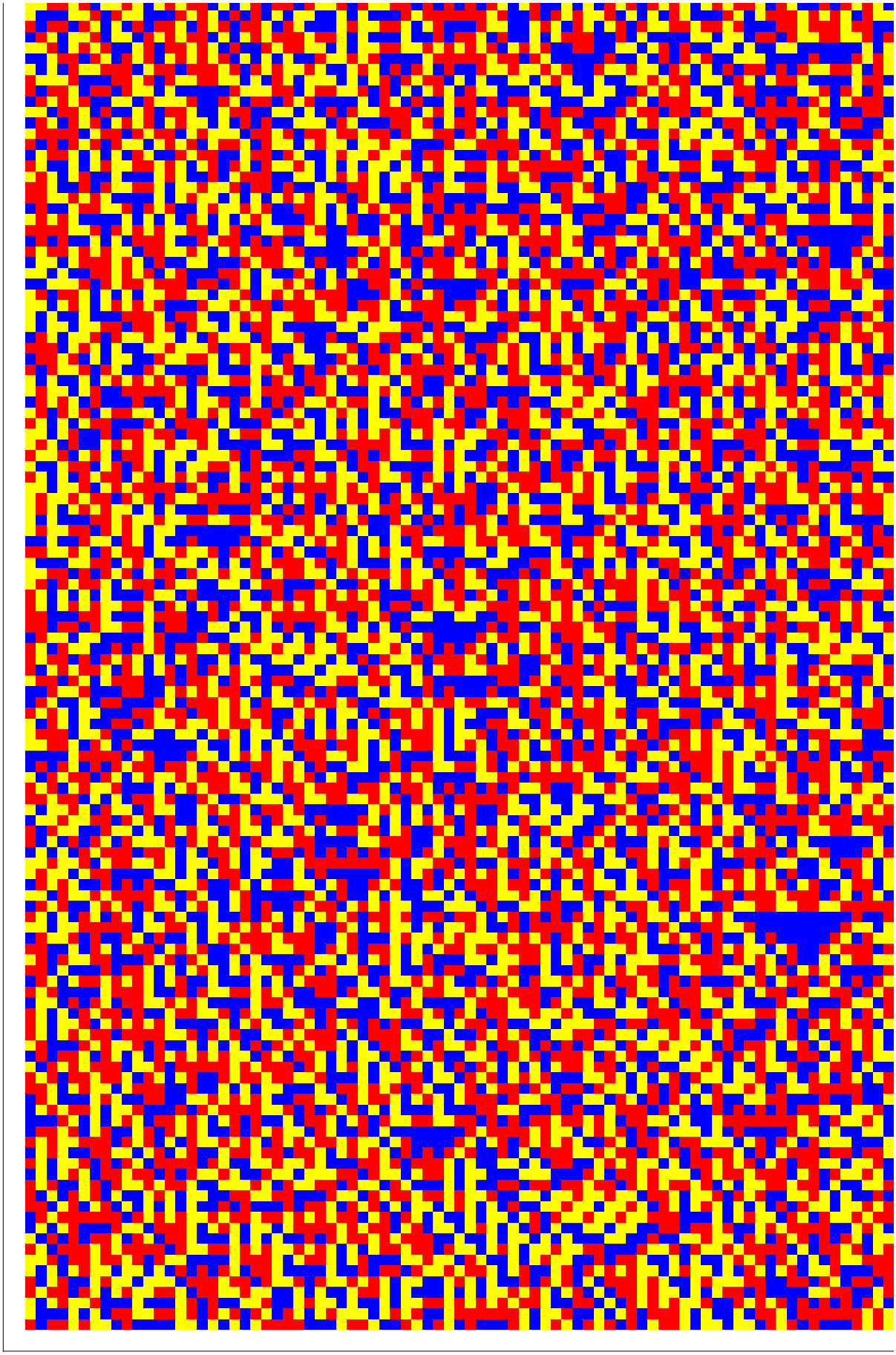
static

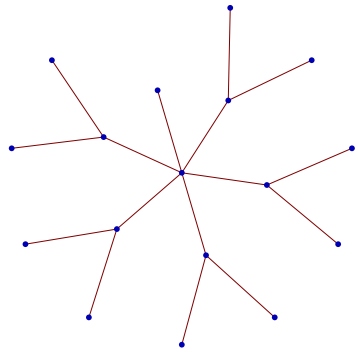




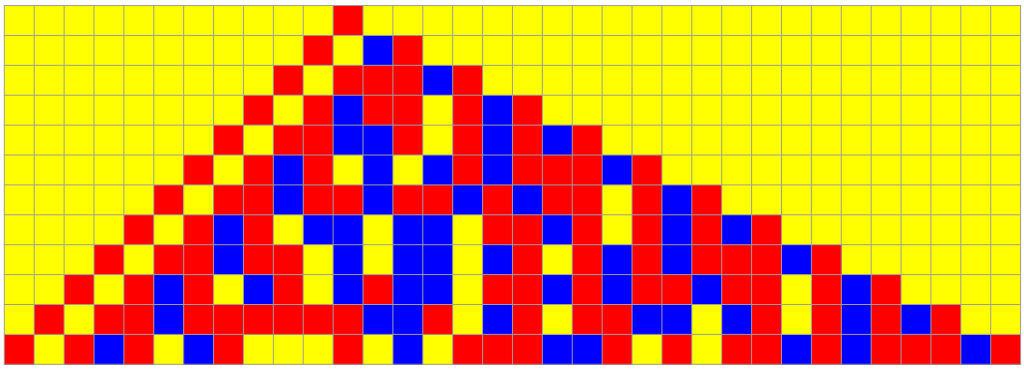


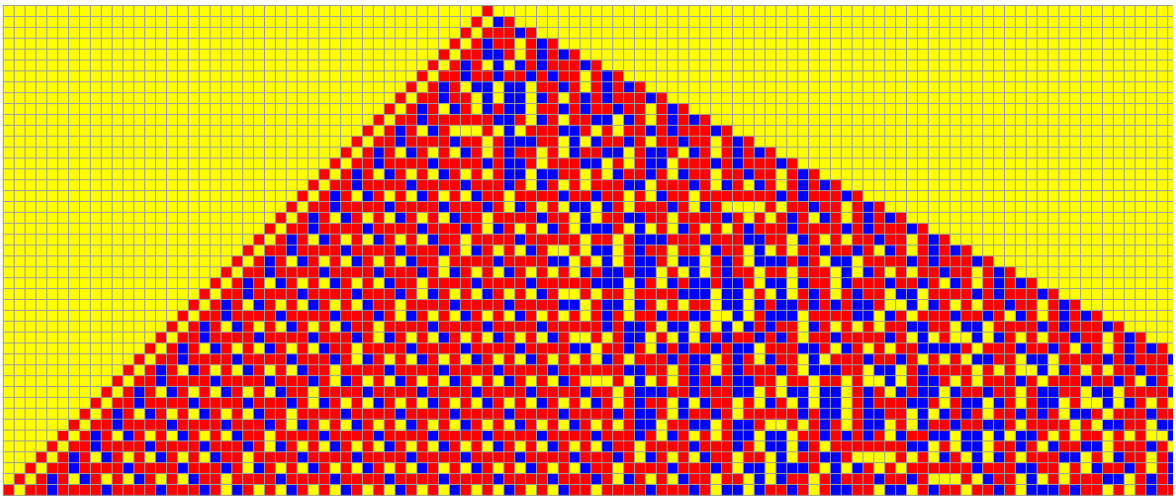


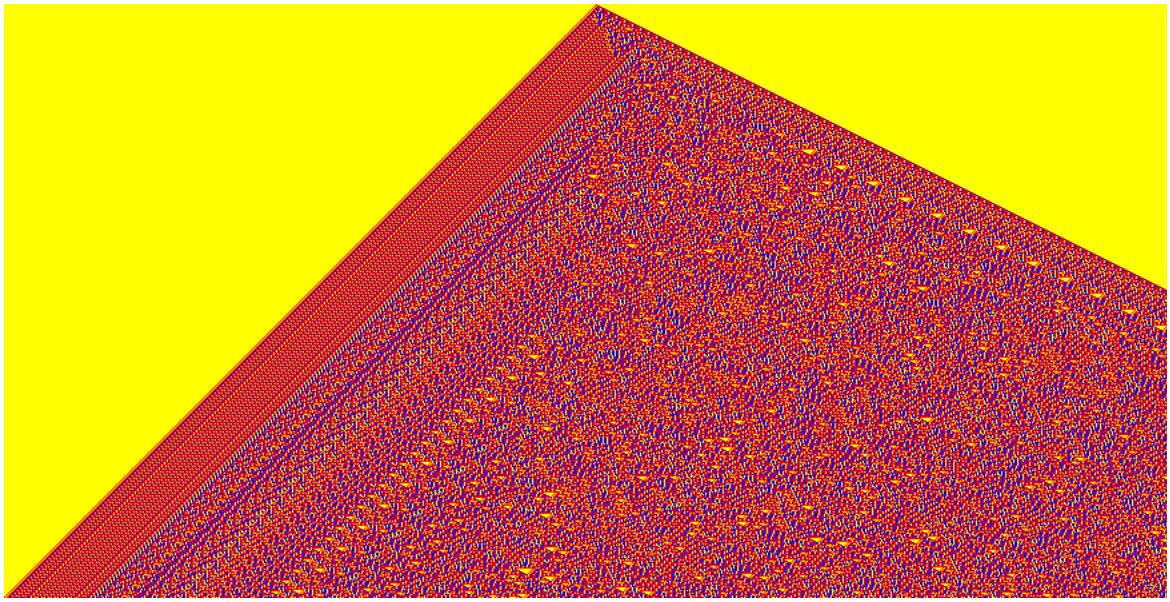


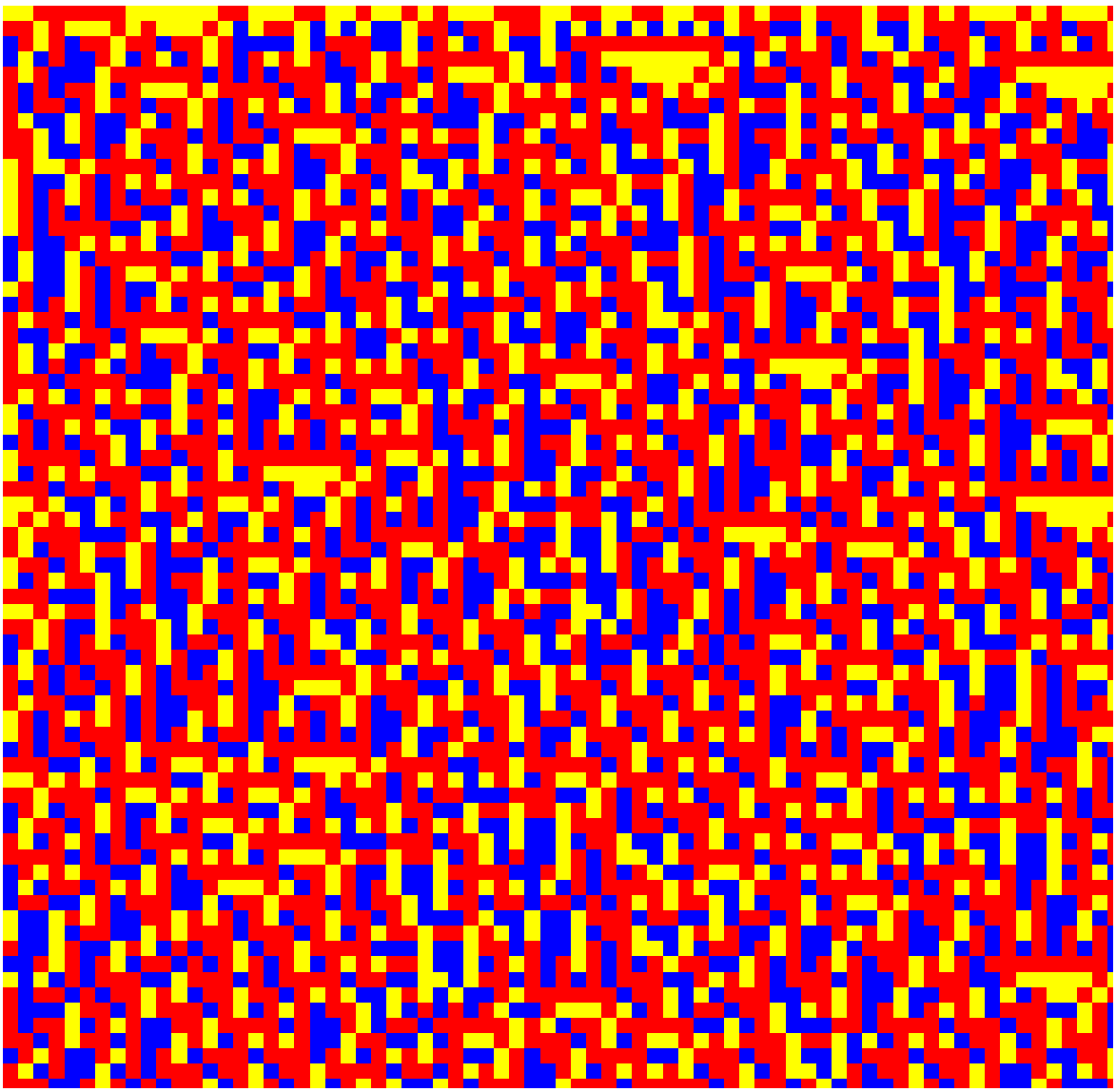


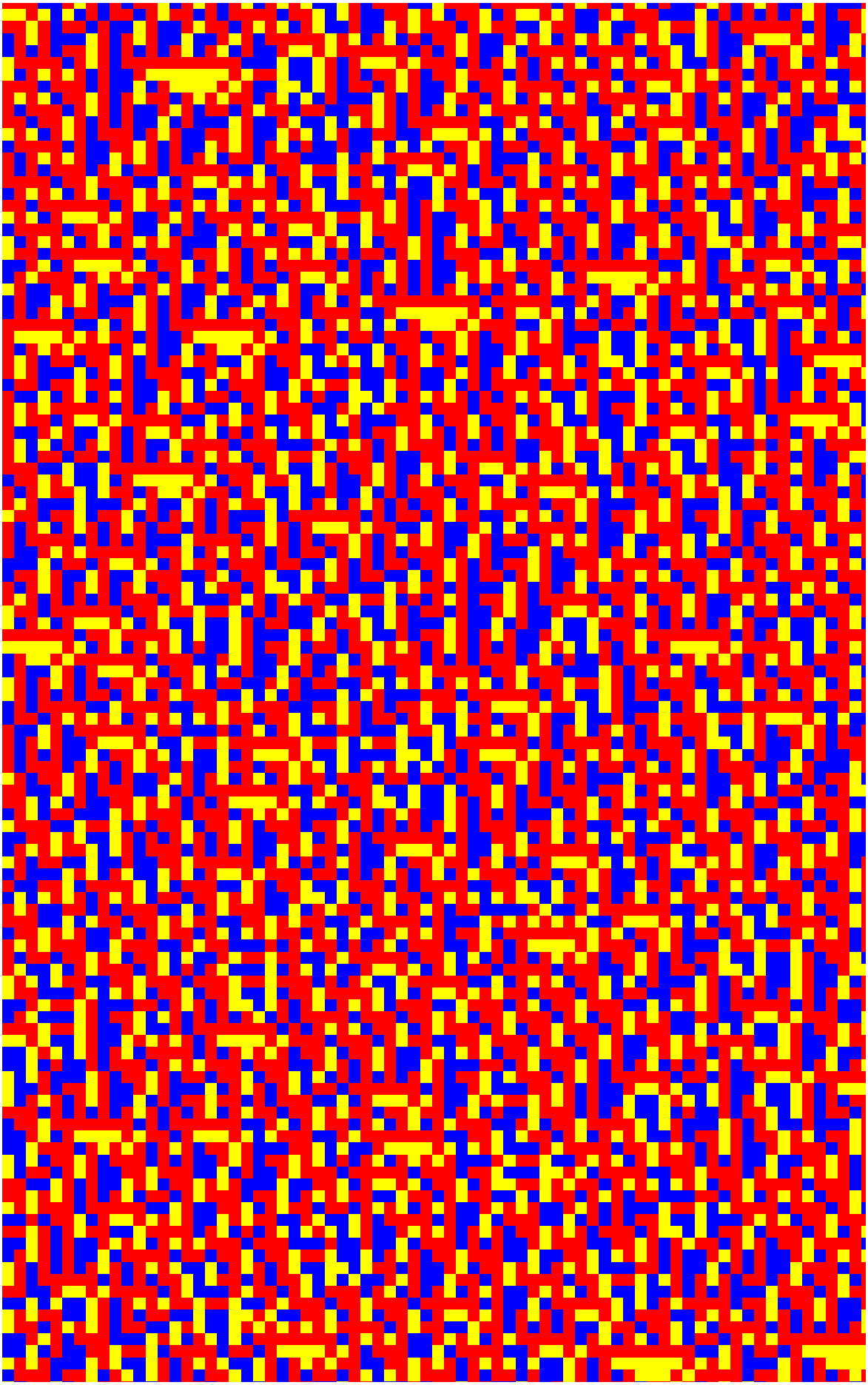
static

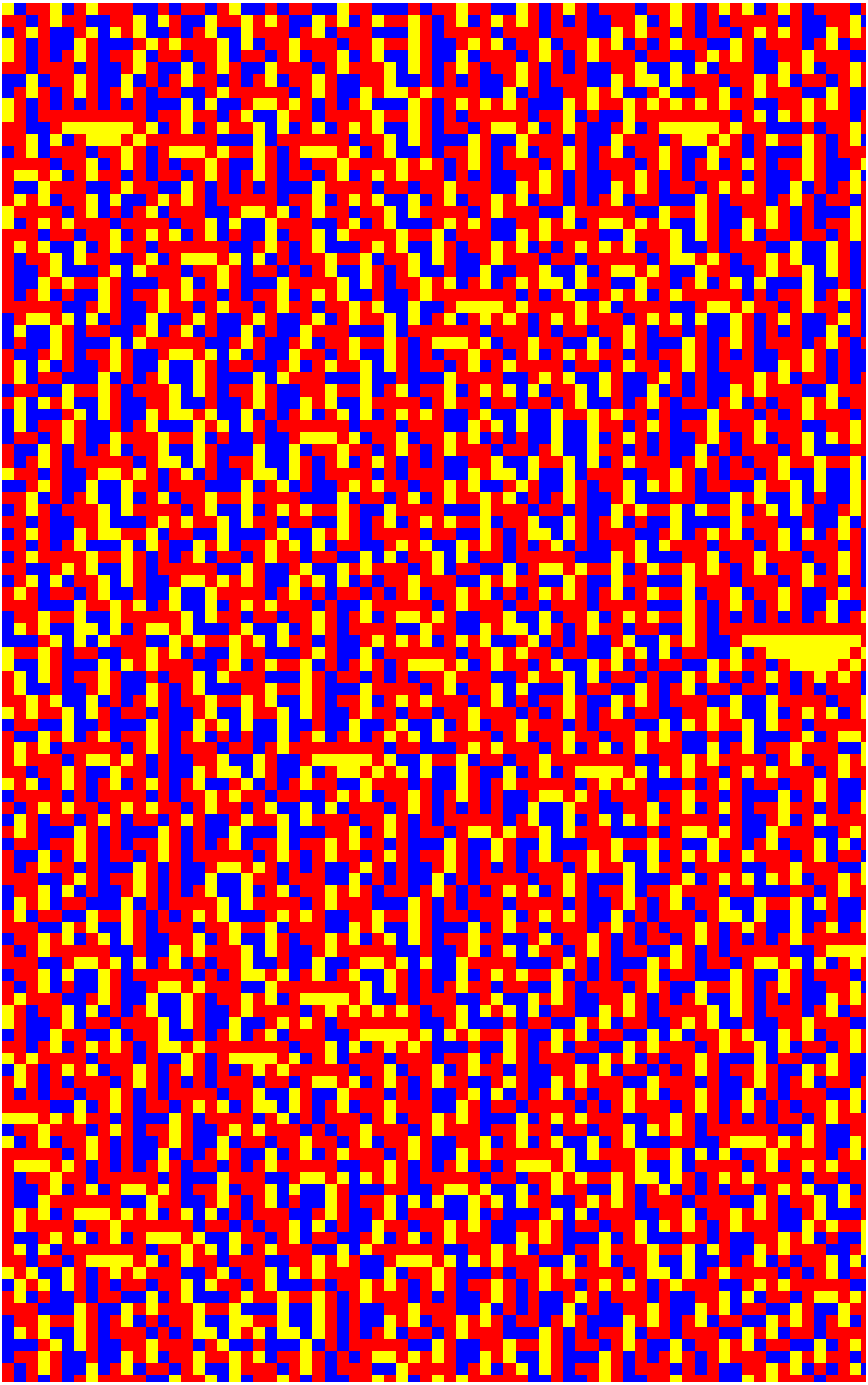


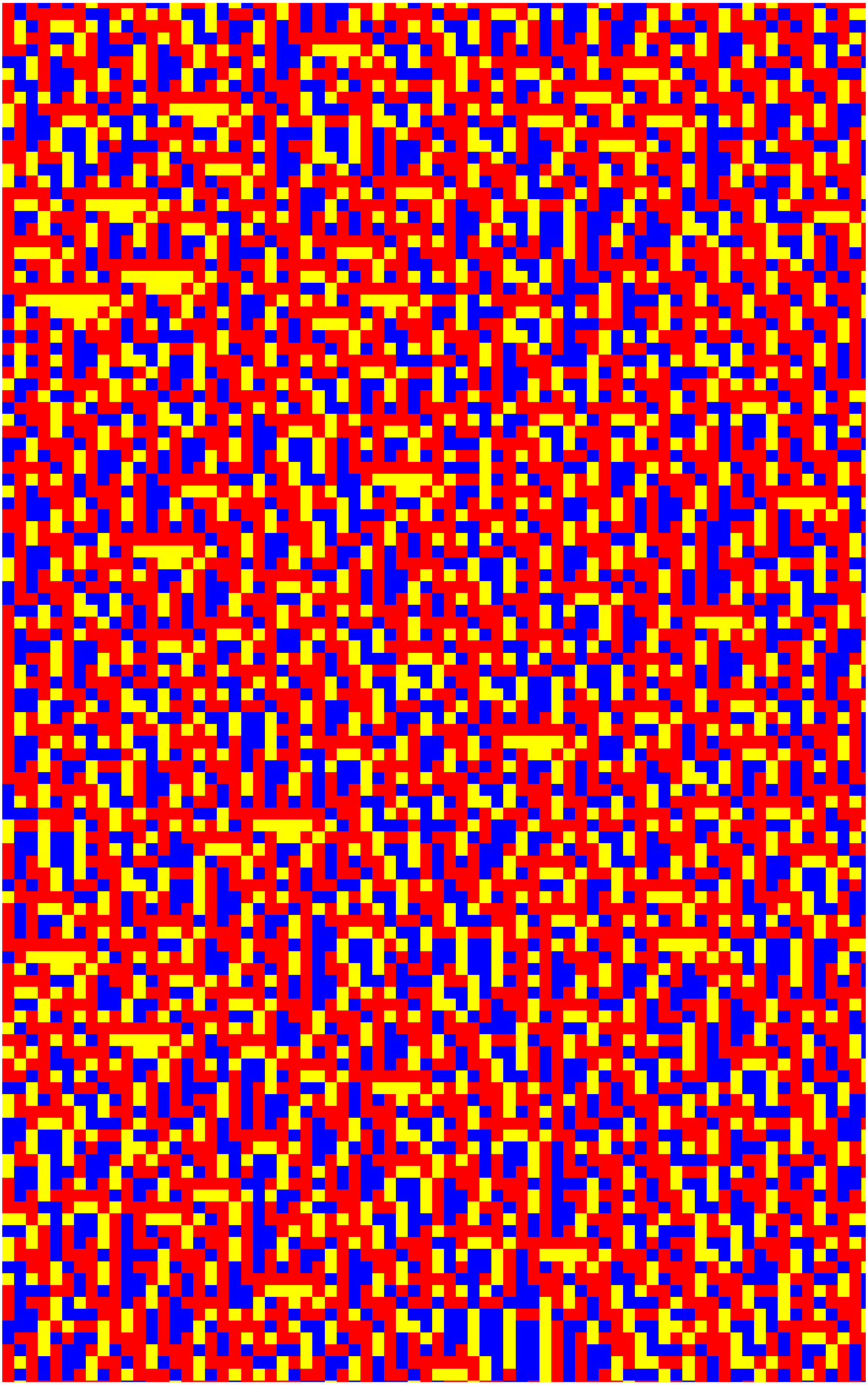


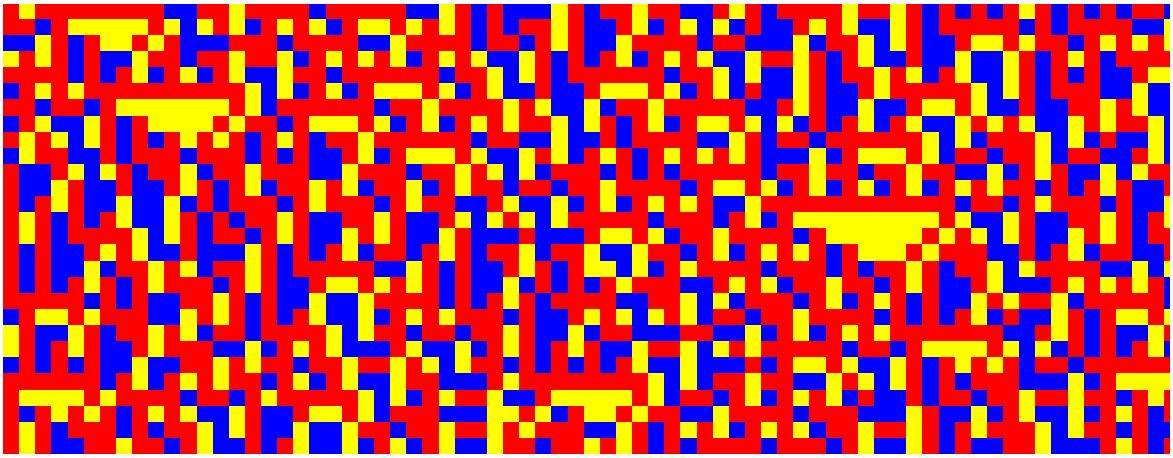


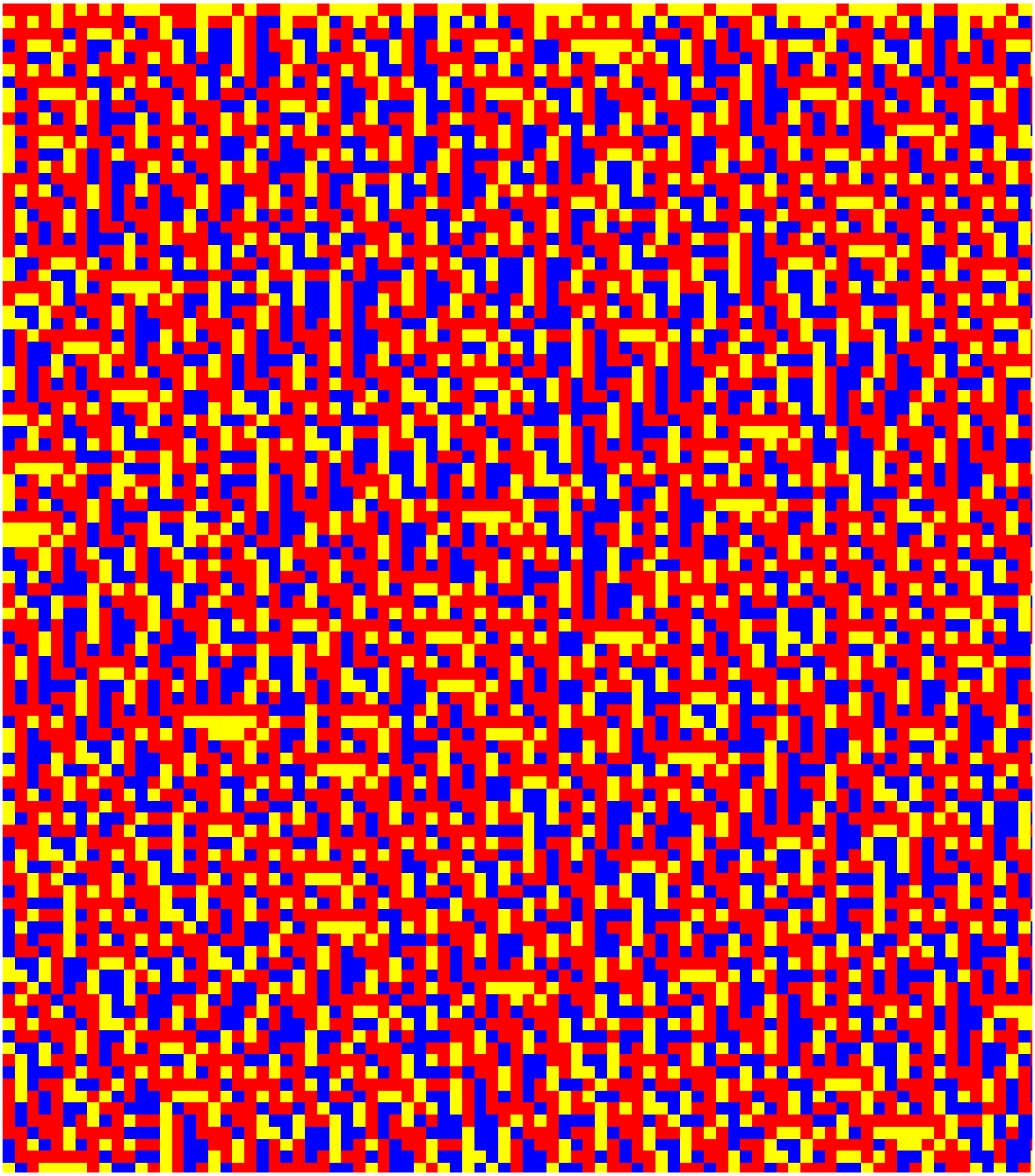


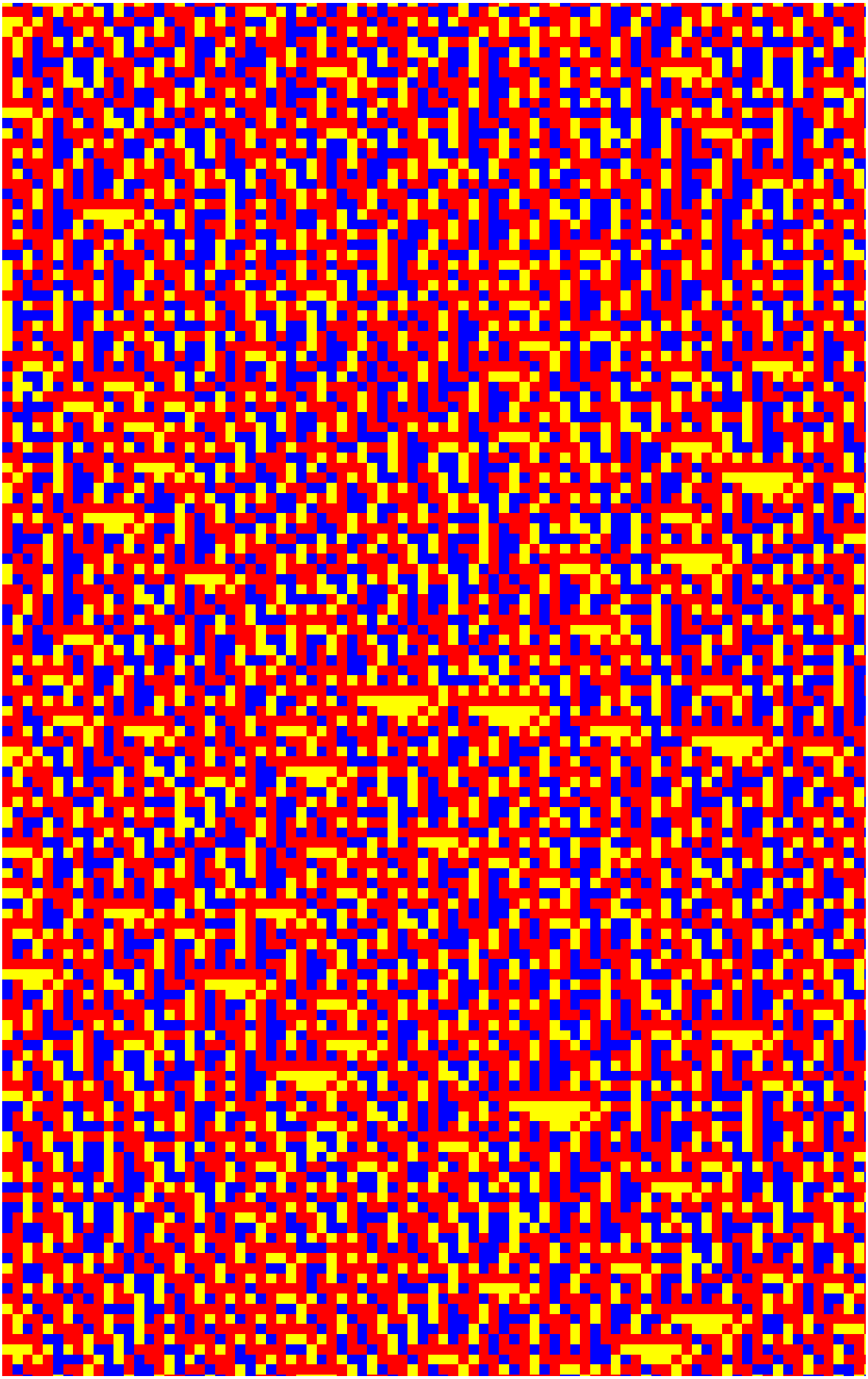


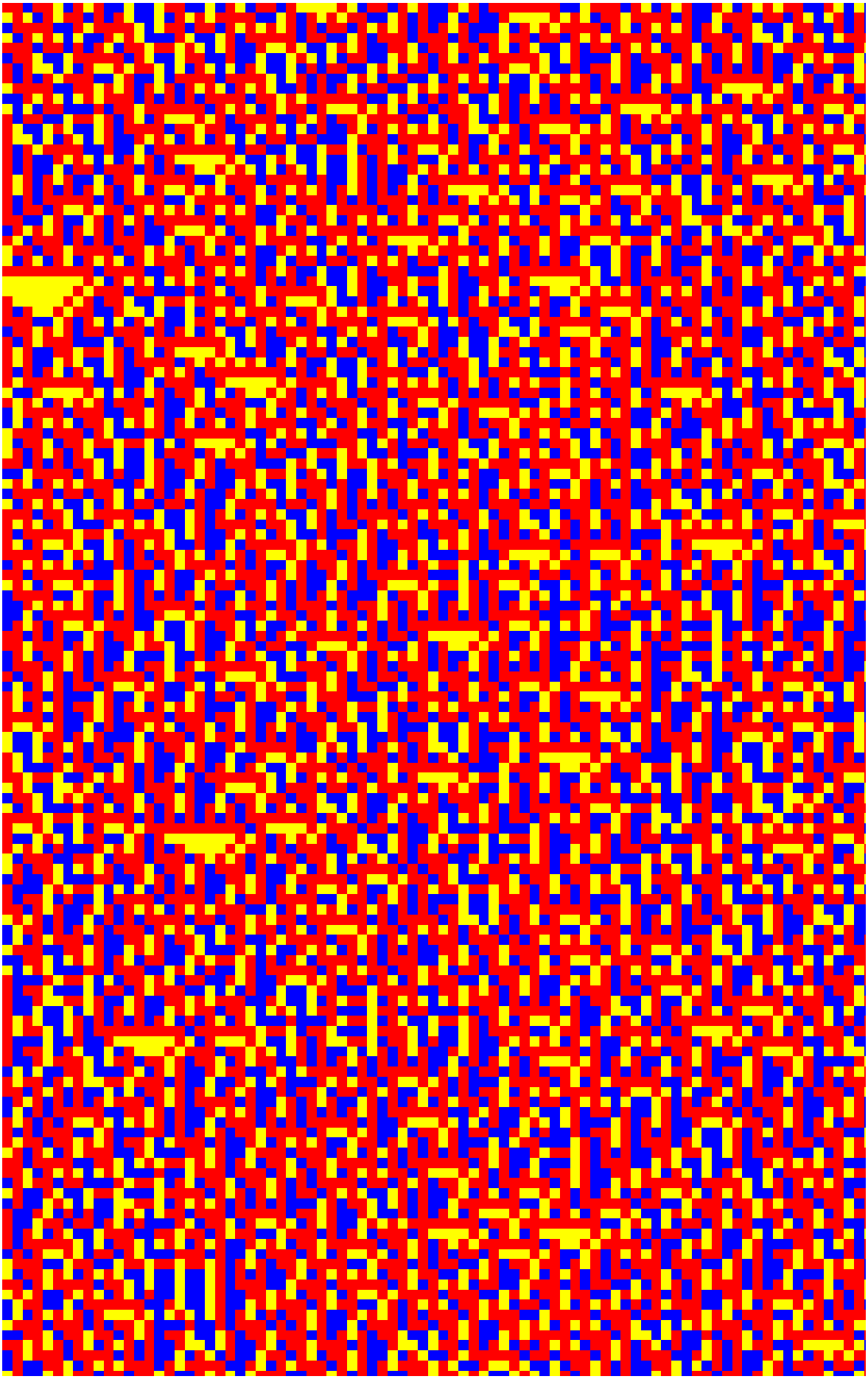


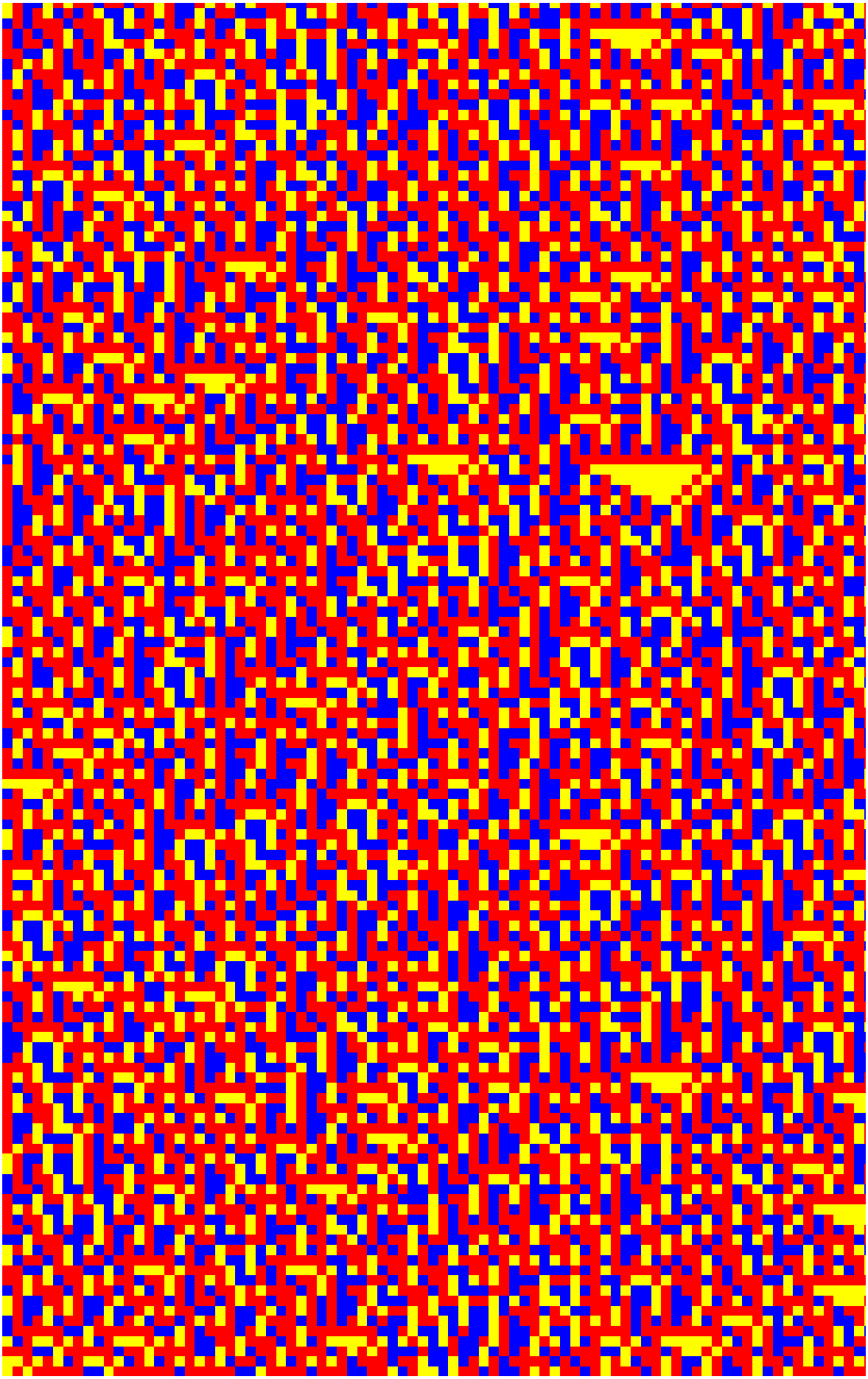


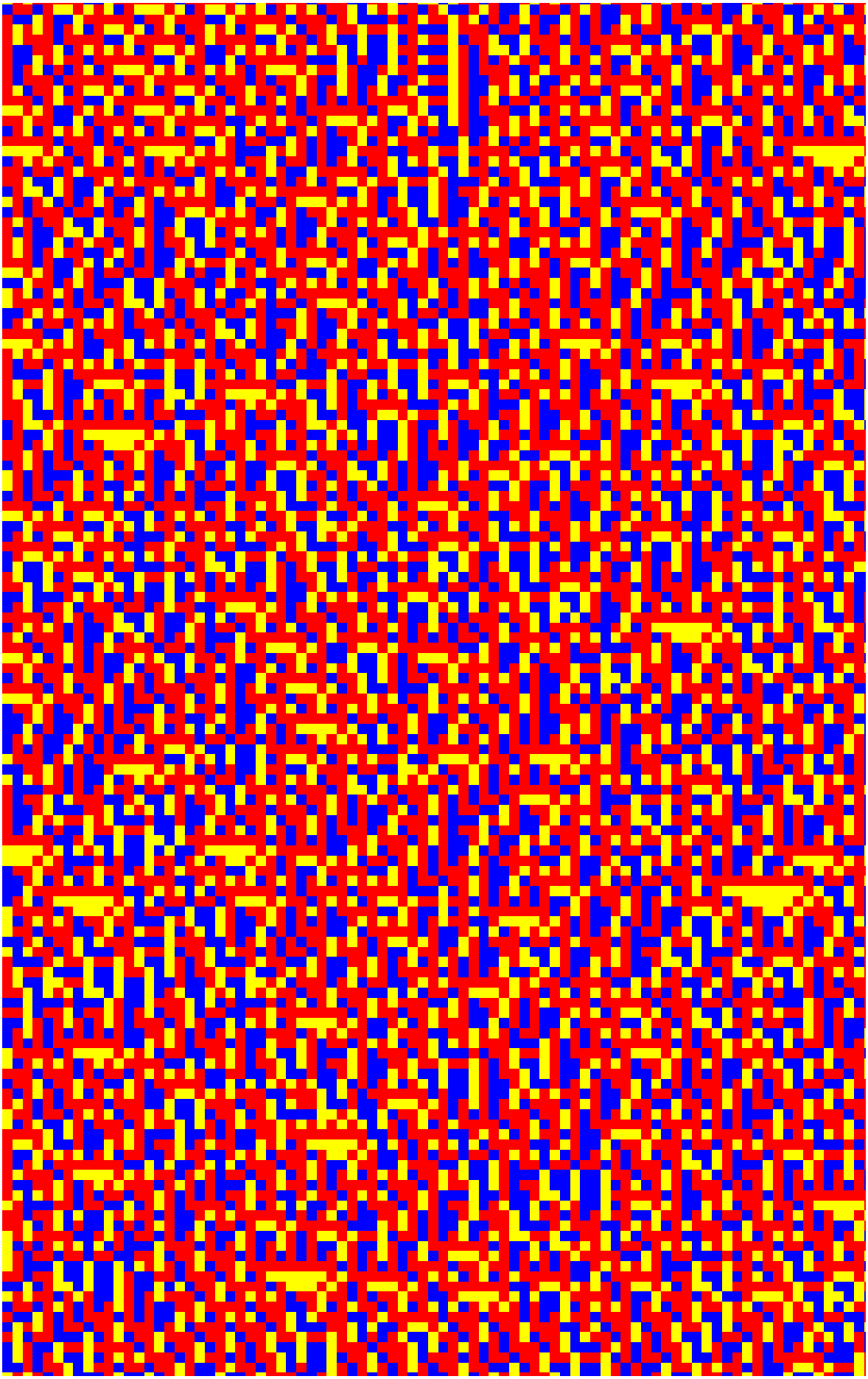


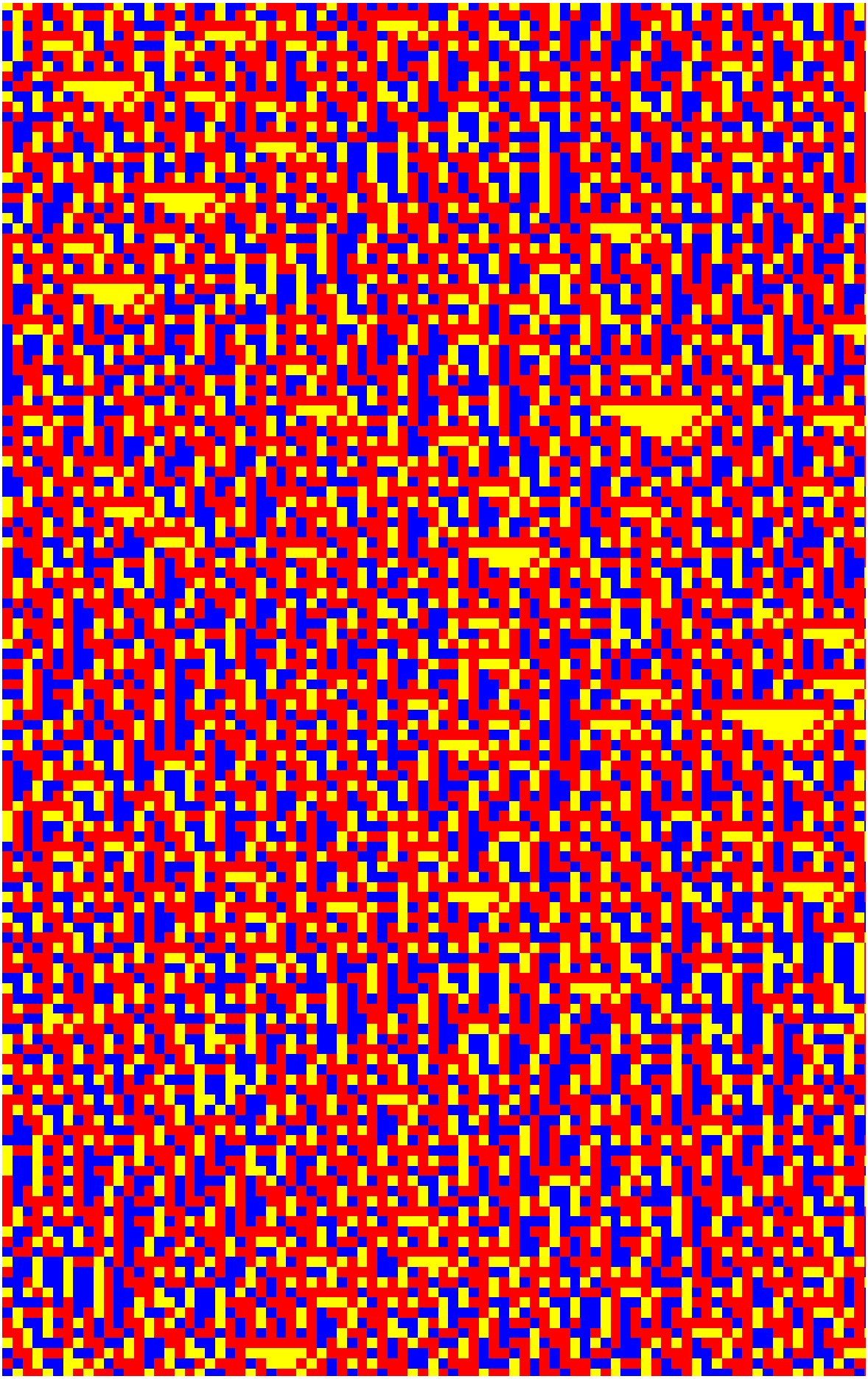


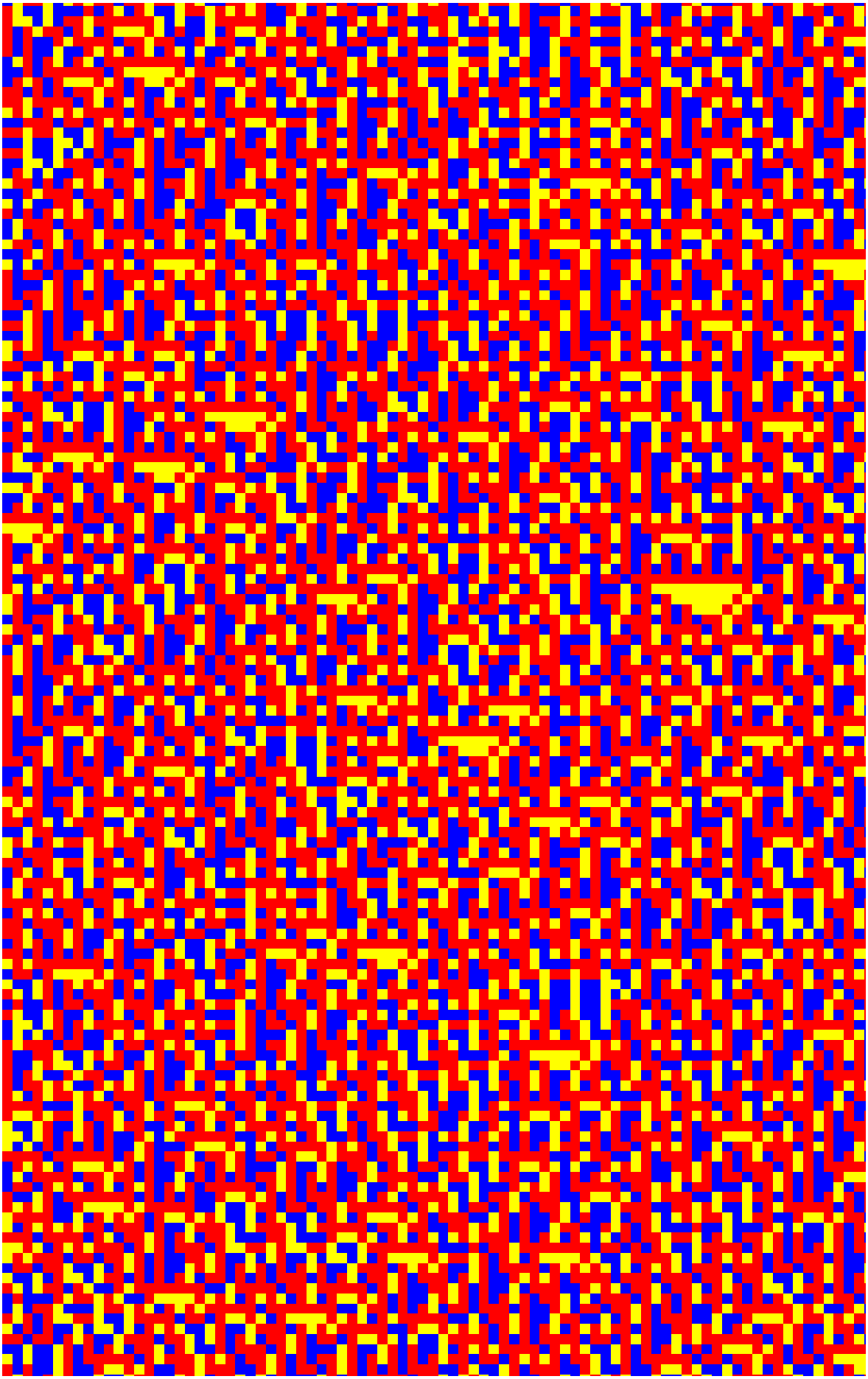


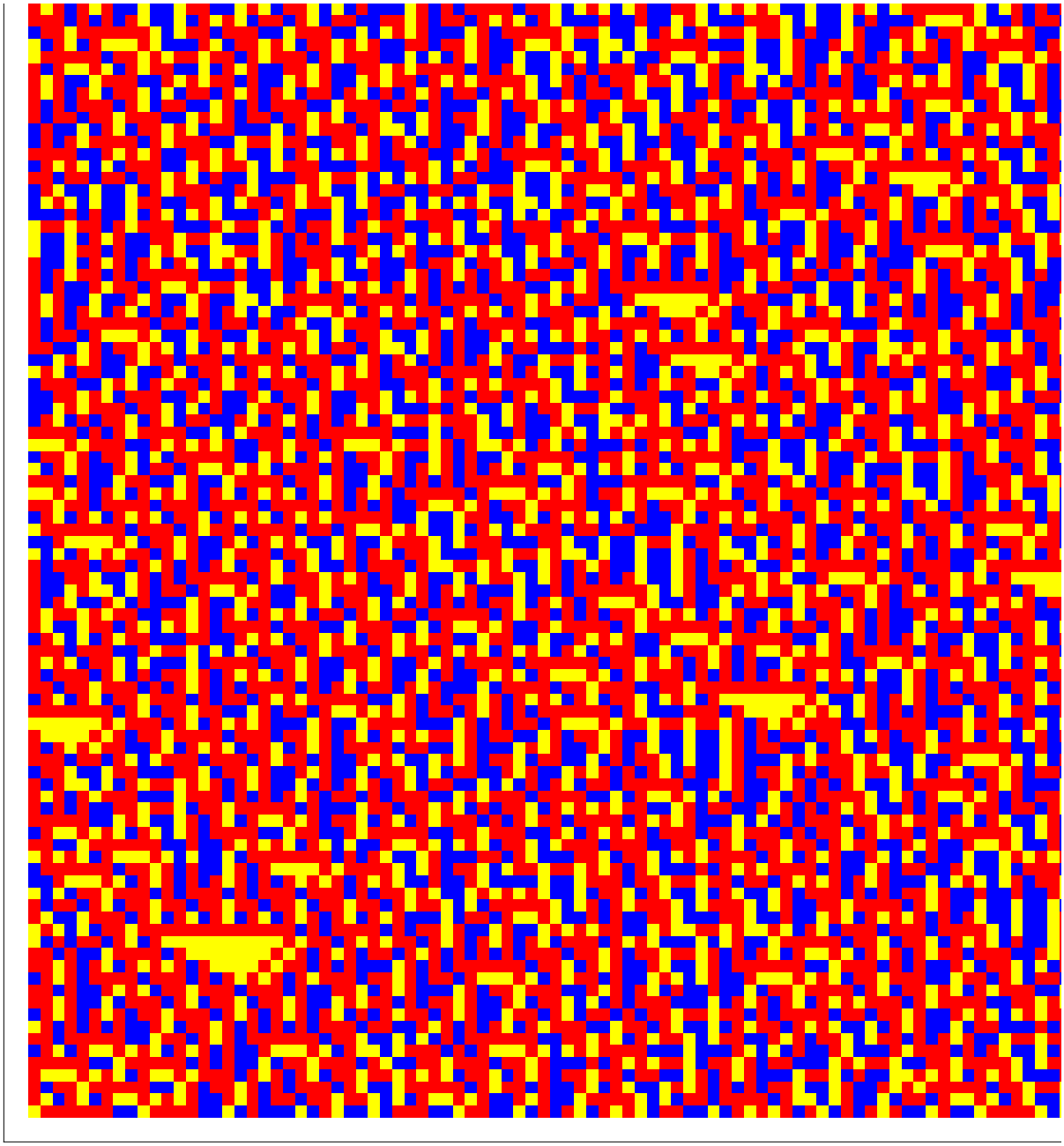


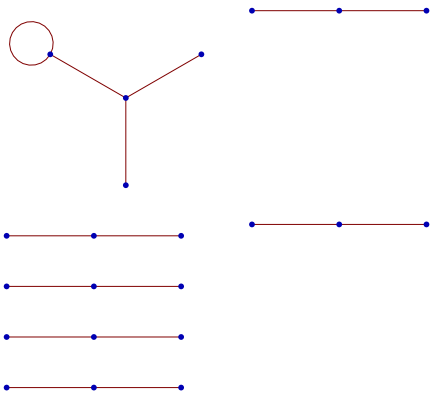


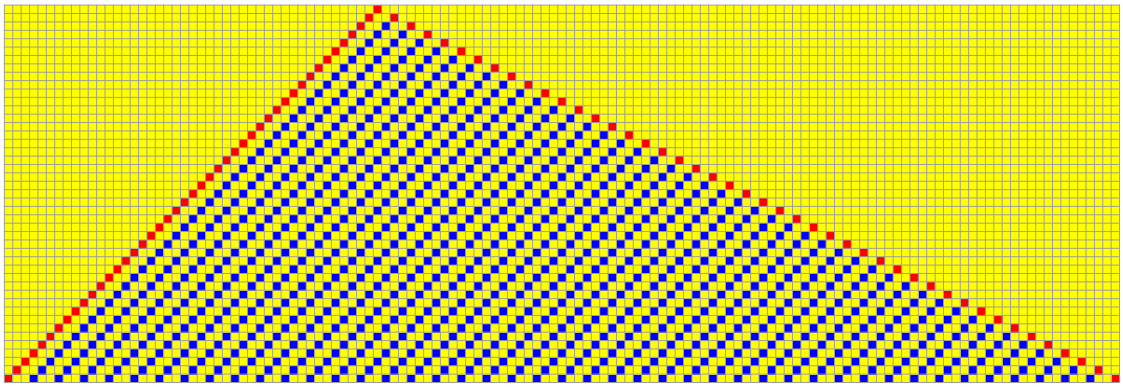












Catalog of $CA^{(5,4)}$ Examples

Catalog of $CA^{(5,5)}$ Examples

Decomposition

Tables

Dekomposits