— vordenker-archive —

Rudolf Kaehr
(1942-2016)

Title

Morphic Cellular Automata Systems – Part III: Indicational CAs

Archive-Number / Categories

3_38 / 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

As it is reasonable to define "*indicational*" stack machines it is equally reasonable to define the concept of indicational cellular automata. In both cases, the definitions relies on the primary 'semiotic' properties and not on the specific features of the calculus of indication, like the double crossing.
The same hold for all other definitions of CAs. The specific semiotic properties of digital CAs are not considered, what counts are the transition rules on semiotically identical signs. For kenomic CAs, it is the kenomic definition of the 'data' what is relevant. At least all this is correct at a first glance.
This study of *"Indicational CAs"* is focused on the proto-semiotic conditions of George Spencer Brown's *Calculus of Indication* as developed in his work *Laws of Form,* and not on the 'arithmetical' and 'algebraic' rules of the calculus of indication (CI) itself.
Hence, the 'arithmetical' rules of the process of distinctions, the law of *condensation* and the law of *cancellation* are not thematized in this context but their deep-structure of inscription, the '*topological invariance*' of the notation is chosen as the primary topic and studied in the framework of *cellular automata*.
This approach has not yet been studies properly in the literature about Spencer-Brown's calculus of indication.
With this CA-oriented study of the proto-semiotic conditions of the CI, the field of research of the deep-structure of the CI is opened up.
The interesting results of this study are presented as such, and shall not yet be thematized or interpreted by graphematic and grammatological considerations.
Earlier graphematic characterizations of the *Laws of Form* are helpful for an understanding of the presented approach but shall not be involved into the present considerations.

Citation Information / How to cite

Rudolf Kaehr: "Morphic Cellular Automata Systems – Part III: Indicational CAs", www.vordenker.de (Sommer Edition, 2017) J. Paul (Ed.), http://www.vordenker.de/rk/rk_Morphic-Cellular-Automata-Systems_Indicational-CA-part-3_2014.pdf

Categories of the RK-Archive

K01 Gotthard Günther Studies
K02 Scientific Essays
K03 Polycontexturality – Second-Order-Cybernetics
K04 Diamond Theory
K05 Interactivity
K06 Diamond Strategies
K07 Contextural Programming Paradigm
K08 Formal Systems in Polycontextural Constellations
K09 Morphogrammatics
K10 The Chinese Challenge or A Challenge for China
K11 Memristics Memristors Computation
K12 Cellular Automata
K13 RK and friends

Morphic Cellular Automata Systems

Part III: Indicational CAs,

Dr. phil Rudolf Kaehr

ISSN 2041-4358
(work in progress, vs. 0.2, July 2014)

Part III: Indicational CA

Motivation

As it is reasonable to define *"indicational"* stack machines it is equally reasonable to define the concept of indicational cellular automata. In both cases, the definitions relies on the primary 'semiotic' properties and not on the specific features of the calculus of indication, like the double crossing.
The same hold for all other definitions of CAs. The specific semiotic properties of digital CAs are not considered, what counts are the transition rules on semiotically identical signs. For kenomic CAs, it is the kenomic definition of the 'data' what is relevant. At least all this is correct at a first glance.

This study of *"Indicational CAs"* is focused on the proto-semiotic conditions of George Spencer Brown's *Calculus of Indication* as developed in his work *Laws of Form*, and not on the 'arithmetical' and 'algebraic' rules of the calculus of indication (CI) itself.

Hence, the 'arithmetical' rules of the process of distinctions, the law of *condensation* and the law of *cancellation* are not thematized in this context but their deep-structure of inscription, the *'topological invariance'* of the notation is chosen as the primary topic and studied in the framework of *cellular automata*.

This approach has not yet been studies properly in the literature about Spencer-Brown's calculus of indication.

With this CA-oriented study of the proto-semiotic conditions of the CI, the field of research of the deep-structure of the CI is opened up.

The interesting results of this study are presented as such, and shall not yet be thematized or interpreted by graphematic and grammatological considerations.

Earlier graphematic characterizations of the *Laws of Form* are helpful for an understanding of the presented approach but shall not be involved into the present considerations.

Commutativity of terms for indCA

"A third *deviation* from classical semiotics is less obvious: the commutativity of the concatenation operation. For any two terms "a" and "b" the terms "ab" and "ba" are identical. That this is indeed a semiotic identity (and not just a logical equality) has been stressed by Varga."

"We call two elements $x = (x_1, \dots, x_n)$, $y = (y_1, \dots, y_n) \in A^*$ equivalent, in short notation
(1) $x \approx_1 y$
if and only if there is a permutation $f: \{1, \dots, n\} \longrightarrow \{1, \dots, n\}$ such that
(2) $y_i = x_{f(i)}$ $i = 1, \dots, n$

We could now say that a "string" in the Brownian sense is an equivalence class of A* with respect to $\approx_1$, i.e., an element of the quotient set $A^* / \approx_1$.

(A) If the two given tokens of strings have different lengths, then they are different. If they have equal lengths, then go to (B').
(B') Check whether each atom appears equally often in both string-tokens. If this is the case, then they are equal,

otherwise they are different." (R. Matzka)

http://www.rudolf-matzka.de/dharma/semabs.pdf

http://www.thinkartlab.com/pkl/media/Diamond%20 Calculus/Diamond%20 Calculus.html
http://memristors.memristics.com/Complementary%20 Calculi/Complementary%20 Calculi.html

Toward indicational CAs

Topological invariance of the 'heads' of CA rules.

System of indicational CA rules

R1 ■ ■ ■ / – ■ –	R2 ■ ■ □ / – ■ –	R3 ■ □ □ / – ■ –	R4 □ □ □ / – ■ –
R5 ■ ■ ■ / – □ –	R6 ■ ■ □ / – □ –	R7 ■ □ □ / – □ –	R8 □ □ □ / – □ –

Indicational rules: (aa) $\neq_{ind}$ (bb), (ab) $=_{ind}$ (ba);

cardinality Ind $_{(n,\,m)}$ = $\binom{n+m-1}{n}$,

m = 2, n = 3 : $\binom{3+2-1}{3} = \binom{4}{3} = 4$

Alphabet = {a, b}
card (Ind(3, 2)) = {aaa, aab, abb, bbb}.

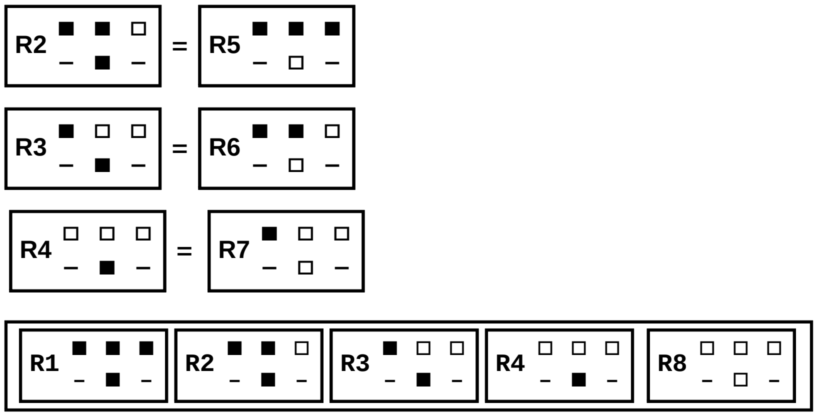
m = 2, n = 4
card (Ind (2, 4)) = 5
{aaaa, bbbb; aaab, abbb, aabb}.

Indicational graph (III)

```
          a
        ↙ ↓ ↘
      a   a   b
      a   b   b
    ↙ ↘ ↙ ↘ ↙ ↘
   a   a   a   b
   a   a   b   b
   a   b   b   b
 ↙ ↘ ↙ ↘ ↙ ↘ ↙ ↘
a   a   a   a   b
a   a   b   a   b
a   a   b   b   b
a   b   b   b   b
```

R1 ■ ■ ■ / – ■ –	R2 ■ ■ □ / – ■ –	R3 ■ □ □ / – ■ –	R4 □ □ □ / – ■ –
R5 ■ ■ ■ / – □ –	R6 ■ ■ □ / – □ –	R7 ■ □ □ / – □ –	R8 □ □ □ / – □ –

Alphabet = {a, b}
card(Ind(3, 2)) = {aaa, aab, abb, bbb} x {a,b}



Basic composition scheme

```
ruleCI[{a_, b_, c_, d_}] :=
    Flatten[{rca[{a}], rca[{b}], rca[{c}], rca[{d}]}]

ruleCIR[{a_, b_, c_, d_, e_}] :=
    Flatten[{cir[{a}], cir[{b}], cir[{c}], cir[{d}], cir[{e}]}]
```

Rule set for indCA$^{(3,2)}$

```
RuleSetCI = {
  rca[{1}] := {1, 1, 1} → 1,
  rca[{5}] := {1, 1, 1} → 0,

  rca[{4}] := {0, 0, 0} → 1,
  rca[{8}] := {0, 0, 0} → 0,

  rca[{3}] :=
   {
    {0, 0, 1} → 1,
    {0, 1, 0} → 1,
    {1, 0, 0} → 1
   },

  rca[{7}] :=
   {
    {0, 0, 1} → 0,
    {0, 1, 0} → 0,
    {1, 0, 0} → 0
   },

  rca[{2}] := {
    {1, 1, 0} → 1,
    {1, 0, 1} → 1,
    {0, 1, 1} → 1
   },

  rca[{6}] :=
   {
    {1, 1, 0} → 0,
    {1, 0, 1} → 0,
    {0, 1, 1} → 0
   }
 }
```

```
{Null, Null, Null, Null, Null, Null, Null, Null}
```

TabView for indCA$^{(3,2)}$ rules

1	5	4	8	3	7	2	6

```
rca[{6}]:=
   {
   {1,1,0}→0,
   {1,0,1}→0,
   {0,1,1}→0
  }
```

MenuView for indCA$^{(3,2)}$ rules

```
2 ∨
rca[{2}]:=
    {
  {1,1,0}→1,
  {1,0,1}→1,
  {0,1,1}→1
 }
```

Rule space for indCA$^{(3,2)}$

{1, 5} x {2, 6} x {3, 7} x {4, 8} = 16

1234	5234
1238	5238
1274	5274
1278	5278
1674	5634
1678	5638
1634	5674
1638	5678

Program scheme for indicational CA

```
ArrayPlot[CellularAutomaton[
  {
   CI - rules_i = perm_i[{a, b, c}] → d, a, b, c ∈ {0, 1}, 1 ≤ i ≥ 4
  },
 init, steps],
 Graphics = ColorRules -> {1 -> Red, 0 -> Yellow}]
```

Indicational normal form (inf)

inf([■ □ ■]) = inf([□ ■ ■])= [■ ■ □])
inf([□ ■ □]) = inf([□ □ ■]) = [■ □ □)]
inf([■ ■ ■]) = [■ ■ ■]
inf([□ □ □]) = [□ □ □]

Indicational normal form (inf):
inf([■ □ ■]) = inf([□ ■ ■])= [■ ■ □])
inf([□ ■ □]) = inf([□ □ ■]) = [■ □ □)]
inf([■ ■ ■]) = [■ ■ ■]
inf([□ □ □]) = [□ □ □]

Indicational normal forms are not identical with *proto*-normal forms, *pnf*, of kenogrammatics.

Calculus of indication: a=a, a≠b, ab=ba

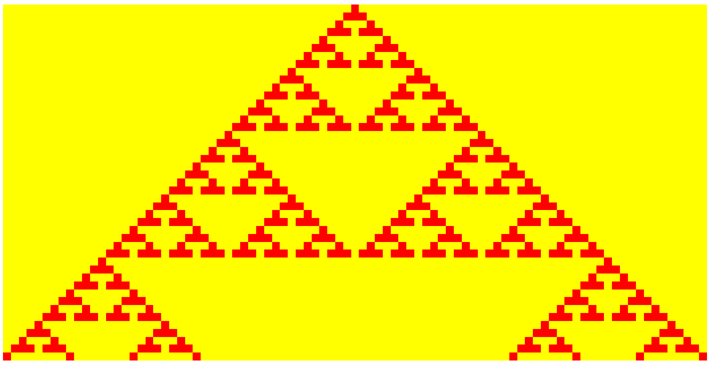
For 1D indCA$^{(2,2)}$:
[aab] = [baa] = [aba]

[abb] = [baa] = [bab]
[aaa] = [aaa]
[bbb] = [bbb]

pnf[([aaa]) = pnf([bbb]) but
inf[([aaa]) ≠ inf([bbb]).

System of indCA$^{(3,2)}$

14



```
(*
Manipulate[ListAnimate[Table[ArrayPlot[CellularAutomaton[ruleCI,{{1},0},n],
     ColorRules->{1->Red,0->Yellow,2→ Blue, 3→ Green},ImageSize→{400,200}],
    {ruleCI,
     {
      ruleCI[{1,2,3,4}],
      ruleCI[{1,2,3,8}],
      ruleCI[{1,2,7,4}],
      ruleCI[{1,2,7,8}],

      ruleCI[{1,6,3,4}],
      ruleCI[{1,6,3,8}],
      ruleCI[{1,6,7,4}],
      ruleCI[{1,6,7,8}],

      ruleCI[{5,2,3,4}],
      ruleCI[{5,2,3,8}],
      ruleCI[{5,2,7,4}],
      ruleCI[{5,2,7,8}],

      ruleCI[{5,6,3,4}],
      ruleCI[{5,6,3,8}],
      ruleCI[{5,6,7,4}],
      ruleCI[{5,6,7,8}]

     }}]],{n,2,222,1}]
*)
```

Analysis of linear forms

Groups

```
1234
1634

5678
1278
```

```
5234
5634

5674
1678
```

```
1274
```

Equalities

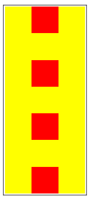
```
5234 = 5634
1234 = 1634
5678 = 1278
```

```
1678 ≈ 5674
```

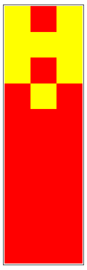
```
1274
```

Examples

rule = 5.6.7.4



rule = 1274

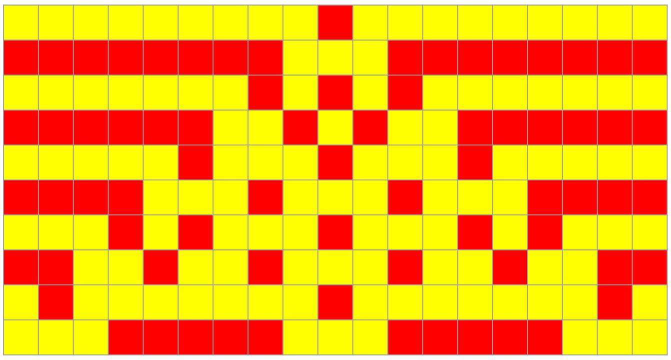


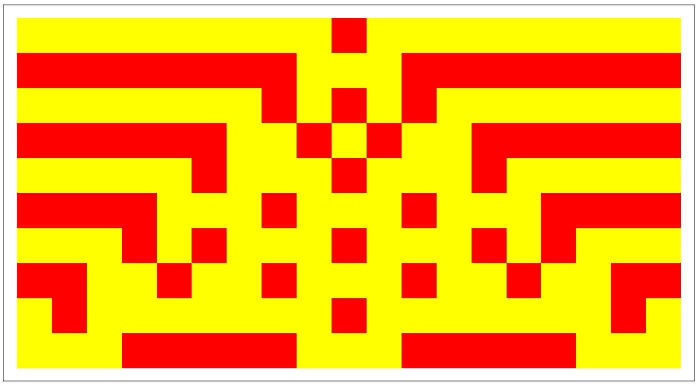
Analysis of developping indCA forms

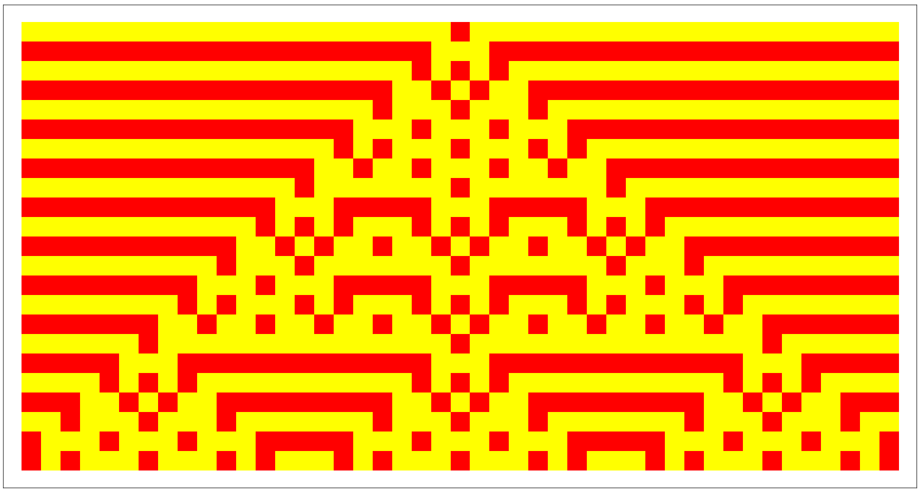
```
1634
1638
```

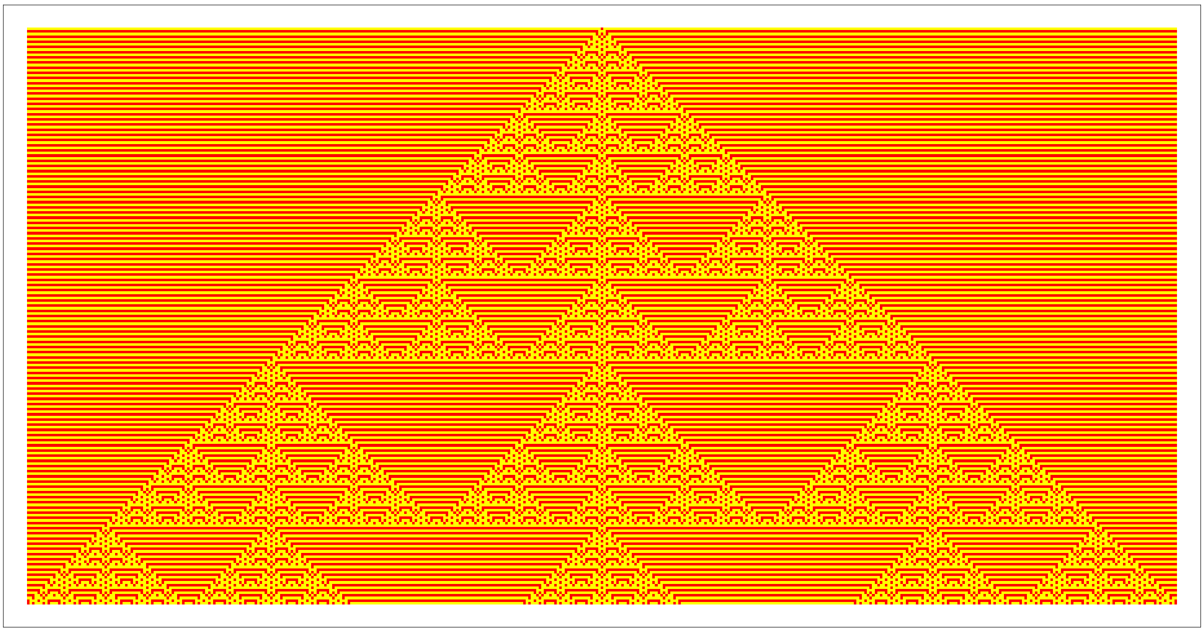
```
5638
5238
```

rule = 1634

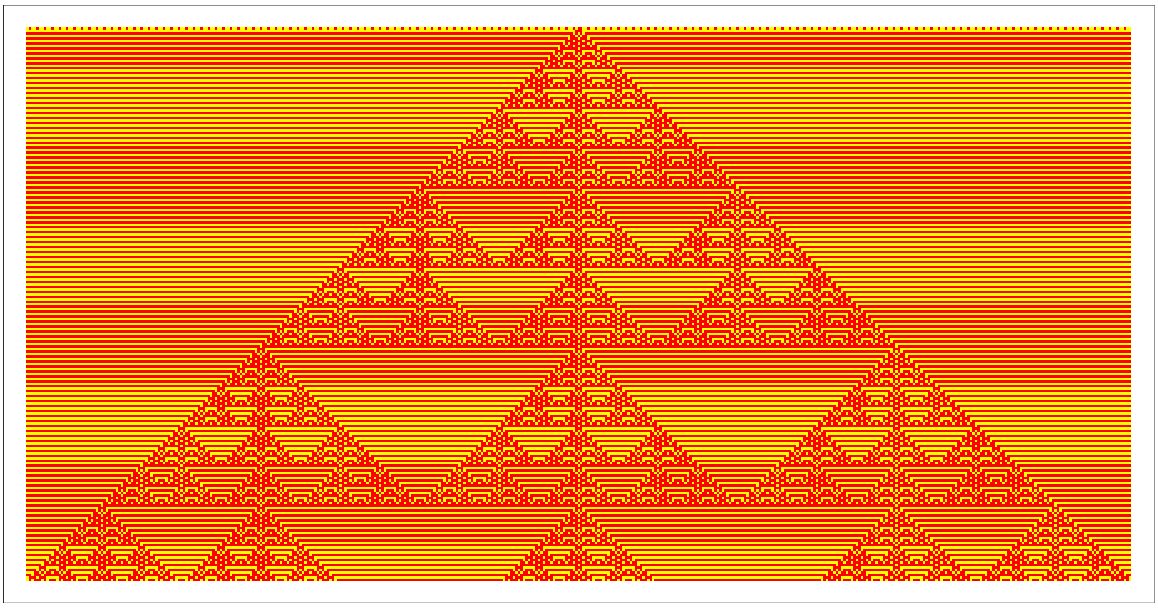




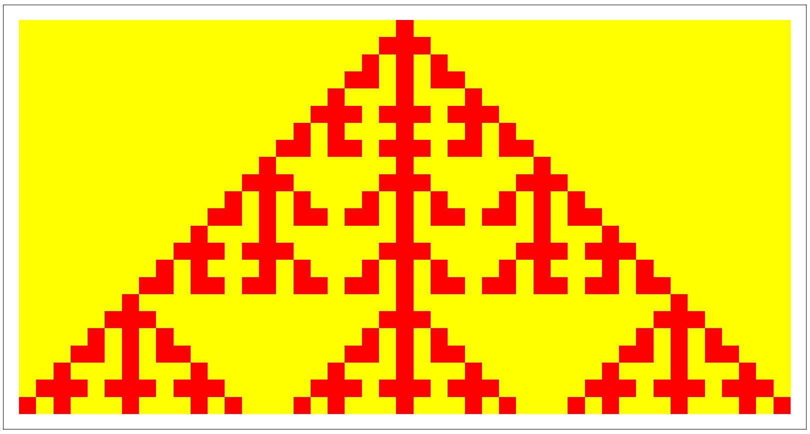


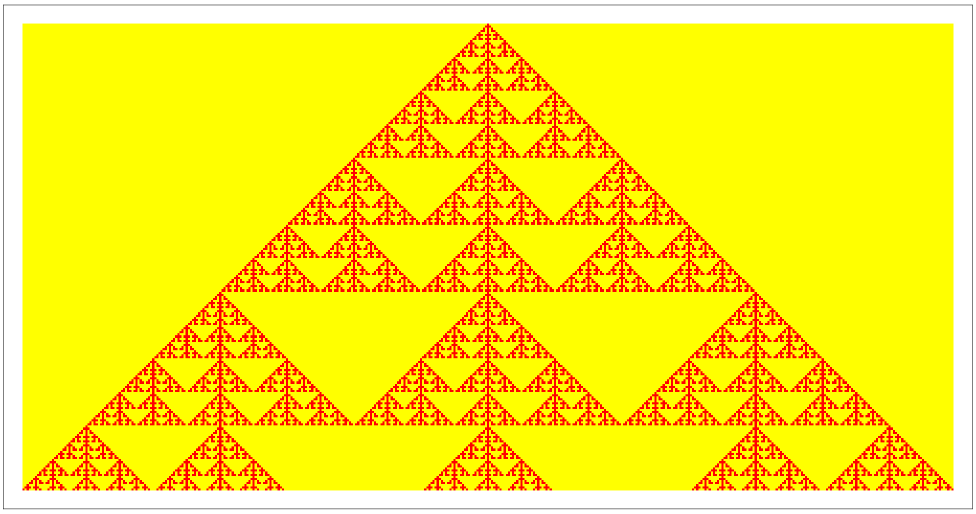


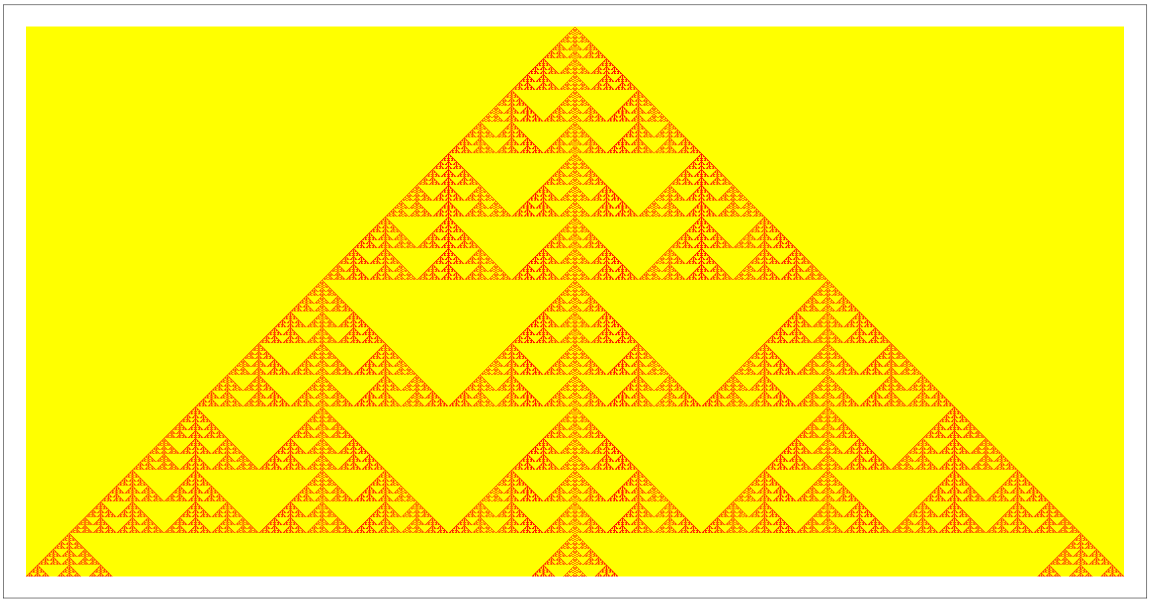
seed = {0, 1, 0}



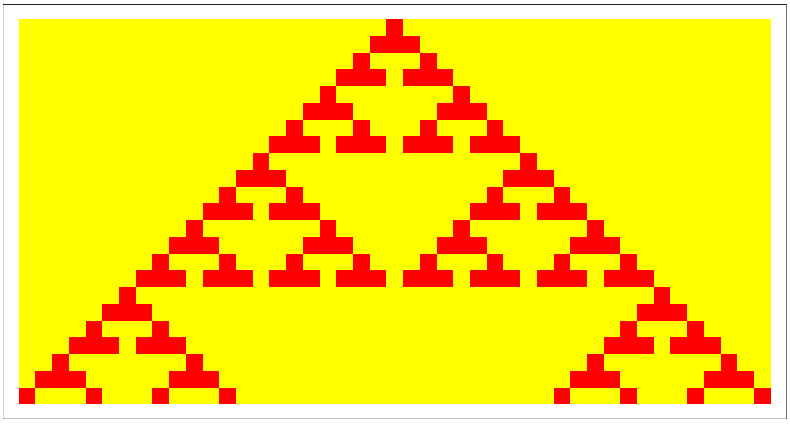
rule = 1638

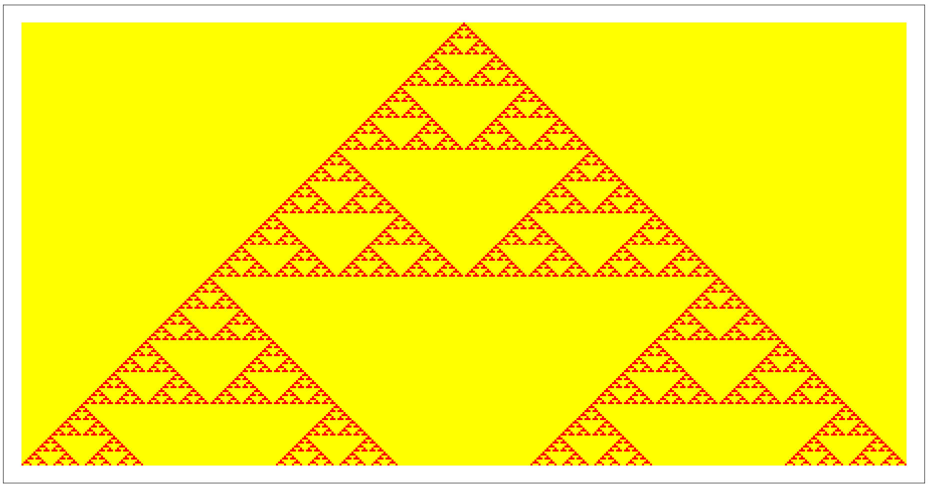


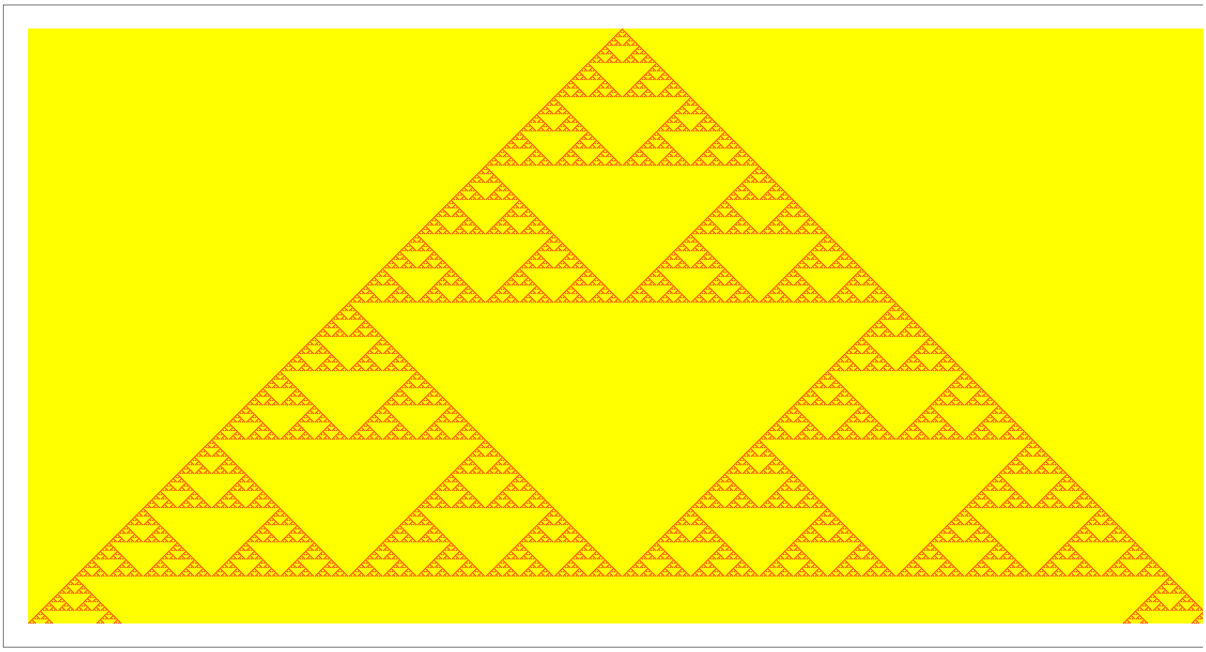




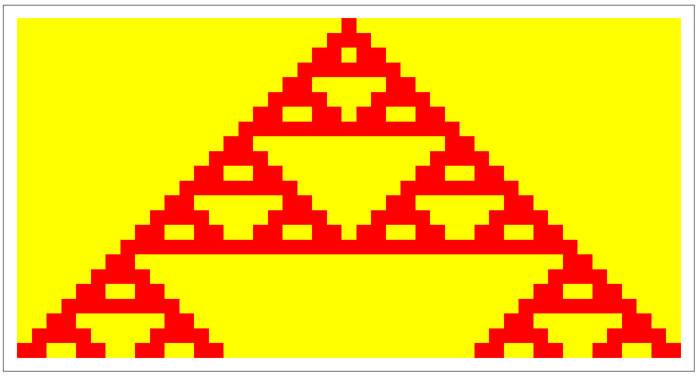
CI - rule = 5.6 .3 .8

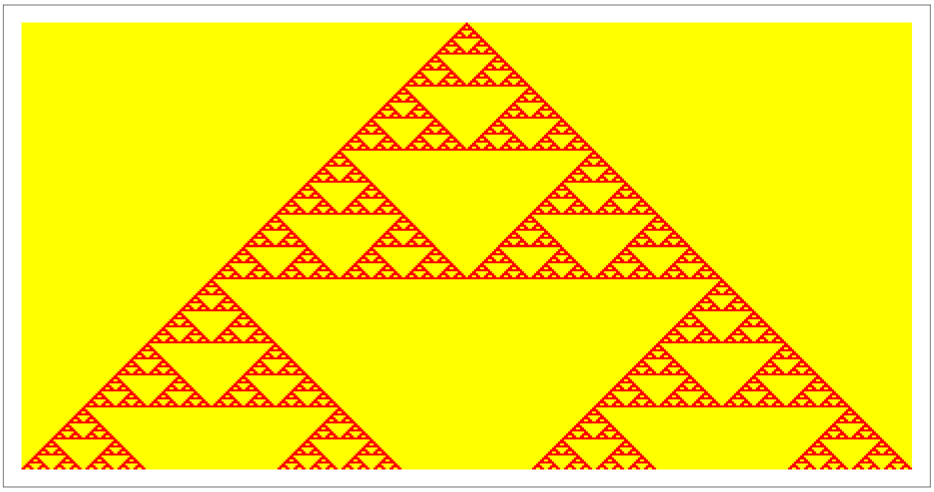


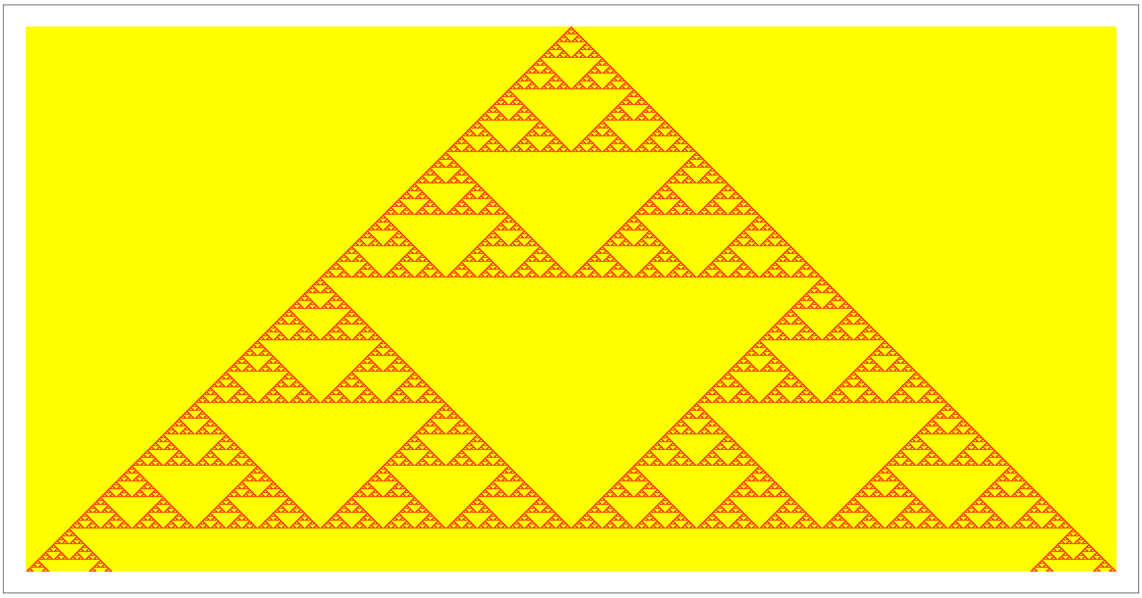




CI - rule = 5.2 .3.8

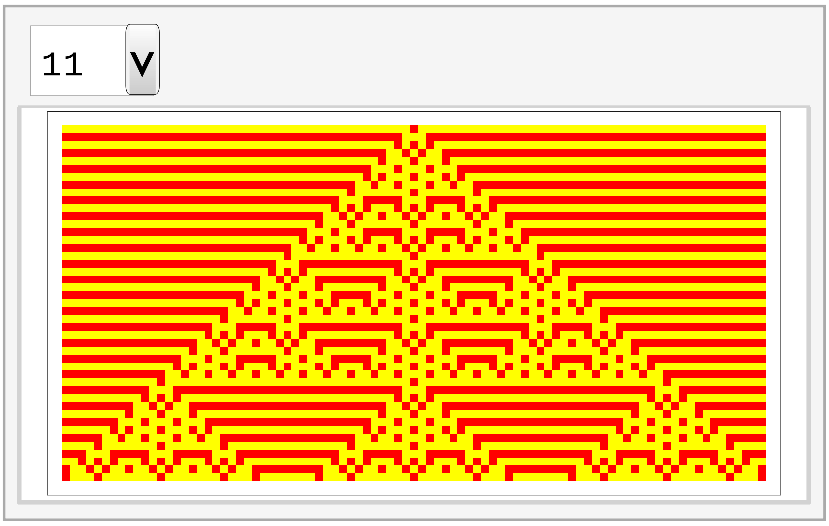




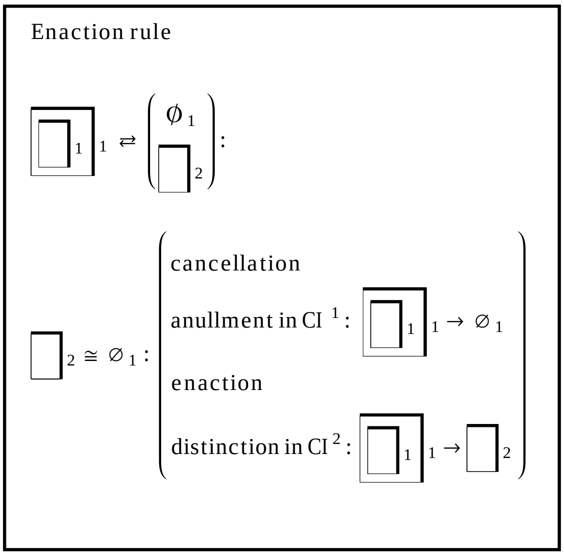


System of indCA$^{(4,2)}$ rules

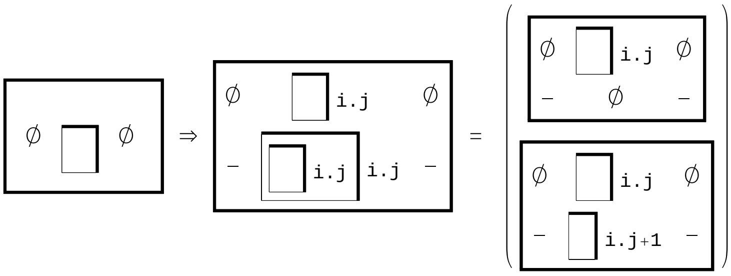
```
(*
Manipulate[MenuView[Table[ArrayPlot[CellularAutomaton[ruleCI,{{1},0},n],
    ColorRules->{1->Red,0->Yellow,2→ Blue, 3→ Green},ImageSize→{400,200}],
   {ruleCI,
    {
     ruleCI[{1,2,3,4}],
     ruleCI[{1,2,3,8}],
     ruleCI[{1,2,7,4}],
     ruleCI[{1,2,7,8}],

     ruleCI[{1,6,3,4}],
     ruleCI[{1,6,3,8}],
     ruleCI[{1,6,7,4}],
     ruleCI[{1,6,7,8}],

     ruleCI[{5,2,3,4}],
     ruleCI[{5,2,3,8}],
     ruleCI[{5,2,7,4}],
     ruleCI[{5,2,7,8}],

     ruleCI[{5,6,3,4}],
     ruleCI[{5,6,3,8}],
     ruleCI[{5,6,7,4}],
     ruleCI[{5,6,7,8}]

    }}]],{n,2,222,1}]
*)
```

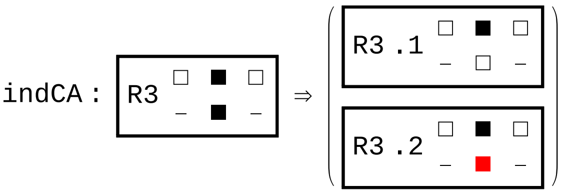


Reflectional enaction for indCA$^{(3,3)}$ rules



Example for indCA enaction





with the function ■ (■) = $\binom{\square}{\blacksquare}$.

Hence, the set of indCA rules might be extended by its interactional enaction rules which are relating to CA systems of neighbor contextures. This will be studied in another paper.

Combinatorics

***m* = *n* = 3**
card(Ind(3, 3)) = 10
Ind(3, 3) = {aaa, bbb, ccc, aab, aac, abb, acc, bcc, bbc, abc}

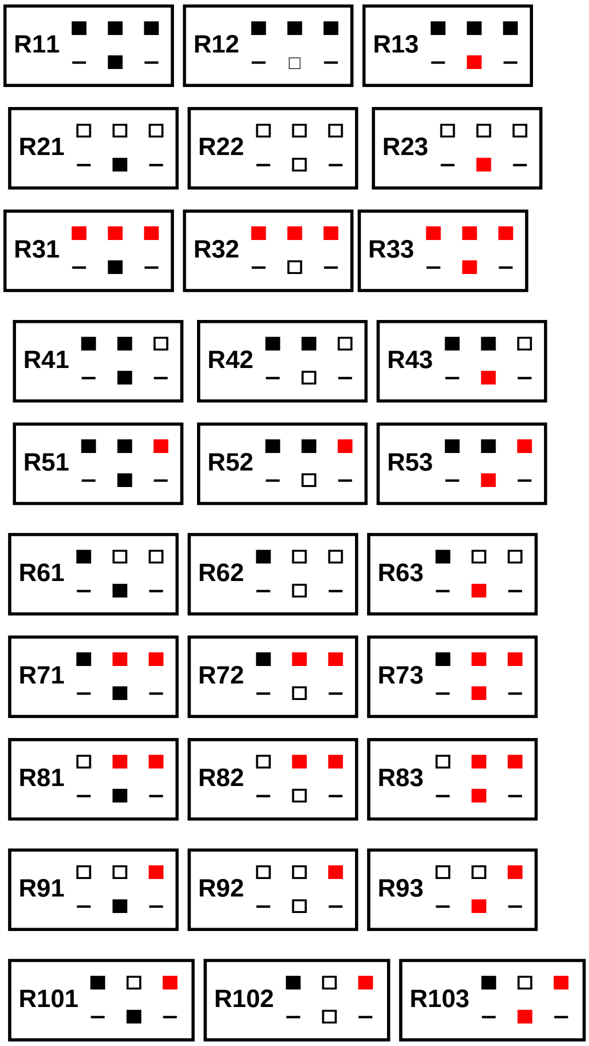
card (Ind (4, 3)) = card(Ind(3, 3)) x {3} = 30

$$m = 4,\ n = 3 : \binom{4+3-1}{4} = \binom{6}{4} = 30$$

Set of combinations for indCA$^{(4,3)}$:

{aaa, bbb, ccc, aab, aac, abb, acc, bcc, bbc, abc} x {a, b, c} =

Example



Rule scheme

```
ruleCIR[{a_, b_, c_, d_, e_, f_, g_, h_, i_, k_}] :=
    Flatten[
  {cir[{a}], cir[{b}], cir[{c}], cir[{d}],
   cir[{e}], cir[{f}], cir[{g}],
   cir[{h}], cir[{i}], cir[{k}]}]
```

Rule set for indCA$^{(3,3)}$

```
ruleSetCIR =
 {
 cir[{11}] := {1, 1, 1} → 1,
  cir[{21}] := {0, 0, 0} → 1,
  cir[{31}] := {2, 2, 2} → 1,

  cir[{12}] := {1, 1, 1} → 0,
 cir[{22}] := {0, 0, 0} → 0,
  cir[{32}] := {2, 2, 2} → 0,

  cir[{13}] := {1, 1, 1} → 2,
 cir[{23}] := {0, 0, 0} → 2,
  cir[{33}] := {2, 2, 2} → 2,

  cir[{41}] := {{1, 1, 0} → 1, {1, 0, 1} → 1, {0, 1, 1} → 1},
  cir[{51}] := {{1, 1, 2} → 1, {1, 2, 1} → 1, {2, 1, 1} → 1},
  cir[{61}] := {{1, 0, 0} → 1, {0, 1, 0} → 1, {0, 0, 1} → 1},

  cir[{42}] := {{1, 1, 0} → 0, {1, 0, 1} → 0, {0, 1, 1} → 0},
  cir[{52}] := {{1, 1, 2} → 0, {1, 2, 1} → 0, {2, 1, 1} → 0},
  cir[{62}] := {{1, 0, 0} → 0, {0, 1, 0} → 0, {0, 0, 1} → 0},

  cir[{43}] := {{1, 1, 0} → 2, {1, 0, 1} → 2, {0, 1, 1} → 2},
  cir[{53}] := {{1, 1, 2} → 2, {1, 2, 1} → 2, {2, 1, 1} → 2},
  cir[{63}] := {{1, 0, 0} → 2, {0, 1, 0} → 2, {0, 0, 1} → 2},

  cir[{71}] := {{1, 2, 2} → 1, {2, 1, 2} → 1, {2, 2, 1} → 1},

  cir[{81}] := {{0, 2, 2} → 1, {2, 0, 2} → 1, {2, 2, 0} → 1},
  cir[{91}] := {{0, 0, 2} → 1, {0, 2, 0} → 1, {2, 0, 0} → 1},

  cir[{72}] := {{1, 2, 2} → 0, {2, 1, 2} → 0, {2, 2, 1} → 0},
  cir[{82}] := {{0, 2, 2} → 0, {2, 0, 2} → 0, {2, 2, 0} → 0},
  cir[{92}] := {{0, 0, 2} → 0, {0, 2, 0} → 0, {2, 0, 0} → 0},

  cir[{73}] := {{1, 2, 2} → 2, {2, 1, 2} → 2, {2, 2, 1} → 1},

  cir[{83}] := {{0, 2, 2} → 2, {2, 0, 2} → 2, {2, 2, 0} → 2},
  cir[{93}] := {{0, 0, 2} → 2, {0, 2, 0} → 2, {2, 0, 0} → 2},

  cir[{101}] := {{1, 0, 2} → 1, {1, 2, 0} → 1, {2, 0, 1} → 1,
                 {2, 1, 0} → 1, {0, 1, 2} → 1, {0, 2, 1} → 1},

  cir[{102}] := {{1, 0, 2} → 0, {1, 2, 0} → 0, {2, 0, 1} → 0,
                 {2, 1, 0} → 0, {0, 1, 2} → 0, {0, 2, 1} → 0},

  cir[{103}] := {{1, 0, 2} → 2, {1, 2, 0} → 2, {2, 0, 1} → 2,
                 {2, 1, 0} → 2, {0, 1, 2} → 2, {0, 2, 1} → 2}
 }
```

```
72 ∨
cir[{72}]:={{1,2,2}→0, {2,1,2}→0, {2,2,1}→0}
```

indCA$^{(3,3)}$ Rule scheme *ruleCIR*, 10x3

```
{0, 1, 1} → 2,
{1, 0, 1} → 2,
{1, 1, 0} → 2,

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

{0, 0, 2} → 2,
{0, 2, 0} → 2,
{2, 0, 0} → 2,
```

```
{1, 2, 1} → 2,
{1, 1, 2} → 2,
{2, 1, 1} → 2,

{2, 1, 2} → 0,
{2, 2, 1} → 0,
{1, 2, 2} → 0,

{2, 0, 2} → 0,
{0, 2, 2} → 0,
{2, 2, 0} → 0
```

```
{1, 1, 1} → 1,
{2, 2, 2} → 2,
{0, 0, 0} → 0,

{2, 0, 1} → 1,
{2, 1, 0} → 1,
{1, 0, 2} → 1,
{1, 2, 0} → 1,
{0, 1, 2} → 1,
{0, 2, 1} → 1
```

CA Program scheme

```
ArrayPlot[CellularAutomaton[
  {
   {CIR – Rules}
  },
 init, steps],
  ColorRules -> {2 → Green, 1 -> Red, 0 -> Yellow}]
```

Rule space for indCA$^{(3,2)}$

{1, 5} x {2, 6} x {3, 7} x {4, 8} = 16

1234 – 8'	5234
1238	5238
1274	5274
1278	5278
1674	5634
1678	5638
1634	5674
1638	5678 – 8'

Indicational normal form (inf)

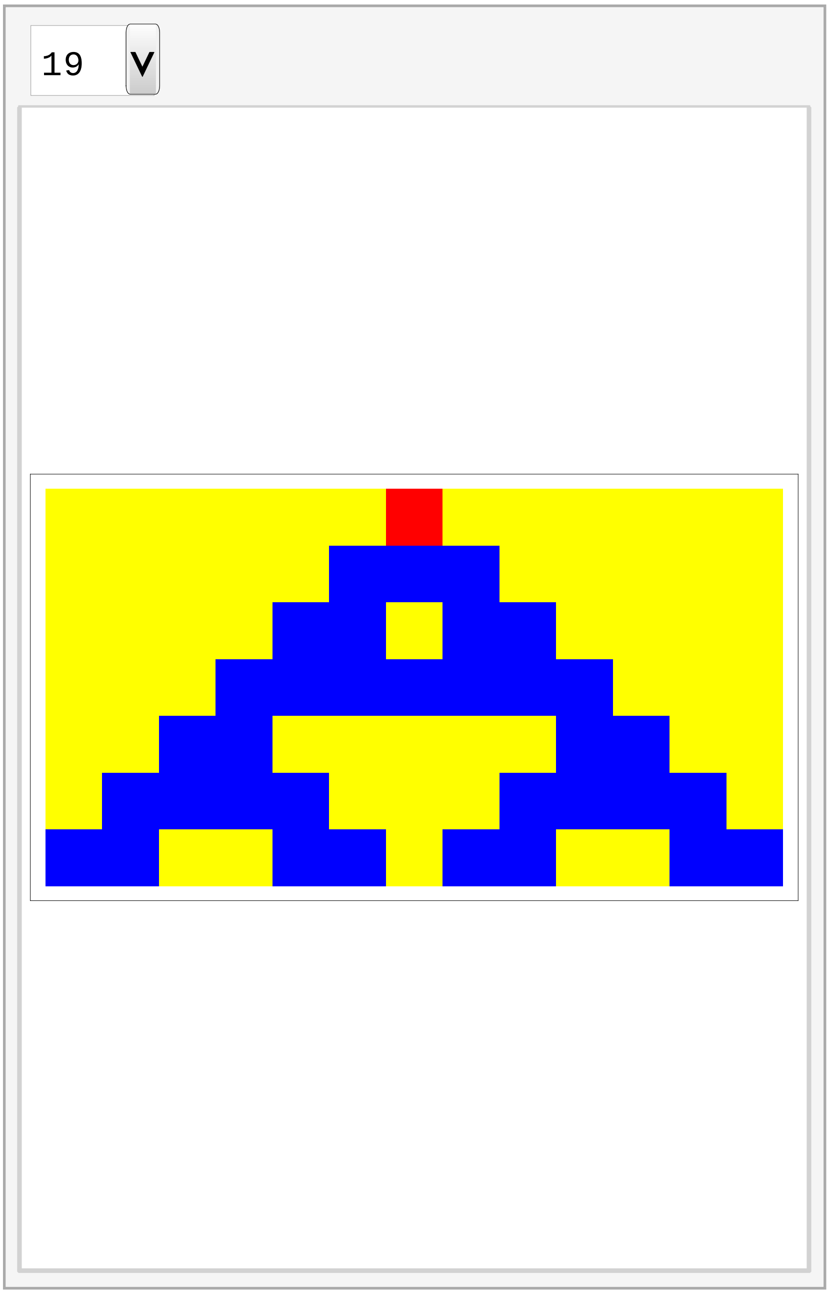
inf([□ ■ □]) = inf([□ □ ■]) = [■ □ □)]
inf([□ 🟥 □]) = inf([□ □ 🟥]) = [🟥 □ □)]
inf([■ 🟥 ■]) = inf([■ ■ 🟥]) = [🟥 ■ ■)]

inf([■ □ ■]) = inf([□ ■ ■]) = [■ ■ □])
inf([🟥 □ 🟥]) = inf([□ 🟥 🟥]) = [🟥 🟥 □])
inf([🟥 ■ 🟥]) = inf([■ 🟥 🟥]) = [🟥 🟥 ■])

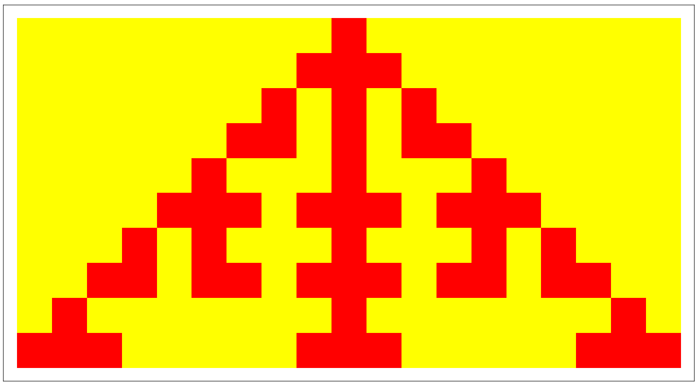
inf([■ ■ ■]) = [■ ■ ■]
inf([□ □ □]) = [□ □ □]
inf([🟥 🟥 🟥]) = [🟥 🟥 🟥]

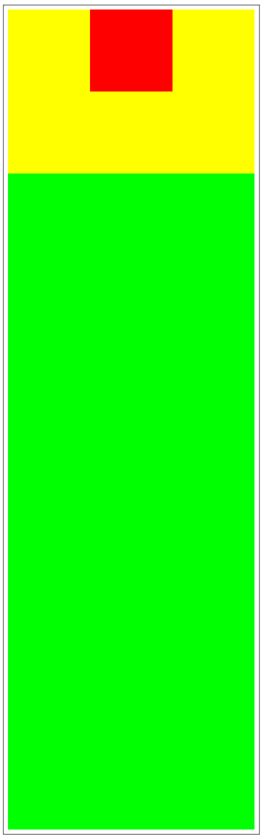
perm [{■, □, 🟥}]

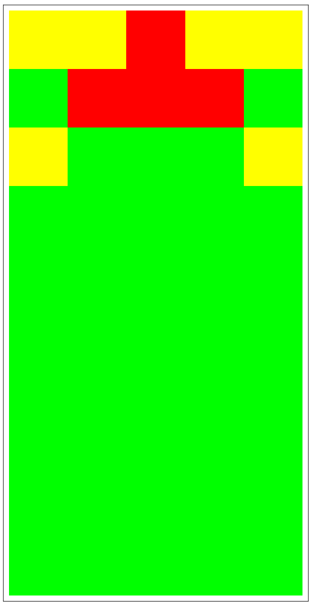
IndCA$^{(3,3)}$

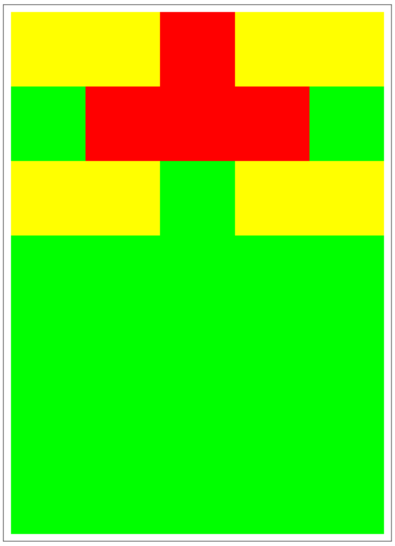


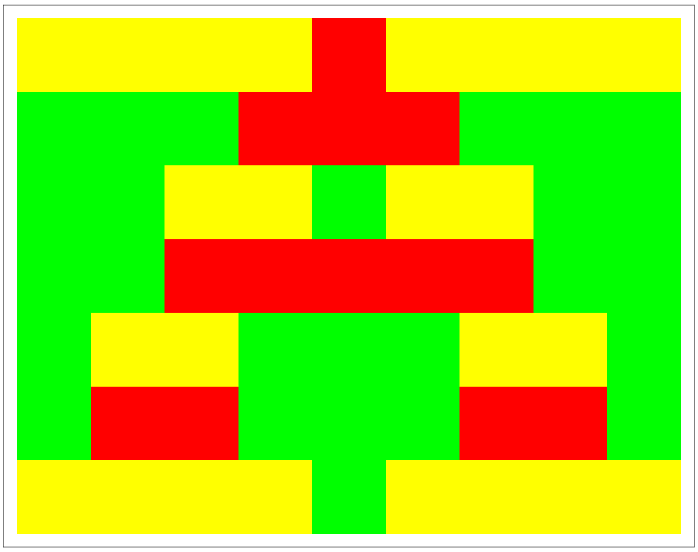

```
ruleCIR[{11, 22, 32, 42, 51, 63, 73, 83, 93, 102}]

{{1, 1, 1} → 1, {0, 0, 0} → 0, {2, 2, 2} → 0, {1, 1, 0} → 0, {1, 0, 1} → 0, {0, 1, 1} → 0,
 {1, 1, 2} → 1, {1, 2, 1} → 1, {2, 1, 1} → 1, {1, 0, 0} → 2, {0, 1, 0} → 2, {0, 0, 1} → 2,
 {1, 2, 2} → 2, {2, 1, 2} → 2, {2, 2, 1} → 1, {0, 2, 2} → 2, {2, 0, 2} → 2,
 {2, 2, 0} → 2, {0, 0, 2} → 2, {0, 2, 0} → 2, {2, 0, 0} → 2, {1, 0, 2} → 0,
 {1, 2, 0} → 0, {2, 0, 1} → 0, {2, 1, 0} → 0, {0, 1, 2} → 0, {0, 2, 1} → 0}

ruleCIR[{11, 22, 32, 42, 53, 61, 72, 83, 92, 102}]
```

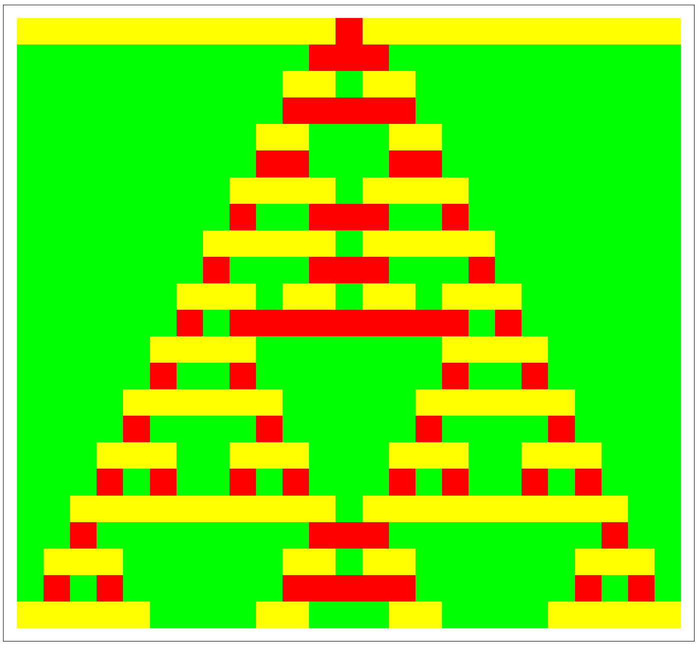




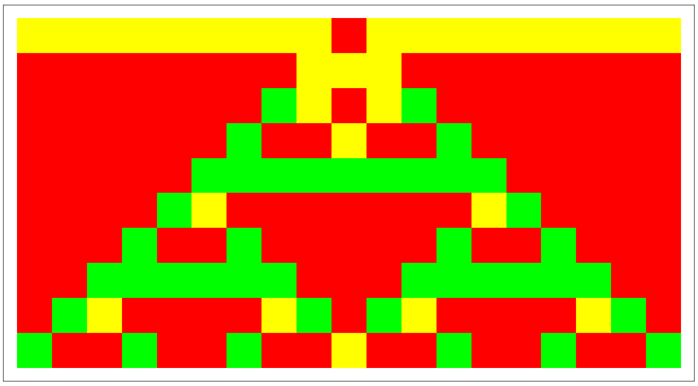


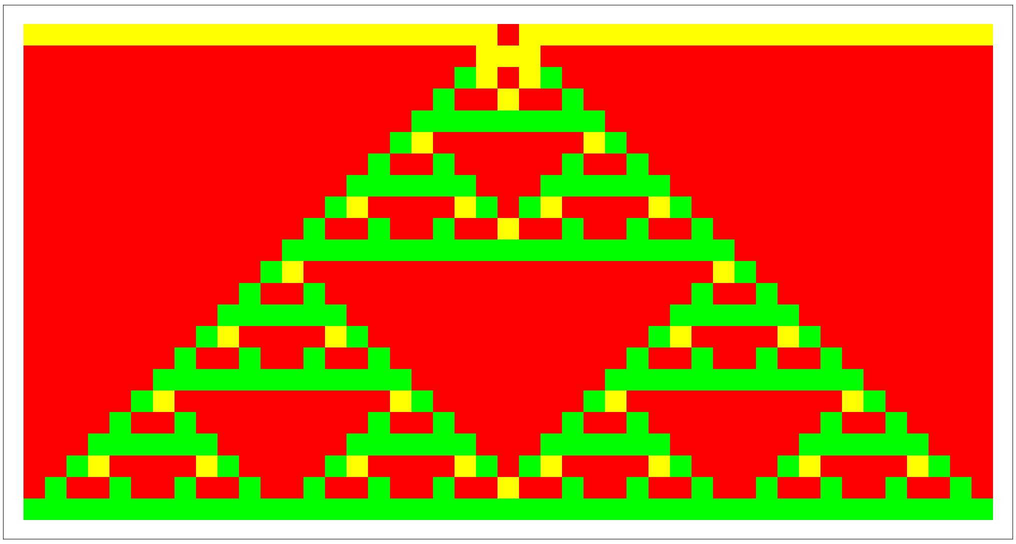


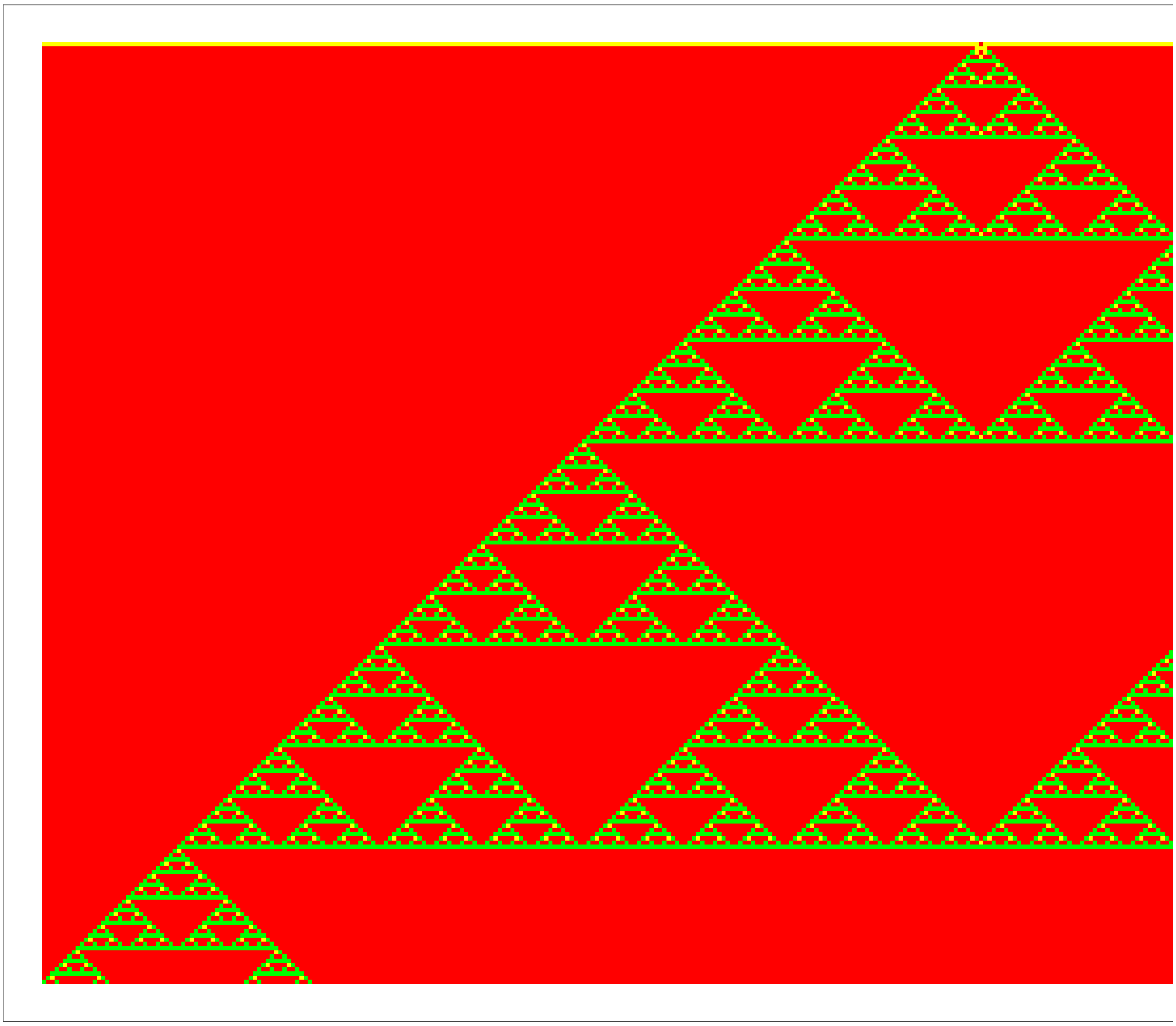


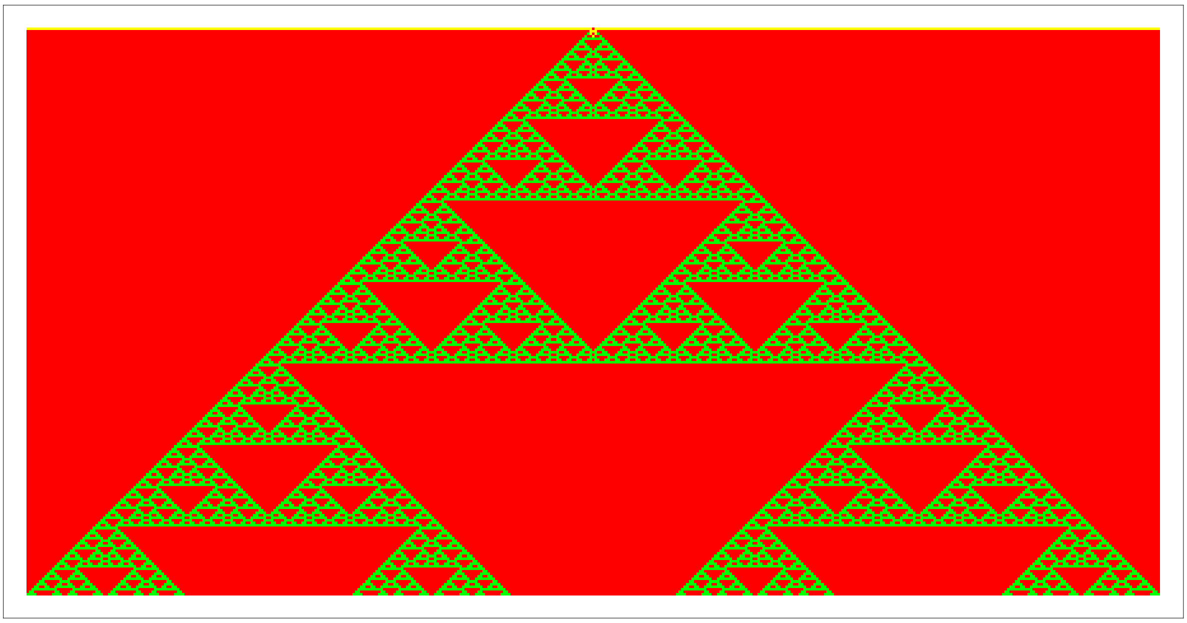


rule = 1.22.33 .4

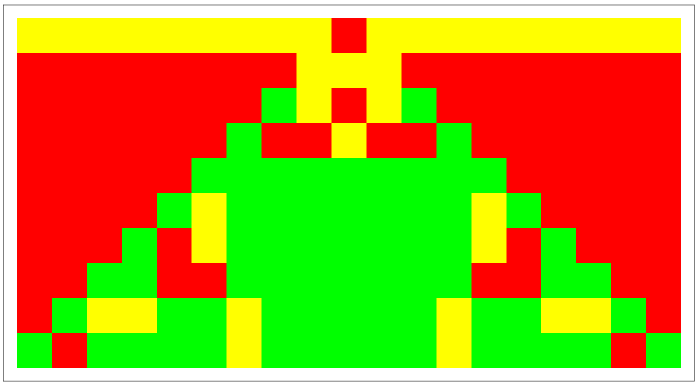


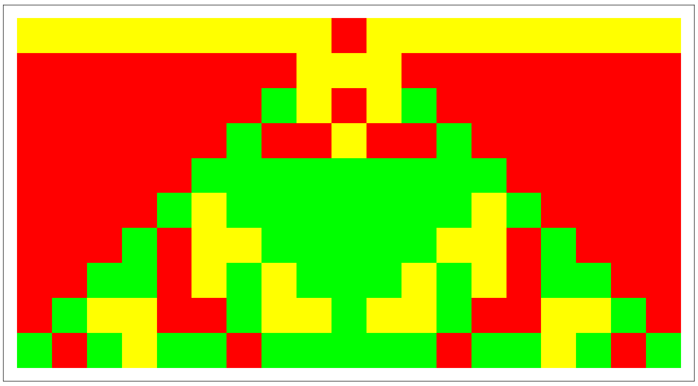


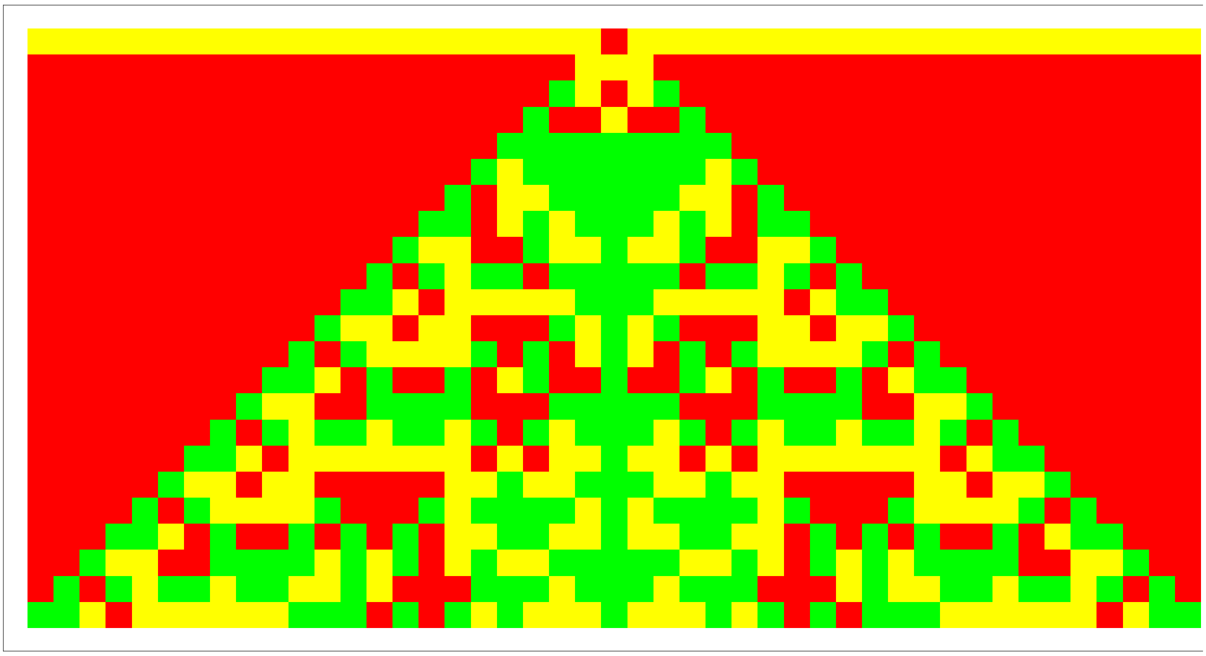


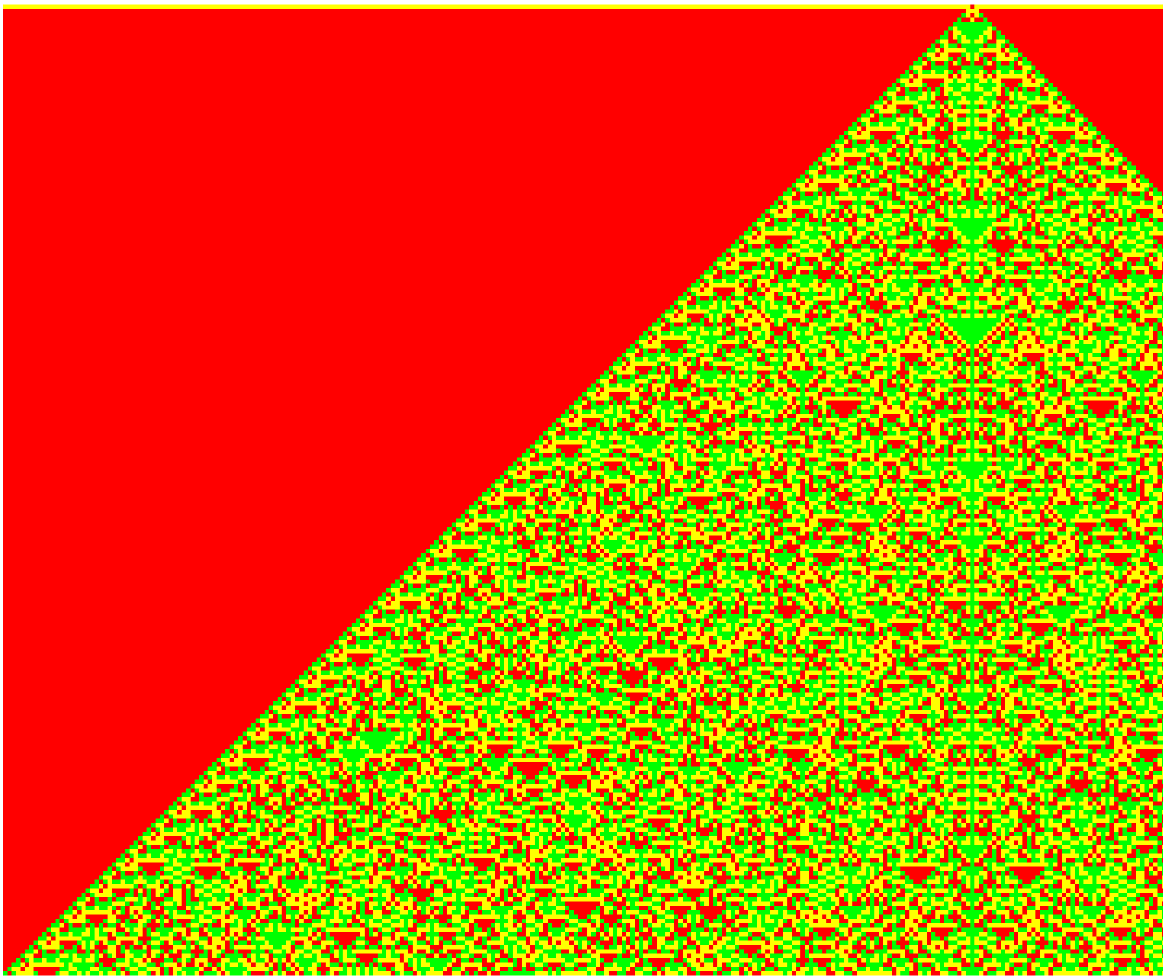


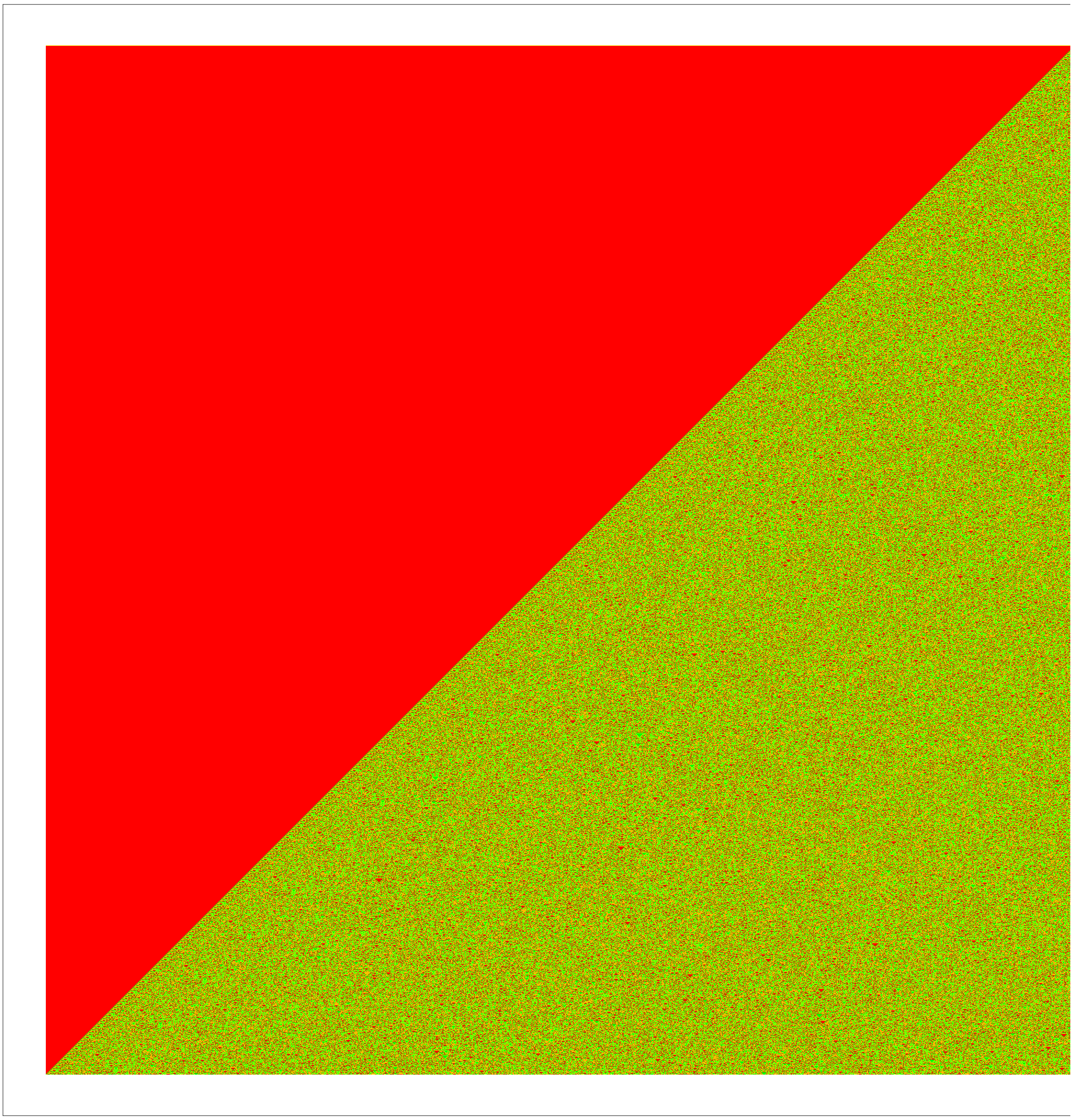
ruleCIR = 1.4.x .22 .32.

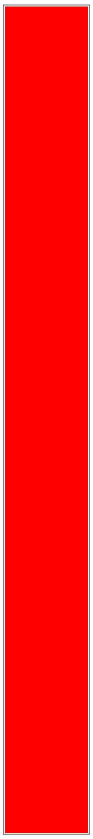


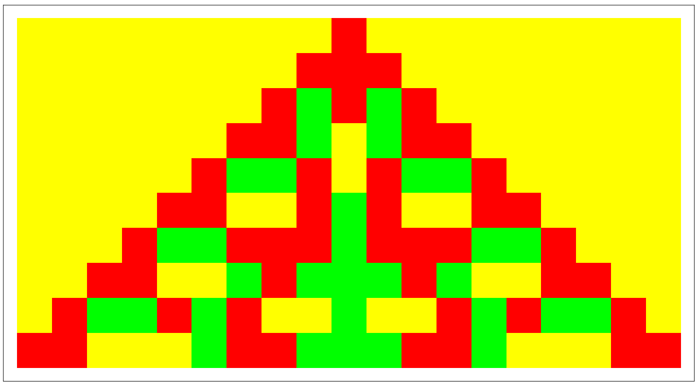


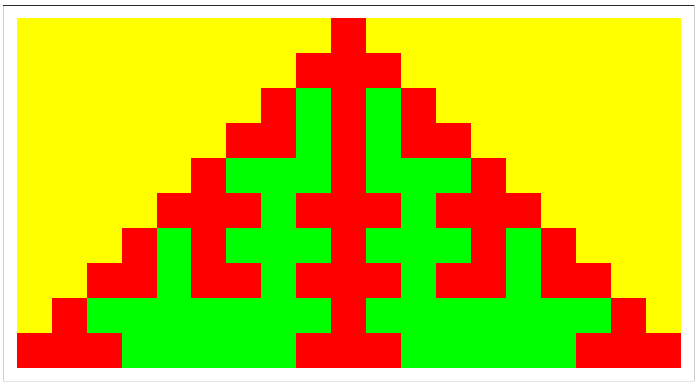


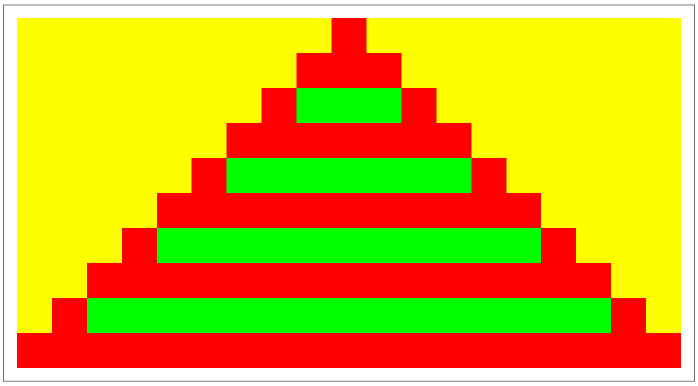


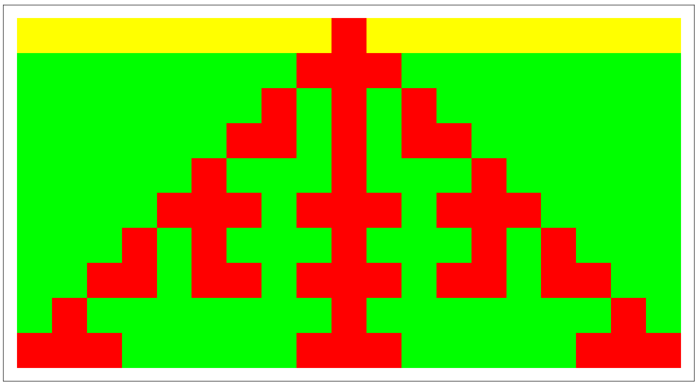


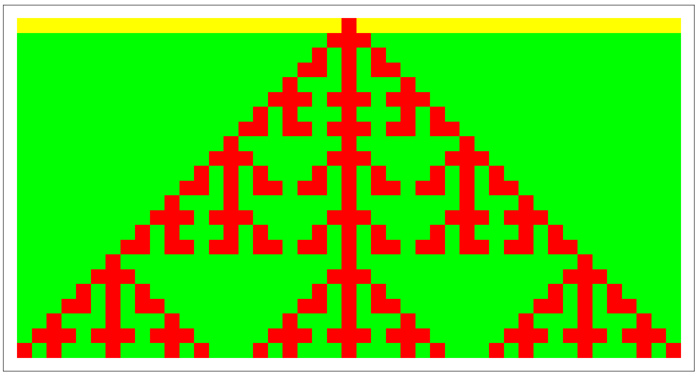


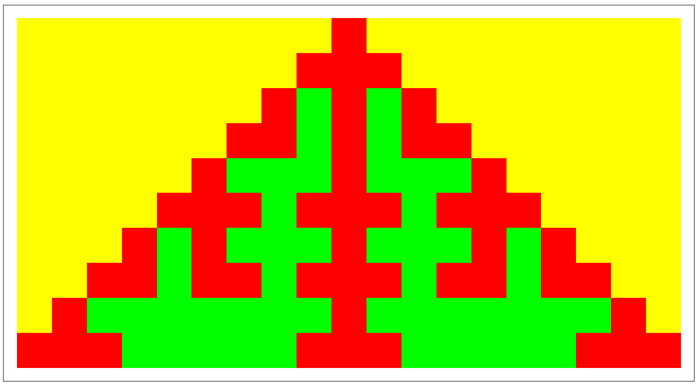


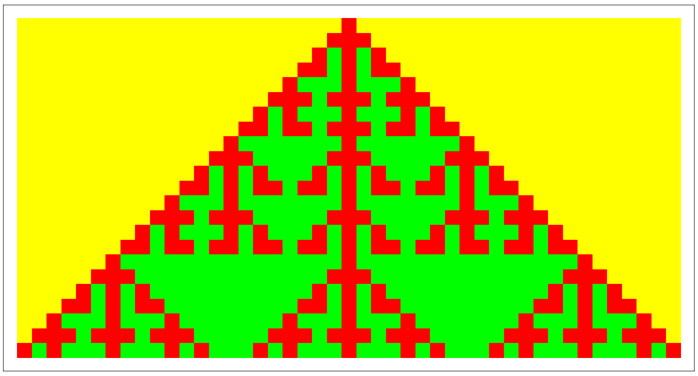


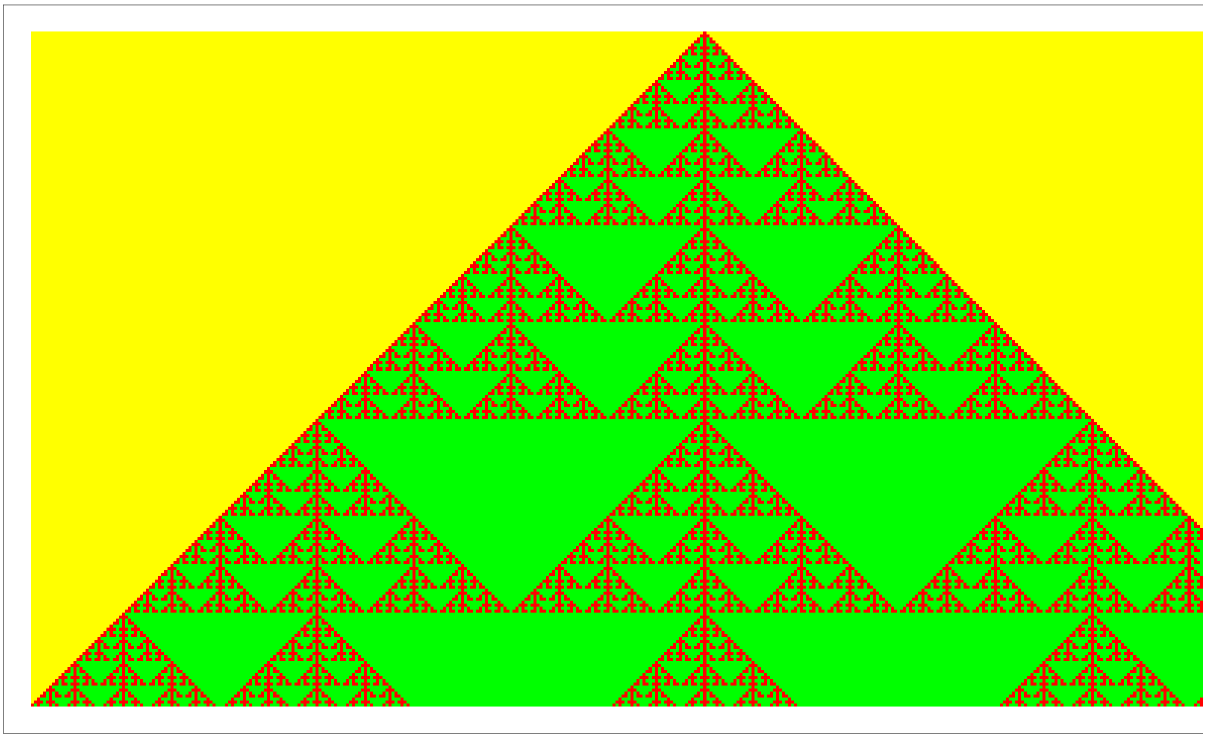


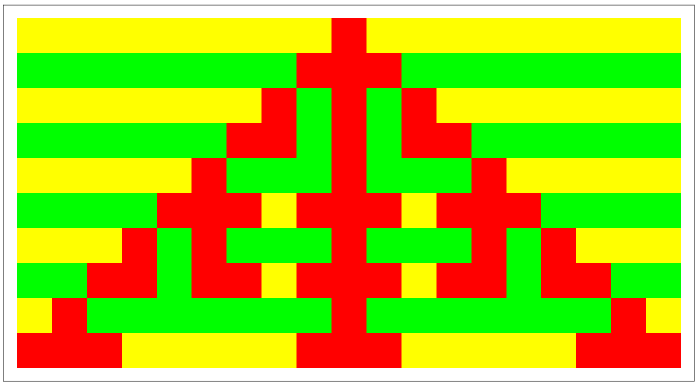


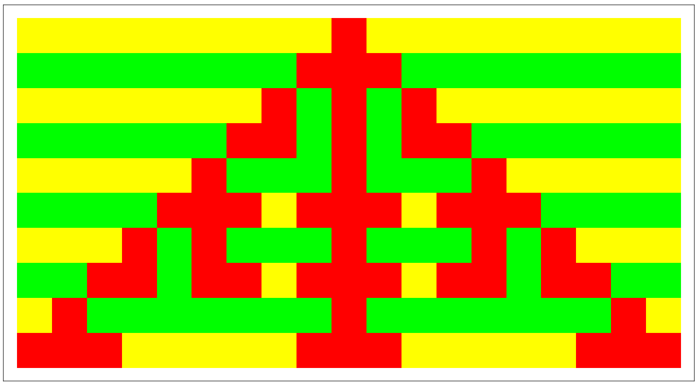


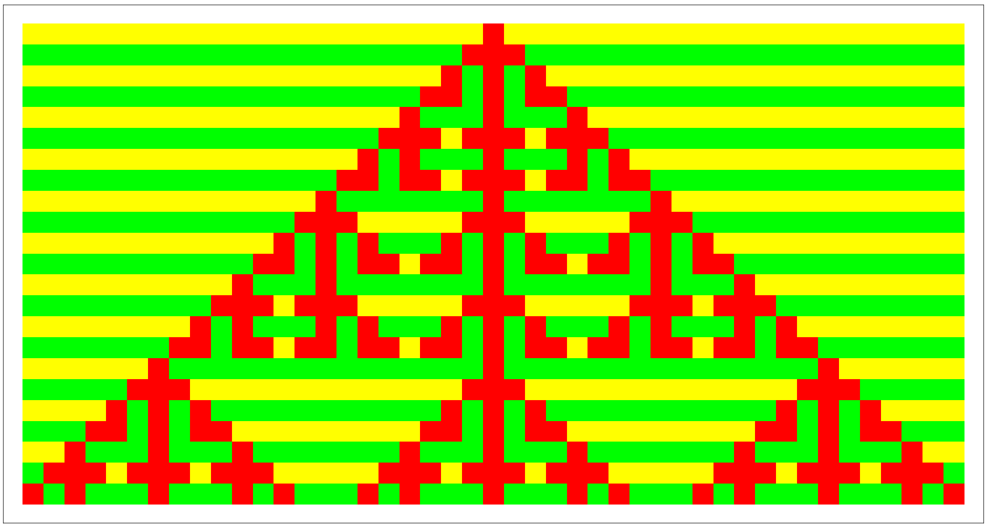


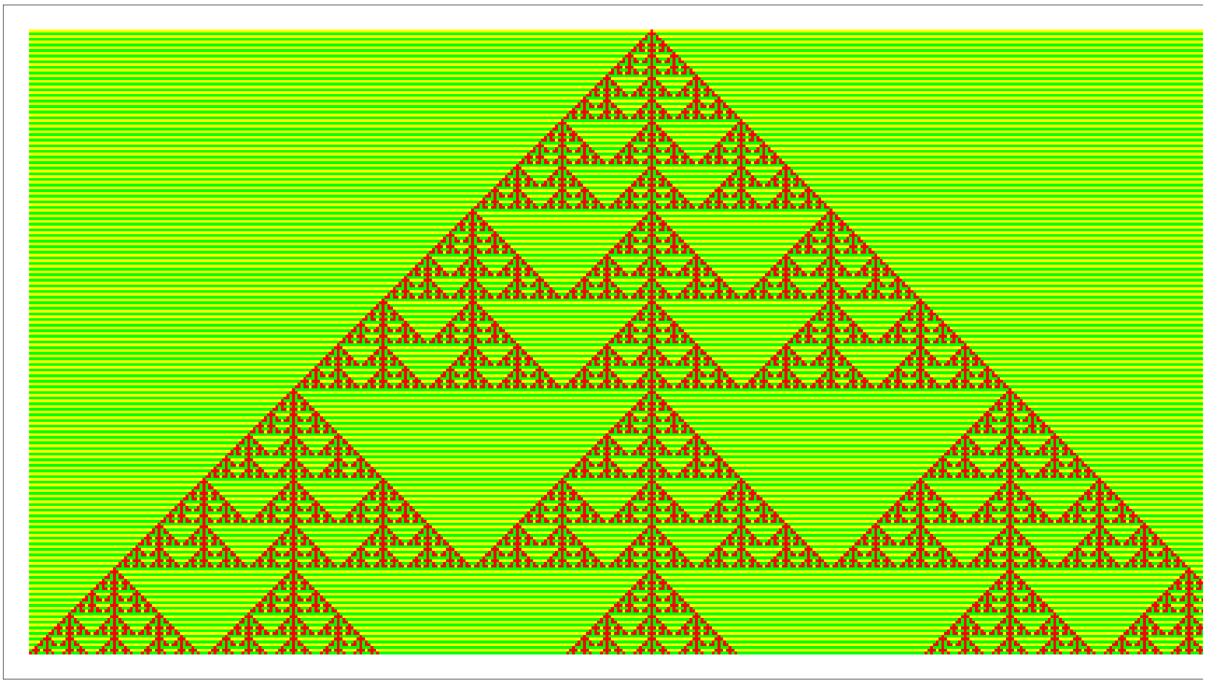


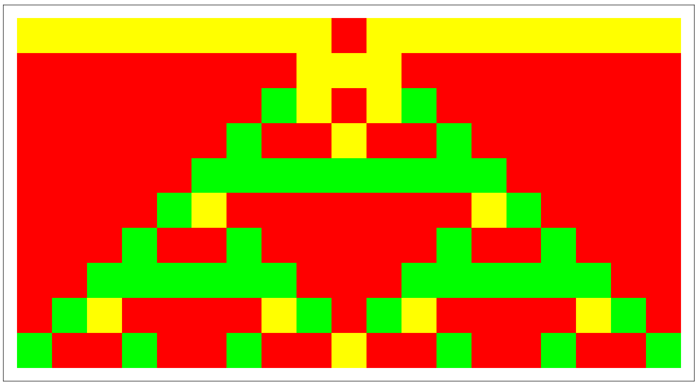


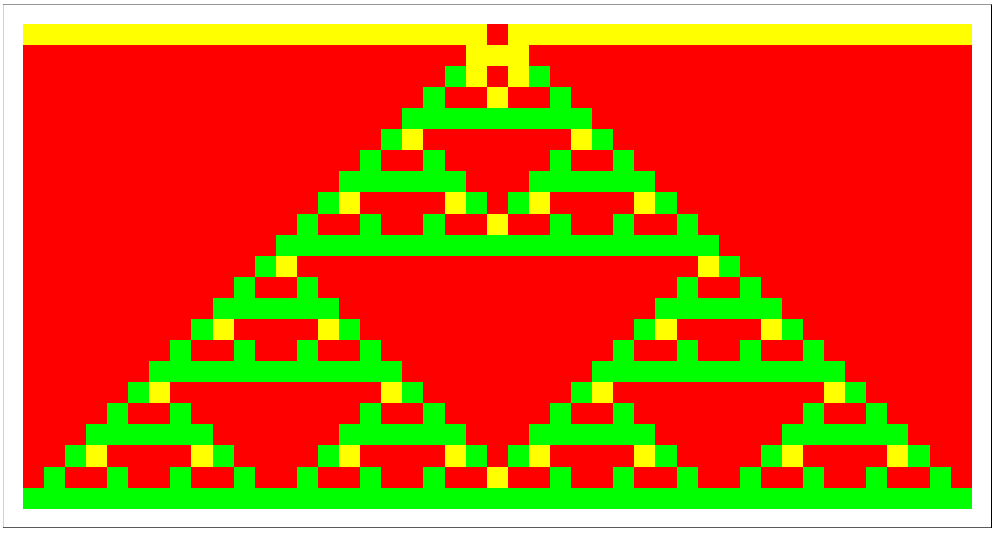


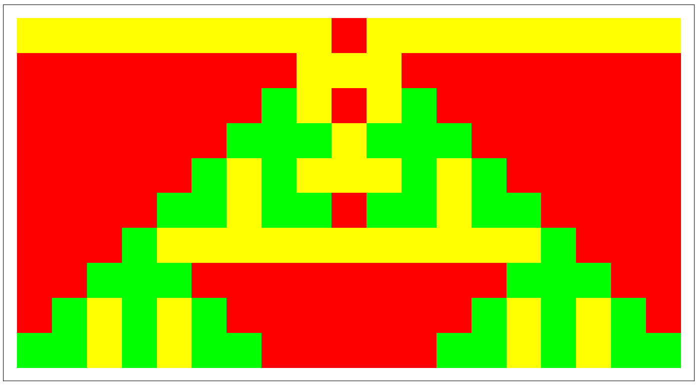


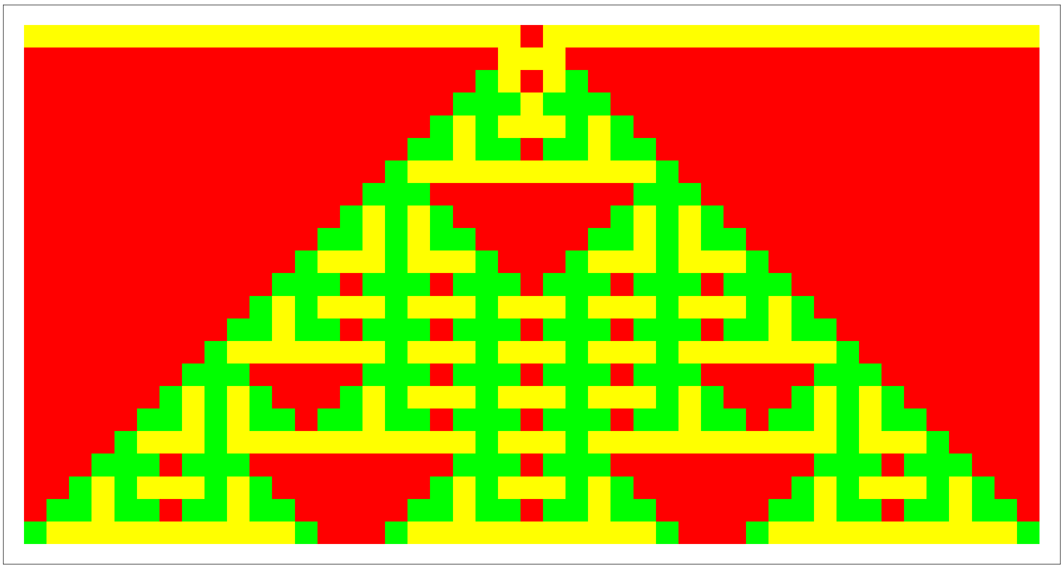


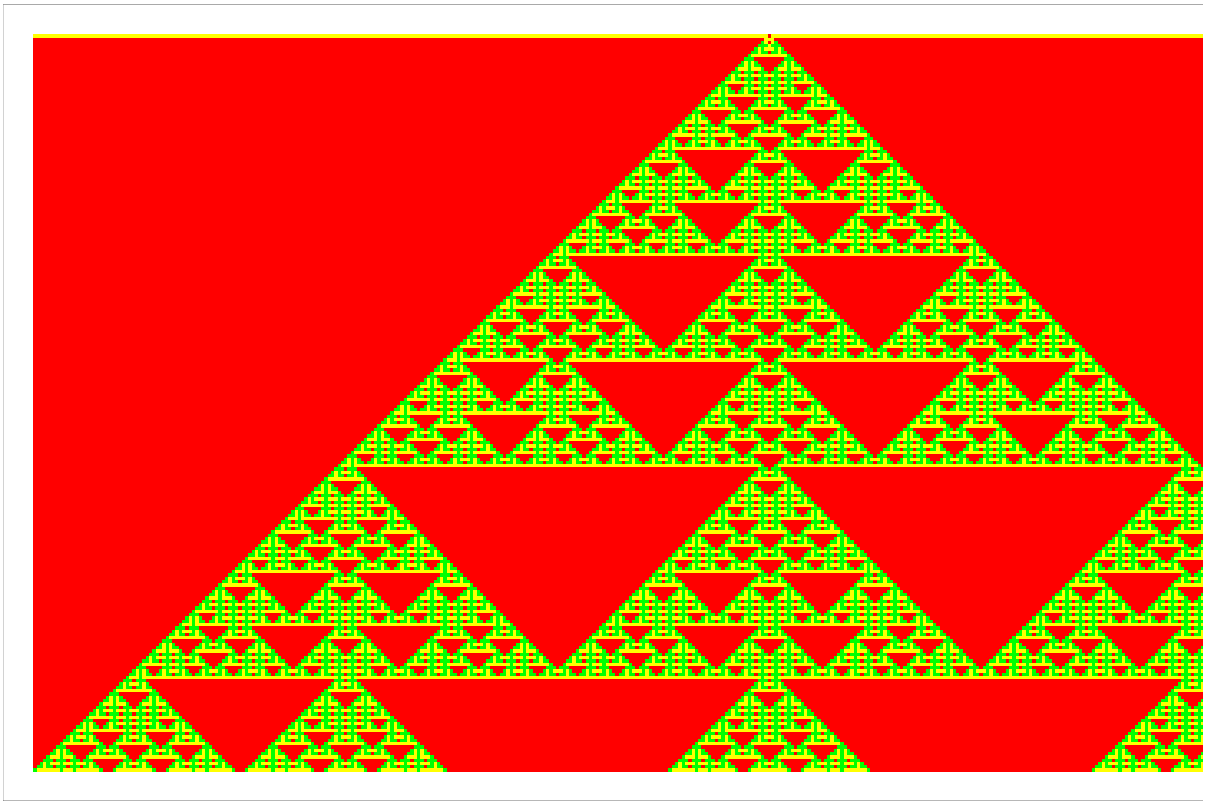


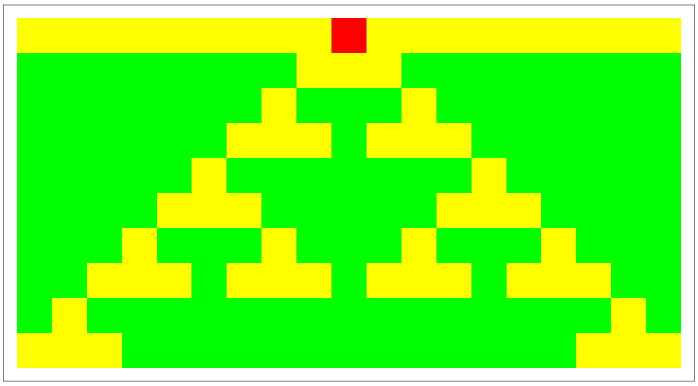


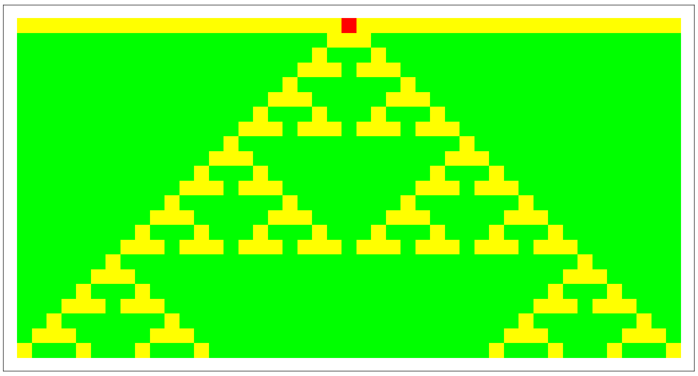


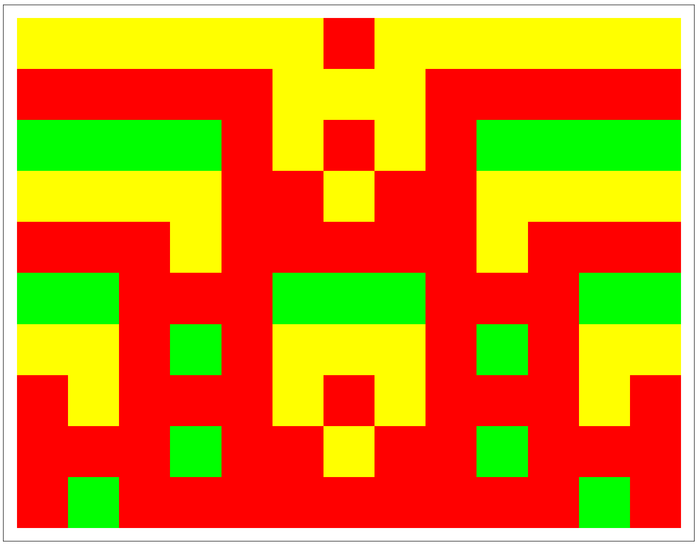


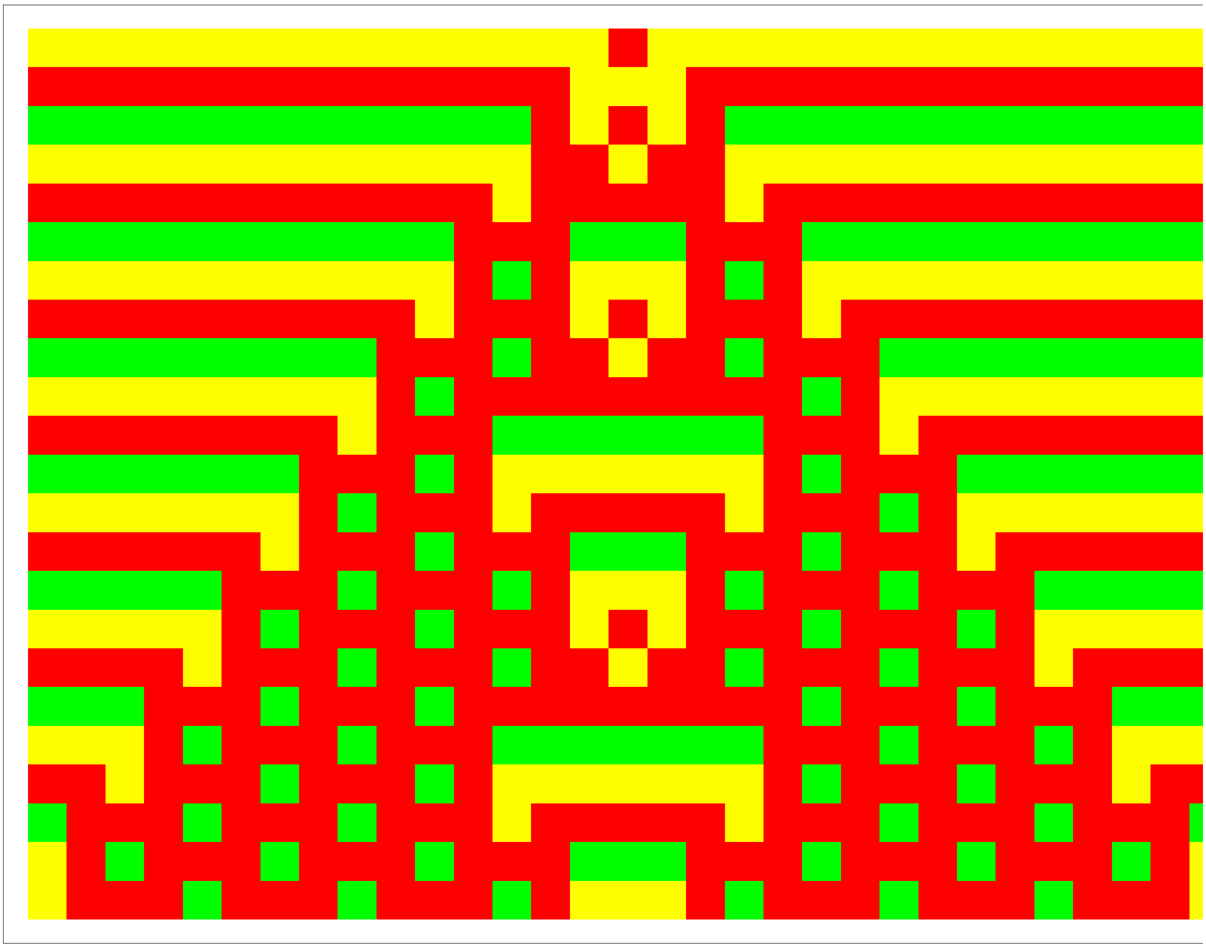


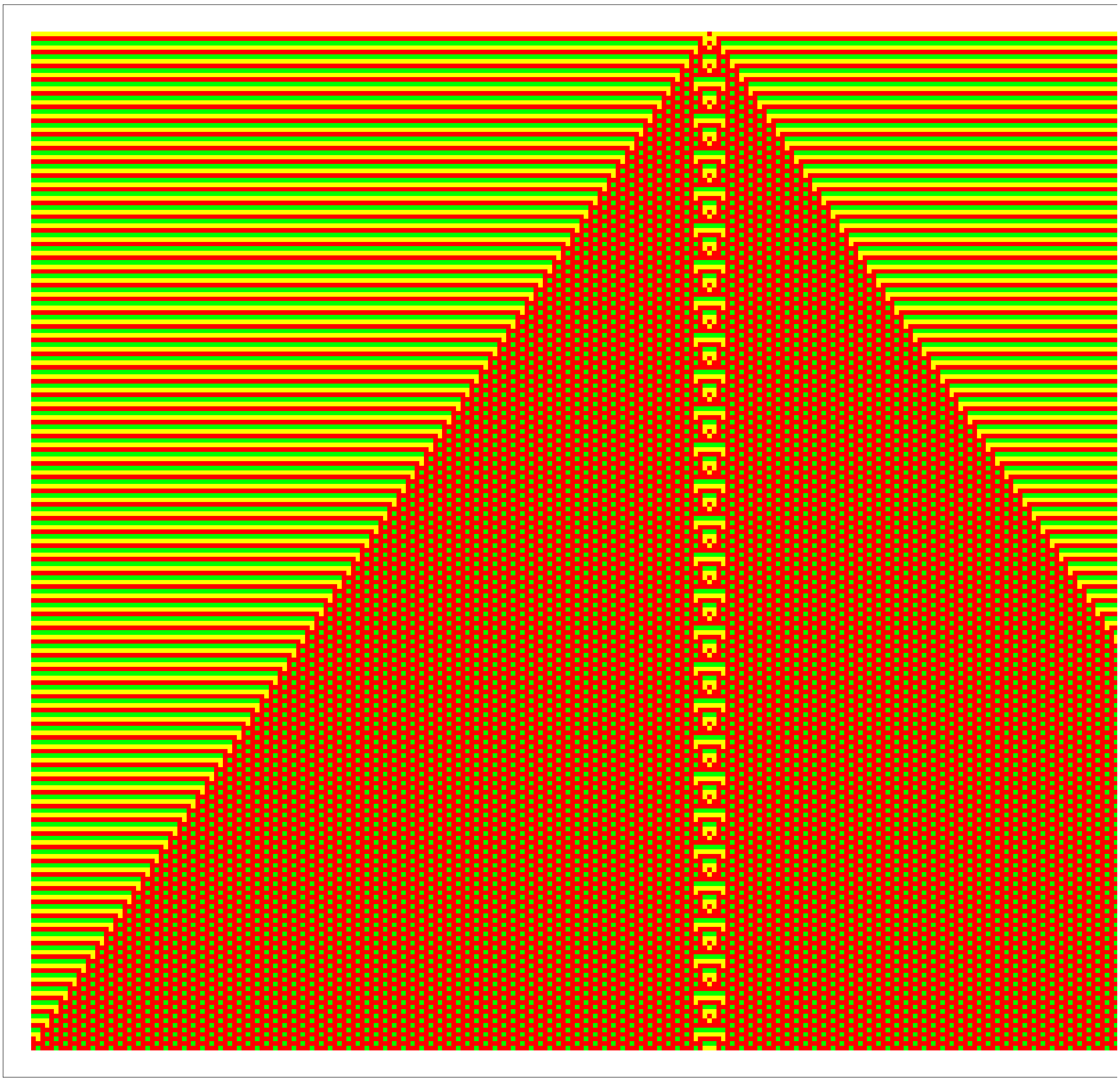


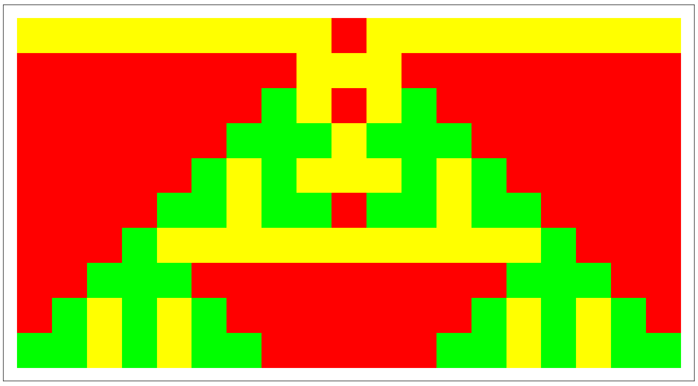


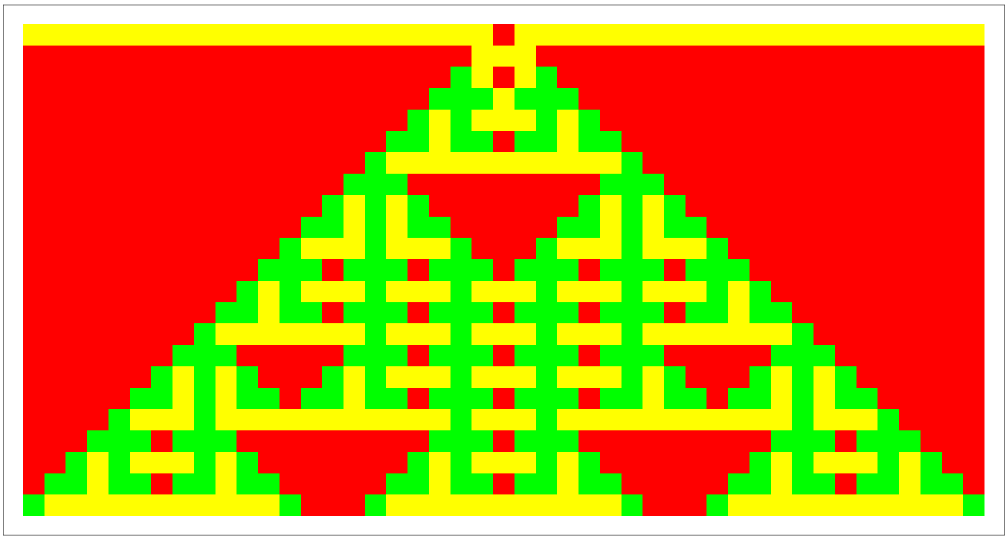


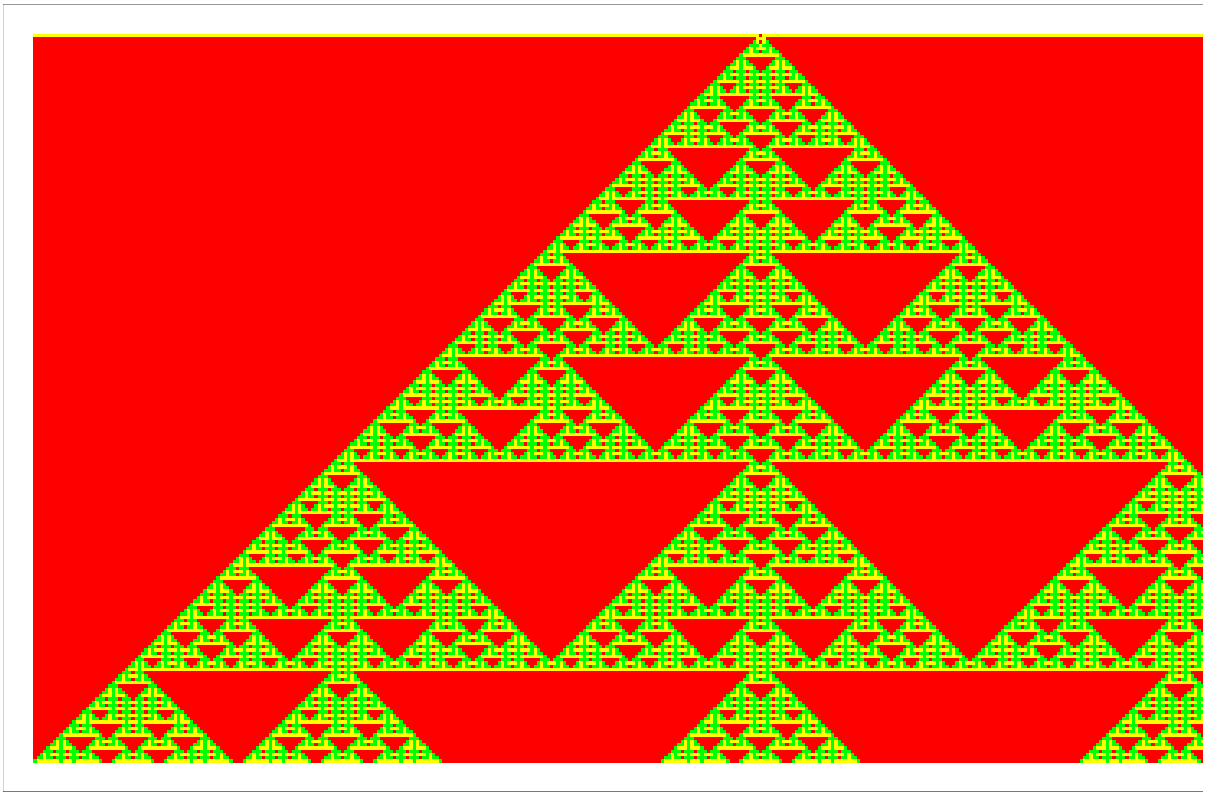


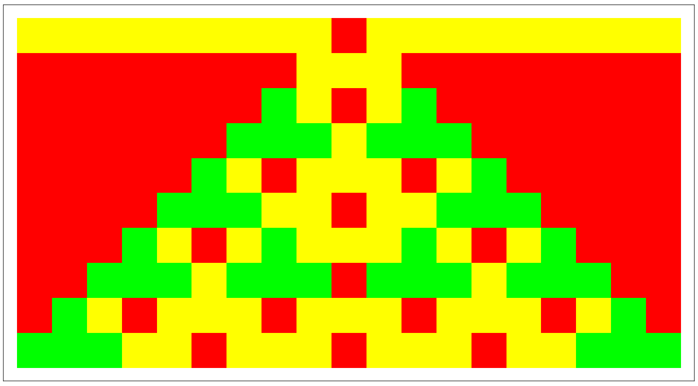


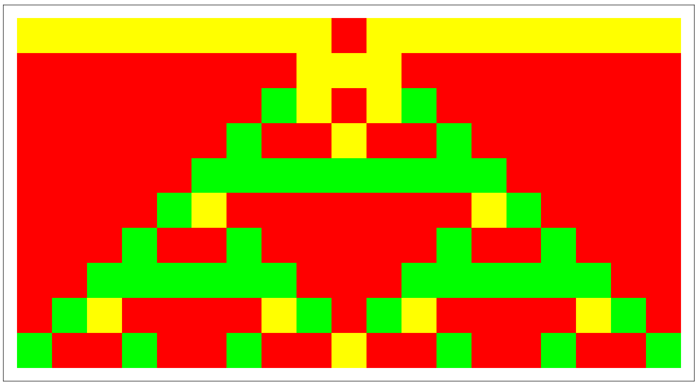


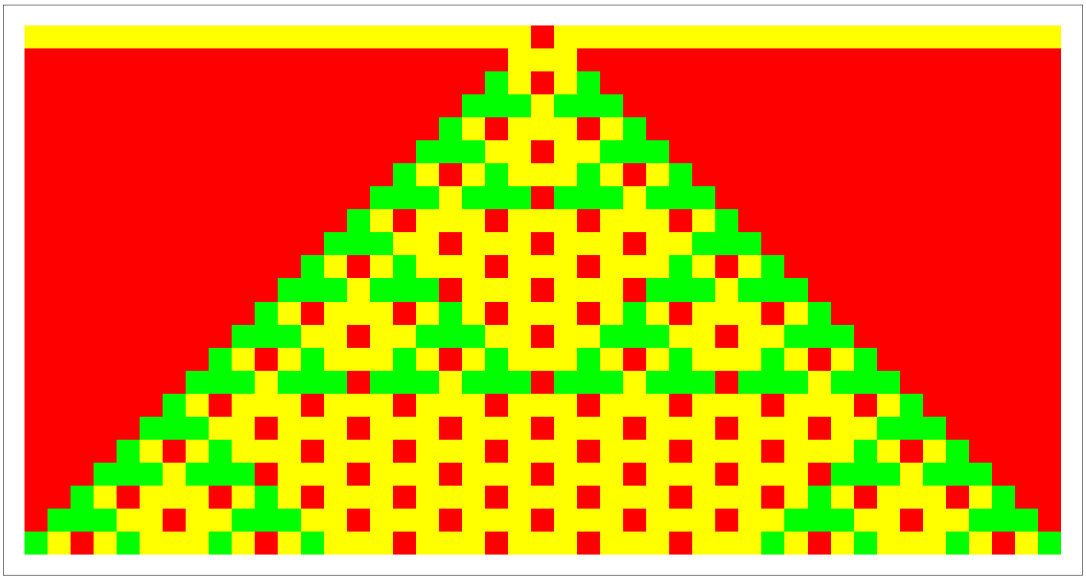


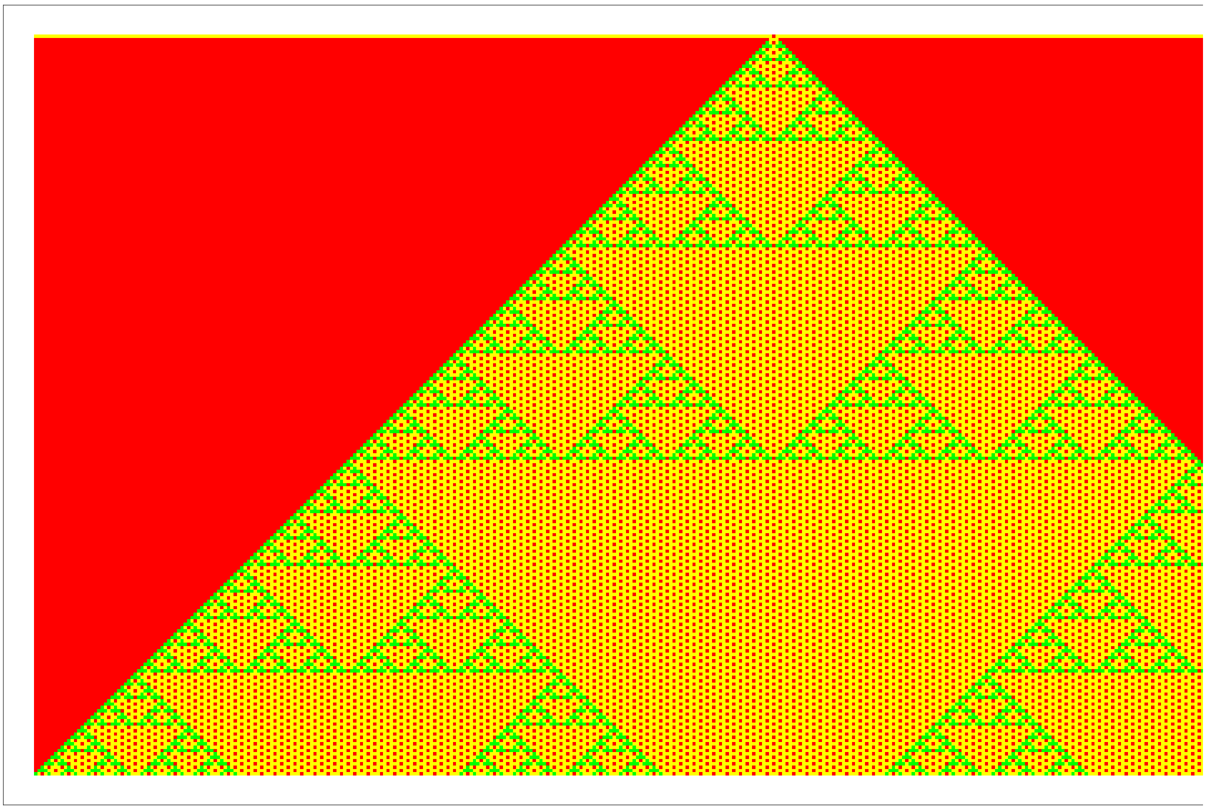


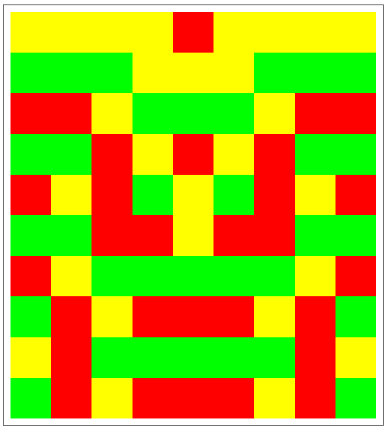




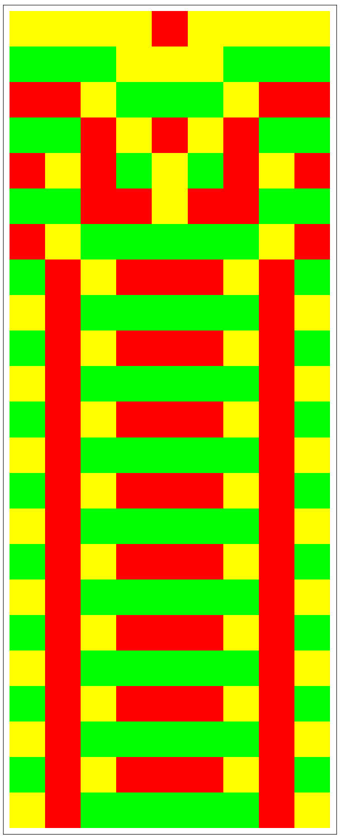




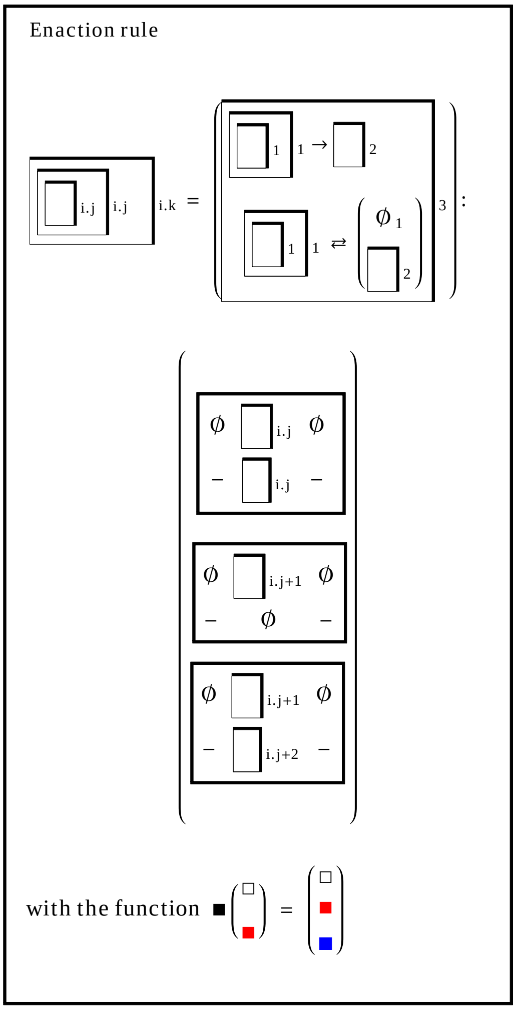




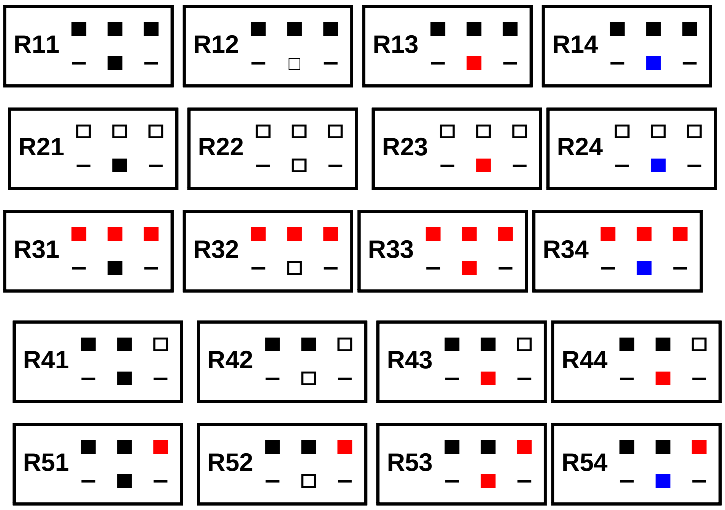
corrected (no difference)

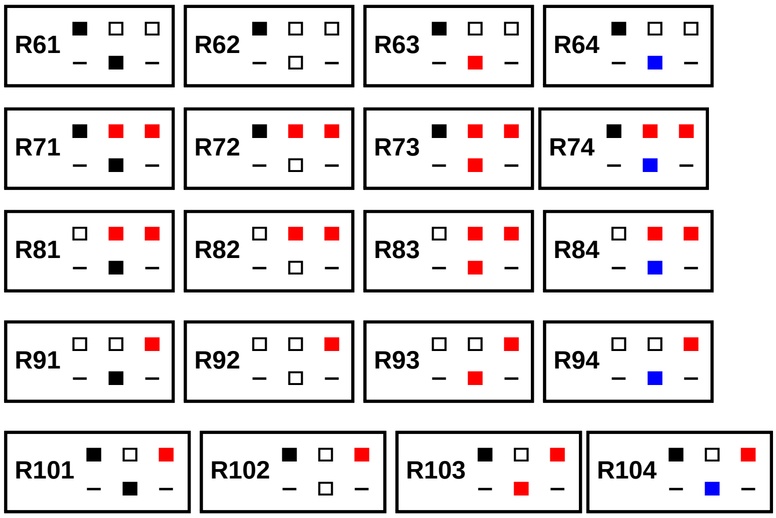


Double reflectional indCI$^{(3,4)}$



Some examples





indCR$^{(3,4)}$ Rule table: *perm*{i,j,k}

Number of rules: 80 = (12+4+4) x 4

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

{1, 0, 0} → 2,
{0, 0, 1} → 2,
{0, 1, 0} → 2,
```

```
{0, 0, 2} → 2,
{0, 2, 0} → 2,
{2, 0, 0} → 2,

{0, 2, 2} → 2,
{2, 0, 2} → 2,
{2, 2, 0} → 2,
```

```
{2, 1, 1} → 1,
{1, 2, 1} → 1,
{1, 1, 2} → 1,

{1, 2, 2} → 2,
{2, 1, 2} → 2,
{2, 2, 1} → 2,
```

```
{0, 3, 3} → 3,
{3, 0, 3} → 3,
{3, 3, 0} → 3,

{0, 0, 3} → 3,
{0, 3, 0} → 3,
{3, 0, 0} → 3,
```

```
{1, 1, 3} → 1,
{1, 3, 1} → 1,
{3, 1, 1} → 1,

{1, 3, 3} → 3,
{3, 3, 1} → 2,
{3, 1, 3} → 3,
```

```
{2, 2, 3} → 2,
{2, 3, 2} → 2,
{3, 2, 2} → 2,

{2, 3, 3} → 3,
{3, 2, 3} → 3,
{3, 3, 2} → 3,
```

```
{1, 1, 1} → 2,
{2, 2, 2} → 3,
{0, 0, 0} → 1,
{3, 3, 3} → 0,
```

```
{2, 0, 1} → 1,
{2, 1, 0} → 1,
{1, 0, 2} → 1,
{1, 2, 0} → 1,
{0, 1, 2} → 1,
{0, 2, 1} → 1,
```

```
{3, 0, 1} → 0,
{3, 1, 0} → 0,
{1, 3, 0} → 0,
{0, 1, 3} → 0,
{0, 3, 1} → 0,
{1, 0, 3} → 0,
```

```
{3, 0, 2} → 2,
{3, 2, 0} → 2,
{2, 0, 3} → 2,
{2, 3, 0} → 2,
{0, 2, 3} → 2,
{0, 3, 2} → 2,
```

```
{3, 2, 1} → 3,
{3, 1, 2} → 3,
{2, 3, 1} → 3,
{2, 1, 3} → 3,
{1, 2, 3} → 3,
{1, 3, 2} → 3
```

indCR$^{(3,4)}$ Rule set: ruleSetRCI

```
RuleSetRCI =
 {
  rci[{4440}] := {0, 0, 0} → 0,
  rci[{4441}] := {0, 0, 0} → 1,
  rci[{4442}] := {0, 0, 0} → 2,
  rci[{4443}] := {0, 0, 0} → 3,

  rci[{1110}] := {1, 1, 1} → 0,
  rci[{1111}] := {1, 1, 1} → 1,
  rci[{1112}] := {1, 1, 1} → 2,
  rci[{1113}] := {1, 1, 1} → 3,

  rci[{2220}] := {2, 2, 2} → 0,
```

```
    rci[{2221}] := {2, 2, 2} → 1,
    rci[{2222}] := {2, 2, 2} → 2,
    rci[{2223}] := {2, 2, 2} → 3,

    rci[{3330}] := {3, 3, 3} → 0,
    rci[{3331}] := {3, 3, 3} → 1,
    rci[{3332}] := {3, 3, 3} → 2,
    rci[{3333}] := {3, 3, 3} → 3,

rci[{1000}] :=
   {
    {1, 0, 0} → 0,
    {0, 0, 1} → 0,
    {0, 1, 0} → 0
   },
  rci[1001] :=
   {
    {1, 0, 0} → 1,
    {0, 0, 1} → 1,
    {0, 1, 0} → 1
   },
  rci[{1002}] :=
   {
    {1, 0, 0} → 2,
    {0, 0, 1} → 2,
    {0, 1, 0} → 2
   },
  rci[{1003}] :=
   {
    {1, 0, 0} → 3,
    {0, 0, 1} → 3,
    {0, 1, 0} → 3
   },
rci[{2000}] :=
   {
    {2, 0, 0} → 0,
    {0, 0, 2} → 0,
    {0, 2, 0} → 0
   },
  rci[2001] :=
   {
    {2, 0, 0} → 1,
    {0, 0, 2} → 1,
    {0, 2, 0} → 1
   },
  rci[{2002}] :=
   {
    {2, 0, 0} → 2,
    {0, 0, 2} → 2,
    {0, 2, 0} → 2
   },
  rci[{2003}] :=
   {
    {2, 0, 0} → 3,
    {0, 0, 2} → 3,
    {0, 2, 0} → 3
   },
   rci[{3000}] :=
   {
```

```
      {3, 0, 0} → 0,
      {0, 0, 3} → 0,
      {0, 3, 0} → 0
     },
   rci[3001] :=
     {
      {3, 0, 0} → 1,
      {0, 0, 3} → 1,
      {0, 3, 0} → 1
     },
   rci[{3002}] :=
     {
      {3, 0, 0} → 2,
      {0, 0, 3} → 2,
      {0, 3, 0} → 2
     },
   rci[{3003}] :=
     {
      {3, 0, 0} → 3,
      {0, 0, 3} → 3,
      {0, 3, 0} → 3
     },

rci[{1100}] :=
     {
      {0, 1, 1} → 0,
      {1, 1, 0} → 0,
      {1, 0, 1} → 0
     },
   rci[{1101}] :=
     {
      {0, 1, 1} → 1,
      {1, 1, 0} → 1,
      {1, 0, 1} → 1
     },
   rci[{1102}] :=
     {
      {0, 1, 1} → 2,
      {1, 1, 0} → 2,
      {1, 0, 1} → 2
     },
   rci[{1103}] :=
     {
      {0, 1, 1} → 3,
      {1, 1, 0} → 3,
      {1, 0, 1} → 3
     },
   rci[{1120}] :=
     {
      {1, 1, 2} → 0,
      {2, 1, 1} → 0,
      {1, 2, 1} → 0
     },
   rci[{1121}] :=
     {
      {1, 1, 2} → 1,
      {2, 1, 1} → 1,
      {1, 2, 1} → 1
     },
```

```
   rci[{1122}] :=
    {
     {2, 1, 1} → 2,
     {1, 2, 1} → 2,
     {1, 1, 2} → 2
    },
   rci[{1123}] :=
    {
     {2, 1, 1} → 3,
     {1, 2, 1} → 3,
     {1, 1, 2} → 3
    },
 rci[{1130}] :=
    {
     {1, 1, 3} → 0,
     {1, 3, 1} → 0,
     {3, 1, 1} → 0
    },
   rci[{1131}] :=
    {
     {1, 1, 3} → 1,
     {1, 3, 1} → 1,
     {3, 1, 1} → 1
    },
   rci[{1132}] :=
    {
     {1, 1, 3} → 2,
     {1, 3, 1} → 2,
     {3, 1, 1} → 2
    },
   rci[{1133}] :=
    {
     {1, 1, 3} → 3,
     {1, 3, 1} → 0,
     {3, 1, 1} → 3
    },
 rci[{1220}] :=
    {
     {1, 2, 2} → 0,
     {2, 1, 2} → 0,
     {2, 2, 1} → 0,
    },
   rci[{1221}] :=
    {
     {1, 2, 2} → 1,
     {2, 1, 2} → 1,
     {2, 2, 1} → 1
    },
   rci[{1222}] :=
    {
     {1, 2, 2} → 2,
     {2, 1, 2} → 2,
     {2, 2, 1} → 2
    },
   rci[{1223}] :=
    {
     {1, 2, 2} → 3,
     {2, 1, 2} → 3,
     {2, 2, 1} → 3
```

```
    },
rci[{1330}] :=
    {
     {1, 3, 3} → 0,
     {3, 3, 1} → 0,
     {3, 1, 3} → 0
    },
  rci[{1331}] :=
    {
     {1, 3, 3} → 1,
     {3, 3, 1} → 1,
     {3, 1, 3} → 1
    },
  rci[{1332}] :=
    {
     {1, 3, 3} → 2,
     {3, 3, 1} → 2,
     {3, 1, 3} → 2
    },
  rci[{1333}] :=
    {
     {1, 3, 3} → 3,
     {3, 3, 1} → 3,
     {3, 1, 3} → 3
    },
rci[{2200}] :=
    {
     {0, 2, 2} → 0,
     {2, 0, 2} → 0,
     {2, 2, 0} → 0
    },
  rci[{2201}] :=
    {
     {0, 2, 2} → 1,
     {2, 0, 2} → 1,
     {2, 2, 0} → 1
    },
  rci[{2202}] :=
    {
     {0, 2, 2} → 2,
     {2, 0, 2} → 2,
     {2, 2, 0} → 2
    },
  rci[{2203}] :=
    {
     {0, 2, 2} → 3,
     {2, 0, 2} → 3,
     {2, 2, 0} → 3
    },
rci[{3300}] :=
    {
     {0, 3, 3} → 0,
     {3, 0, 3} → 0,
     {3, 3, 0} → 0
    },
  rci[{3301}] :=
    {
     {0, 3, 3} → 1,
     {3, 0, 3} → 1,
```

```
      {3, 3, 0} → 1
     },
    rci[{3302}] :=
     {
      {0, 3, 3} →  2,
      {3, 0, 3} → 2,
      {3, 3, 0} → 2
     },
    rci[{3303}] :=
     {
      {0, 3, 3} → 3,
      {3, 0, 3} → 3,
      {3, 3, 0} → 3
     },
  rci[{2230}] :=
     {
      {2, 2, 3} → 0,
      {2, 3, 2} → 0,
      {3, 2, 2} → 0
     },
    rci[{2231}] :=
     {
      {2, 2, 3} → 1,
      {2, 3, 2} → 1,
      {3, 2, 2} → 1
     },
    rci[{2232}] :=
     {
      {2, 2, 3} → 2,
      {2, 3, 2} → 2,
      {3, 2, 2} → 2
     },
    rci[{2233}] :=
     {
      {2, 2, 3} → 3,
      {2, 3, 2} → 3,
      {3, 2, 2} → 3
     },
  rci[{2330}] :=
     {
      {2, 3, 3} → 0,
      {3, 2, 3} → 0,
      {3, 3, 2} → 0
     },
    rci[{2331}] :=
     {
      {2, 3, 3} → 1,
      {3, 2, 3} → 1,
      {3, 3, 2} → 1
     },
    rci[{2332}] :=
     {
      {2, 3, 3} → 2,
      {3, 2, 3} → 2,
      {3, 3, 2} → 2
     },
    rci[{2333}] :=
     {
      {2, 3, 3} → 3,
```

```
      {3, 2, 3} → 3,
      {3, 3, 2} → 3
     },
    rci[{1200}] :=
     {
      {0, 1, 2} → 0,
      {2, 0, 1} → 0,
      {2, 1, 0} → 0,
      {1, 0, 2} → 0,
      {1, 2, 0} → 0,
      {0, 2, 1} → 0
     },
    rci[{1201}] :=
     {
      {0, 1, 2} → 1,
      {2, 0, 1} → 1,
      {2, 1, 0} → 1,
      {1, 0, 2} → 1,
      {1, 2, 0} → 1,
      {0, 2, 1} → 1
     },
  rci[{1202}] :=
     {
      {0, 1, 2} → 2,
      {2, 0, 1} → 2,
      {2, 1, 0} → 2,
      {1, 0, 2} → 2,
      {1, 2, 0} → 2,
      {0, 2, 1} → 2
     },
    rci[{1203}] :=
     {
      {0, 1, 2} → 3,
      {2, 0, 1} → 3,
      {2, 1, 0} → 3,
      {1, 0, 2} → 3,
      {1, 2, 0} → 3,
      {0, 2, 1} → 3
     },

    rci[{1300}] :=
     {
      {0, 1, 3} → 0,
      {3, 0, 1} → 0,
      {3, 1, 0} → 0,
      {1, 0, 3} → 0,
      {1, 3, 0} → 0,
      {0, 3, 1} → 0
     },
    rci[{1301}] :=
     {
      {0, 1, 3} → 1,
      {3, 0, 1} → 1,
      {3, 1, 0} → 1,
      {1, 0, 3} → 1,
      {1, 3, 0} → 1,
      {0, 3, 1} → 1
     },
    rci[{1302}] :=
```

```
      {
       {0, 1, 3} → 2,
       {3, 0, 1} → 2,
       {3, 1, 0} → 2,
       {1, 0, 3} → 2,
       {1, 3, 0} → 2,
       {0, 3, 1} → 2
      },
     rci[{1303}] :=
      {
       {0, 1, 3} → 3,
       {3, 0, 1} → 3,
       {3, 1, 0} → 3,
       {1, 0, 3} → 3,
       {1, 3, 0} → 3,
       {0, 3, 1} → 3
      },
   rci[{3200}] :=
      {
       {3, 0, 2} → 0,
       {3, 2, 0} → 0,
       {2, 0, 3} → 0,
       {2, 3, 0} → 0,
       {0, 2, 3} → 0,
       {0, 3, 2} → 0
      },
     rci[{3201}] :=
      {
       {3, 0, 2} → 1,
       {3, 2, 0} → 1,
       {2, 0, 3} → 1,
       {2, 3, 0} → 1,
       {0, 2, 3} → 1,
       {0, 3, 2} → 1
      },
     rci[{3202}] :=
      {
       {3, 0, 2} → 2,
       {3, 2, 0} → 2,
       {2, 0, 3} → 2,
       {2, 3, 0} → 2,
       {0, 2, 3} → 2,
       {0, 3, 2} → 2
      },
     rci[{3203}] :=
      {
       {3, 0, 2} → 3,
       {3, 2, 0} → 3,
       {2, 0, 3} → 3,
       {2, 3, 0} → 3,
       {0, 2, 3} → 3,
       {0, 3, 2} → 3
      },
   rci[{3210}] :=
      {
       {3, 2, 1} → 0,
       {3, 1, 2} → 0,
       {2, 3, 1} → 0,
       {2, 1, 3} → 0,
```

```
     {1, 2, 3} → 0,
     {1, 3, 2} → 0
    },
   rci[{3211}] :=
    {
     {3, 2, 1} → 1,
     {3, 1, 2} → 1,
     {2, 3, 1} → 1,
     {2, 1, 3} → 1,
     {1, 2, 3} → 1,
     {1, 3, 2} → 1
    },
   rci[{3212}] :=
    {
     {3, 2, 1} → 2,
     {3, 1, 2} → 2,
     {2, 3, 1} → 2,
     {2, 1, 3} → 2,
     {1, 2, 3} → 2,
     {1, 3, 2} → 2
    },
   rci[{3213}] :=
    {
     {3, 2, 1} → 3,
     {3, 1, 2} → 3,
     {2, 3, 1} → 3,
     {2, 1, 3} → 3,
     {1, 2, 3} → 3,
     {1, 3, 2} → 3
    }
  }
```

indCR$^{(3,4)}$ruleSetRCI: MenuView (todo)

```
15 ∨
Null
```

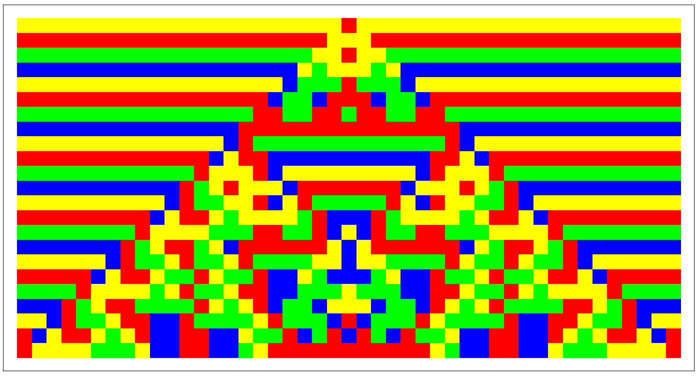
indCR$^{(3,4)}$ Rule scheme

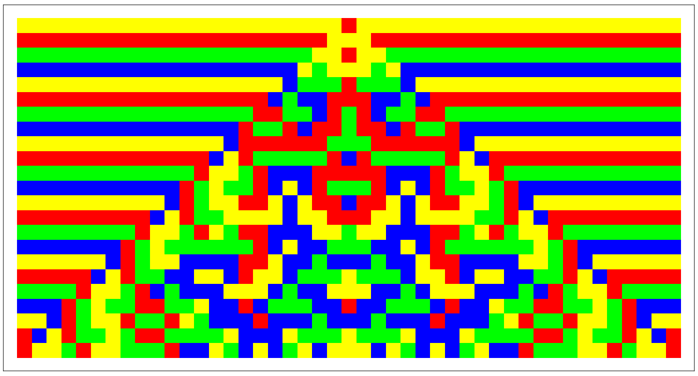
```
ruleRCI[{a_, b_, c_, d_, e_, f_, g_, h_, i_, j_,
         k_, h_, l_, m_, n_, o_, p_, q_, r_, s_
        }] :=

   Flatten[{
    rci[{a}], rci[{b}], rci[{c}], rci[{d}],
    rci[{e}], rci[{f}], rci[{g}], rci[{h}],
    rci[{i}], rci[{j}], rci[{k}], rci[{h}],
    rci[{l}], rci[{m}], rci[{n}], rci[{o}],
    rci[{p}], rci[{q}], rci[{r}], rci[{s}]
   }]
```

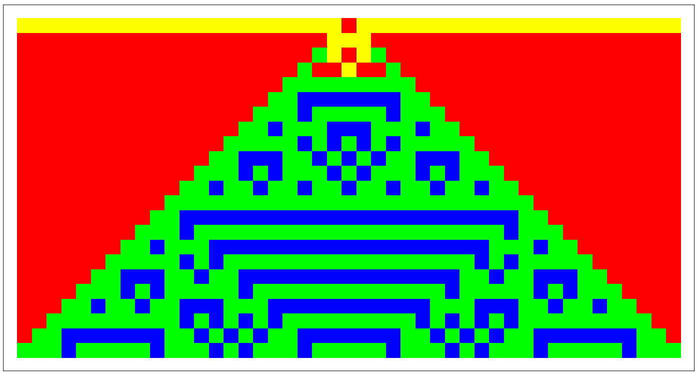
Dynamic view of indCR$^{(3,4)}$ forms (todo)

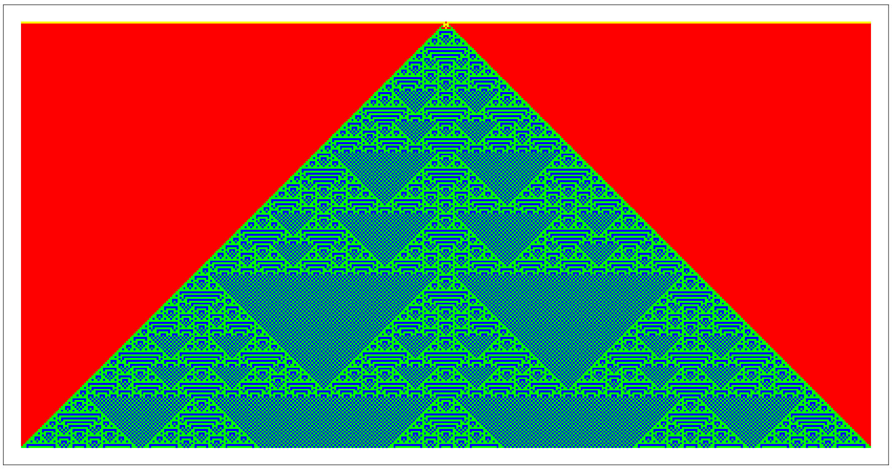
Examples of complete forms:

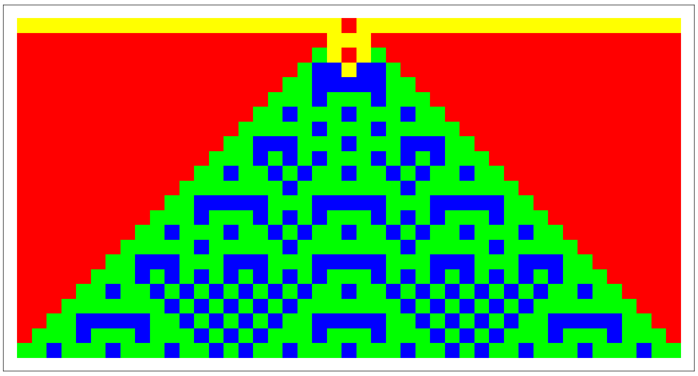
First: Symmetric indCA$^{(4,3)}$

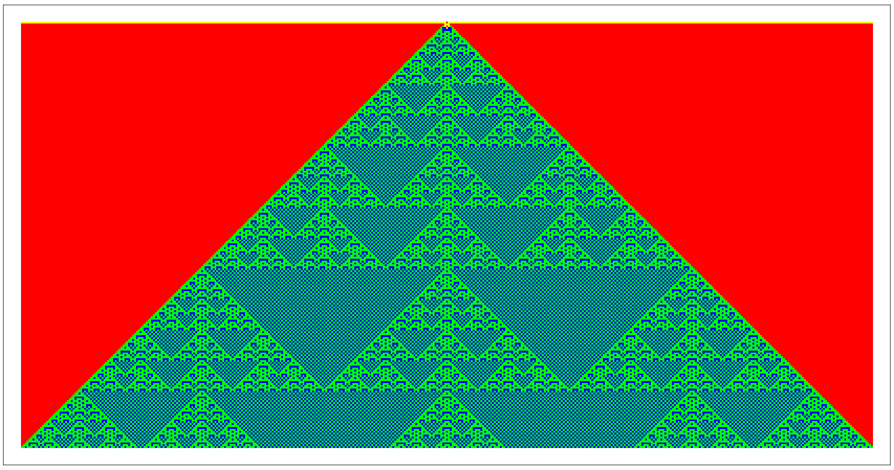


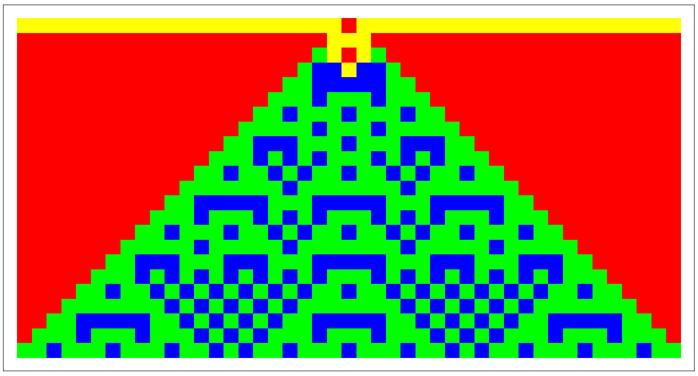


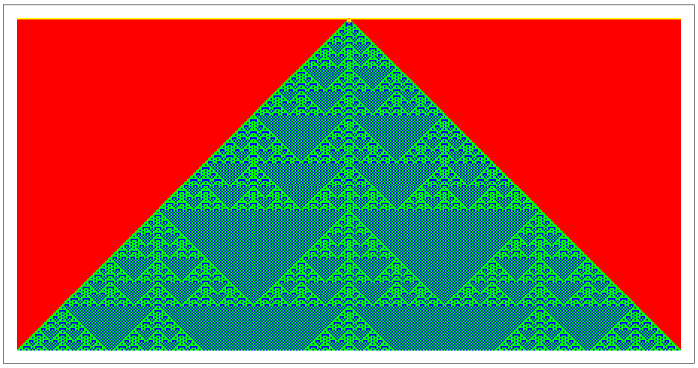


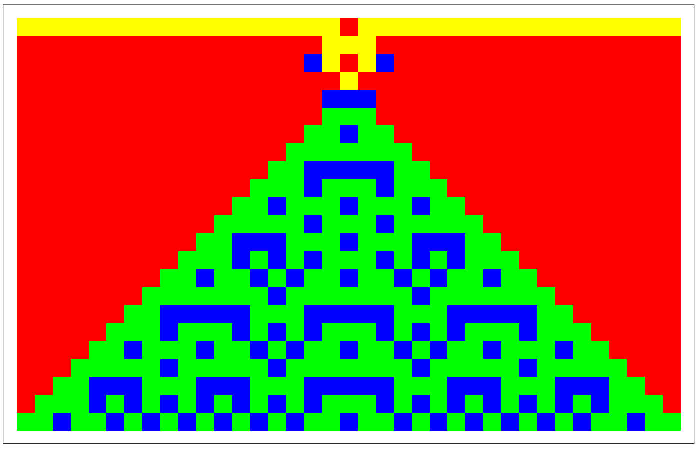


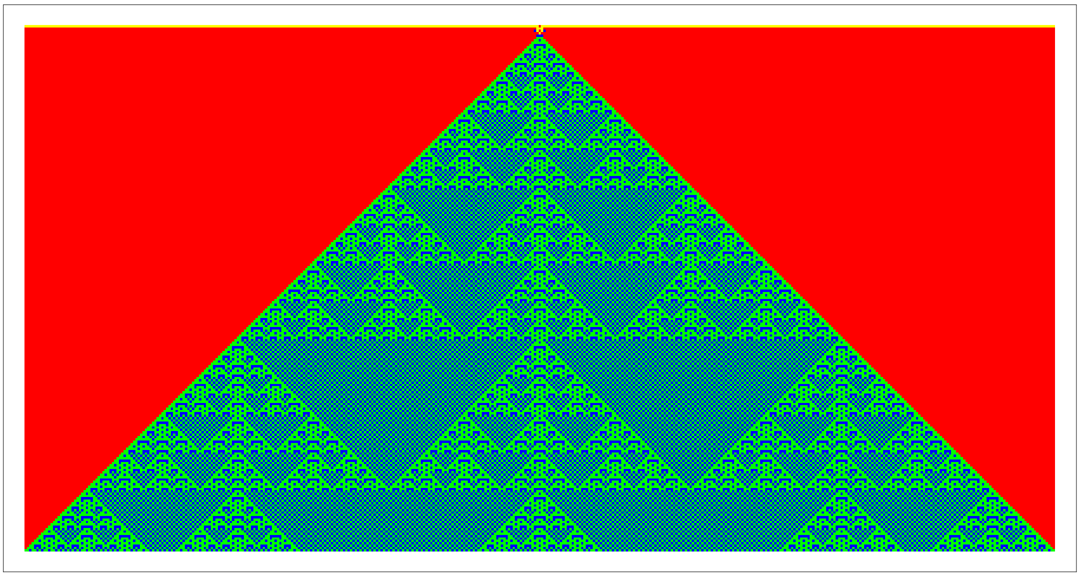


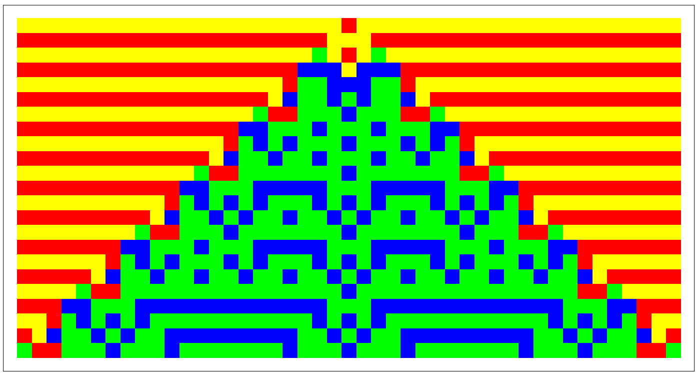


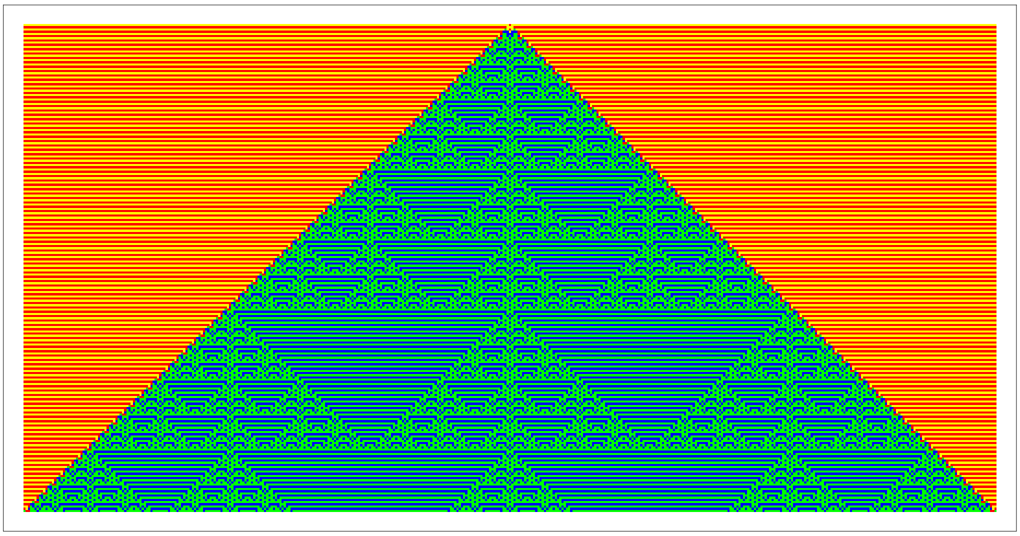


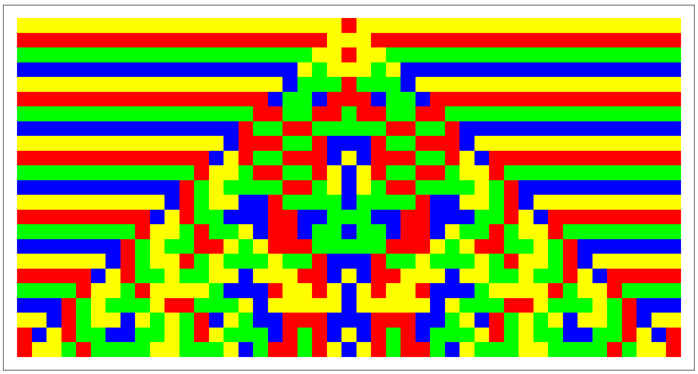


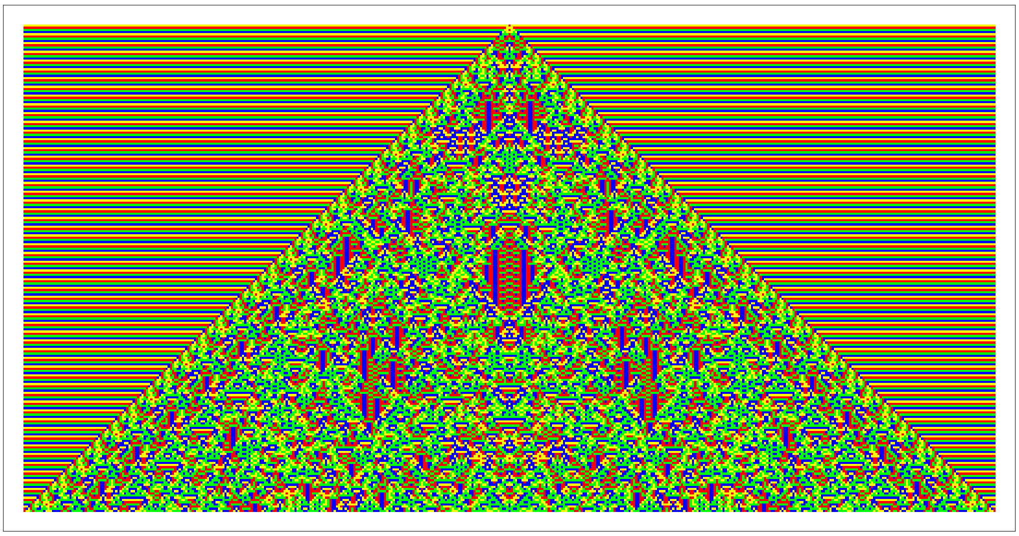


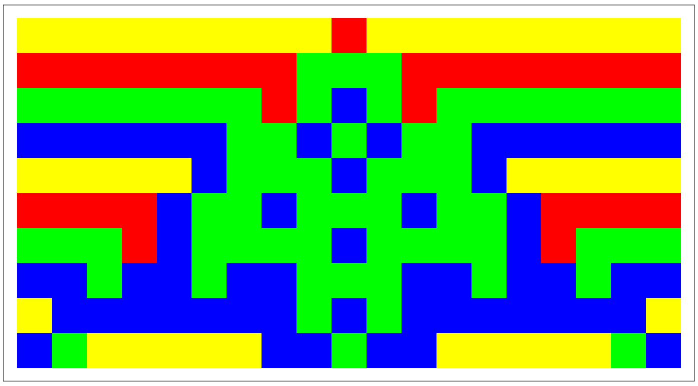


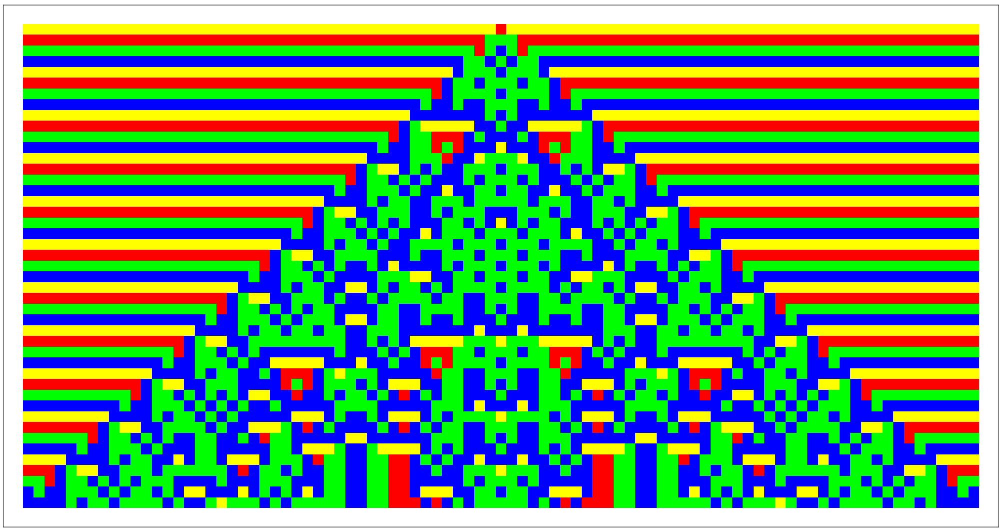


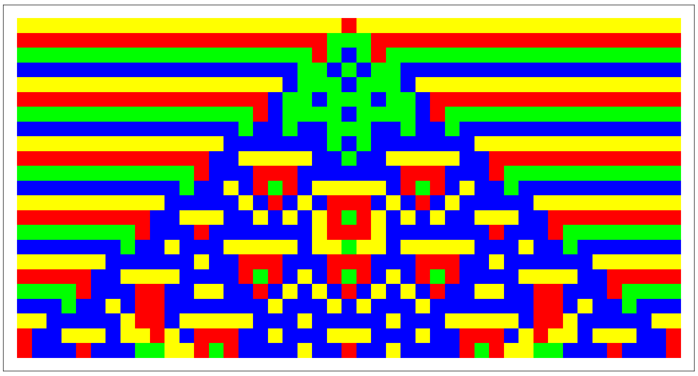


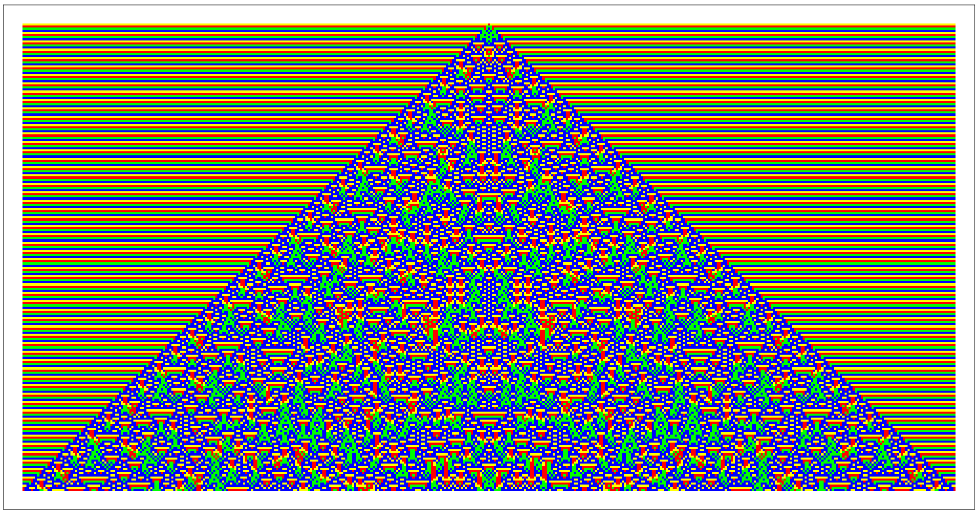




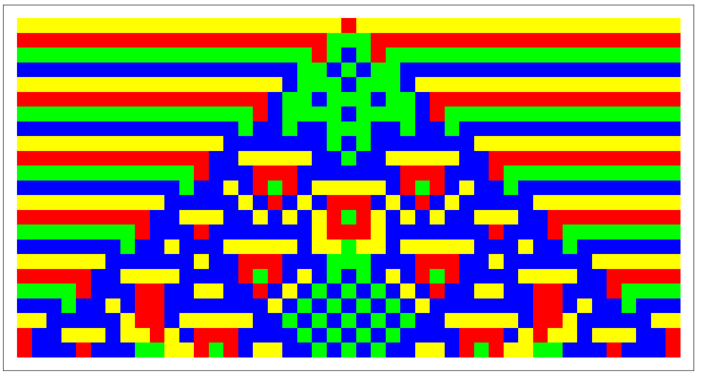


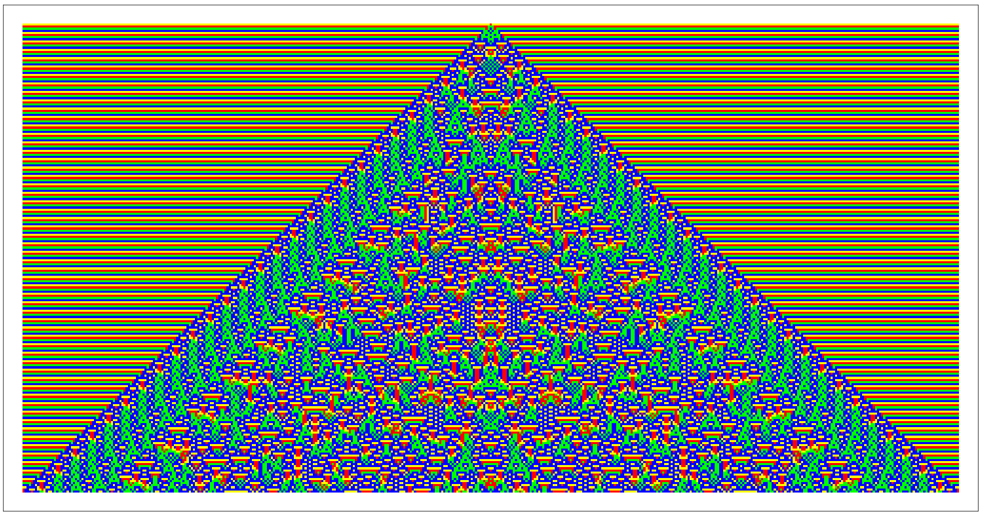


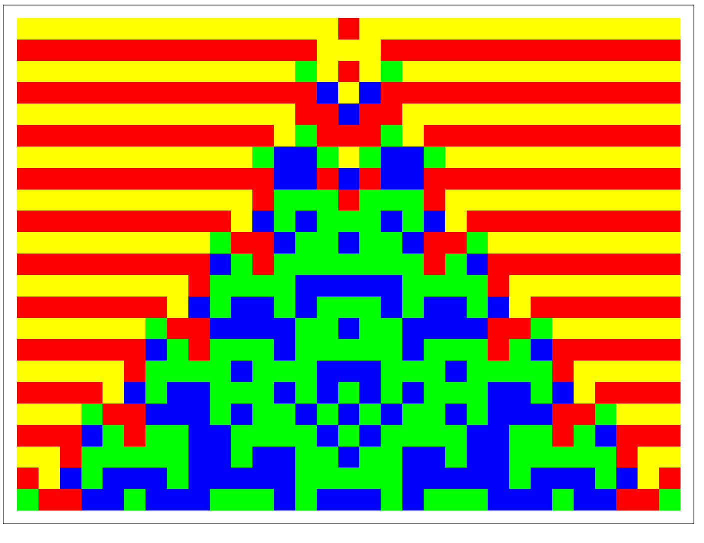


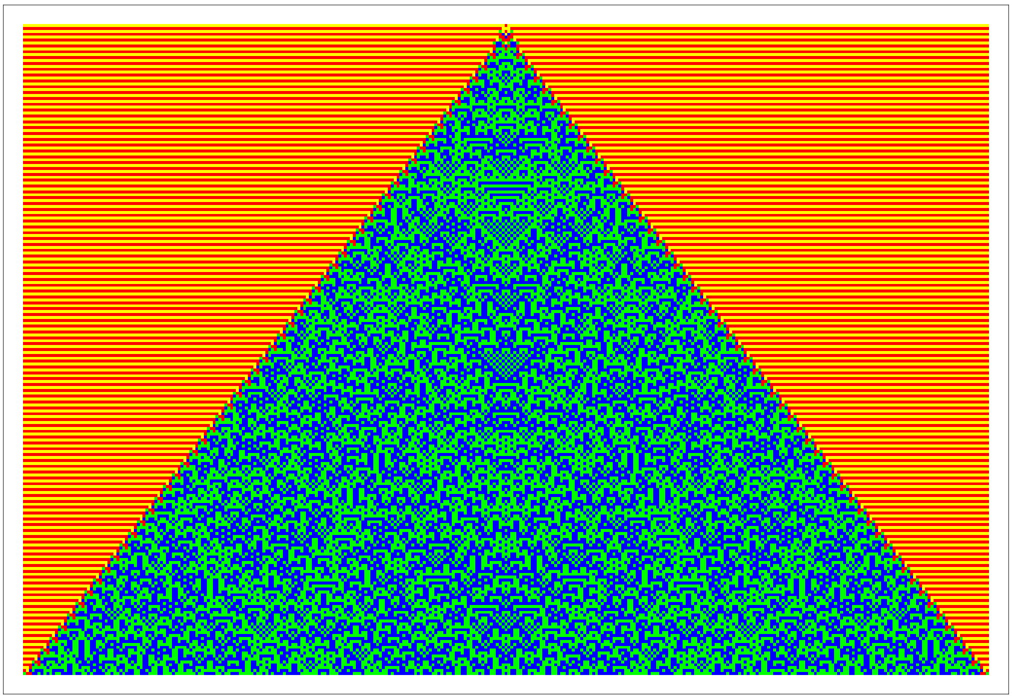


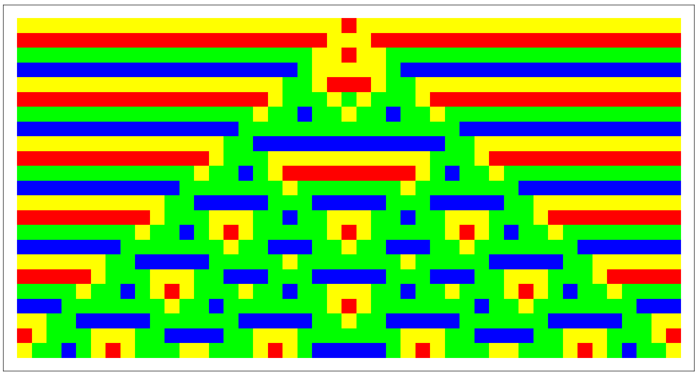
correction, two

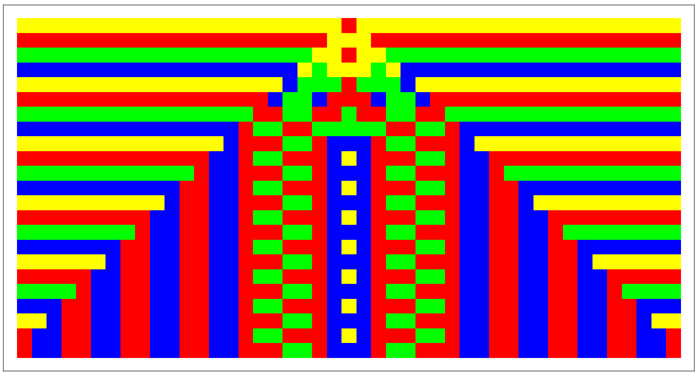


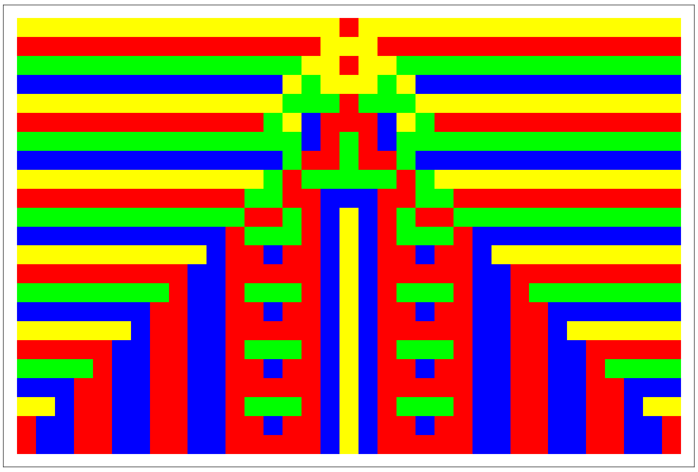


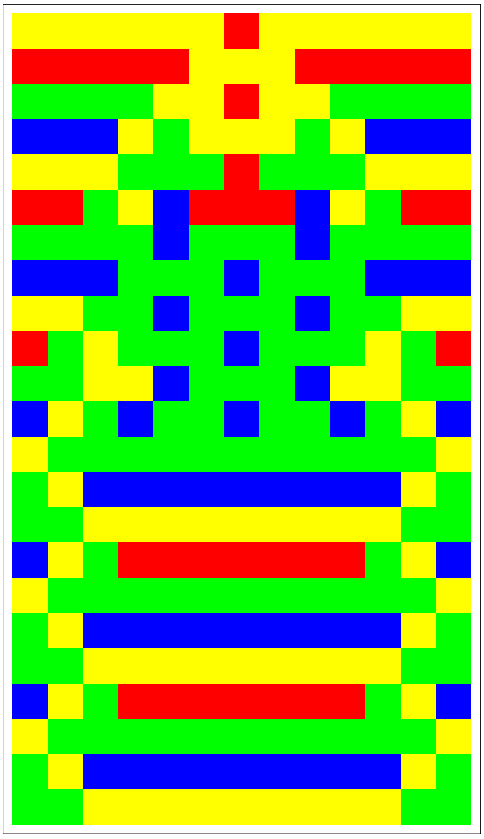


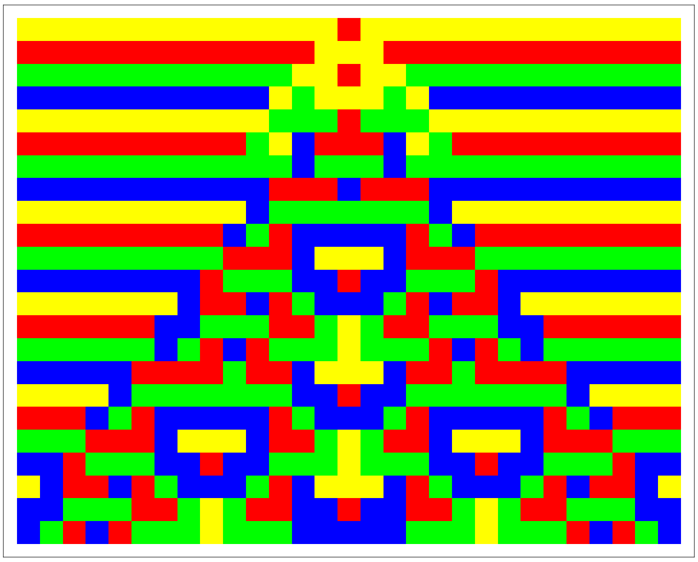


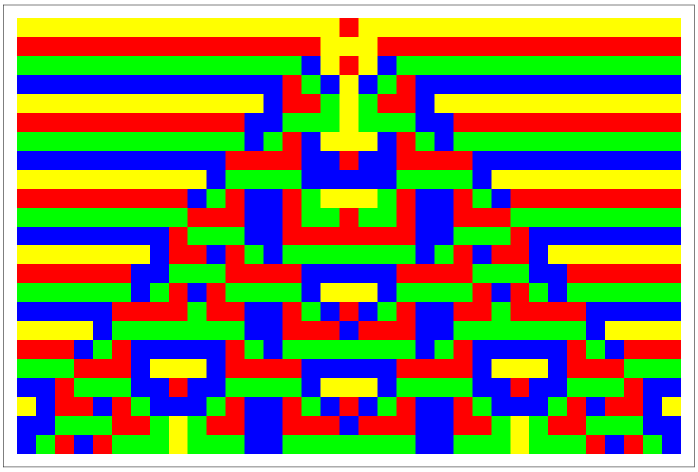


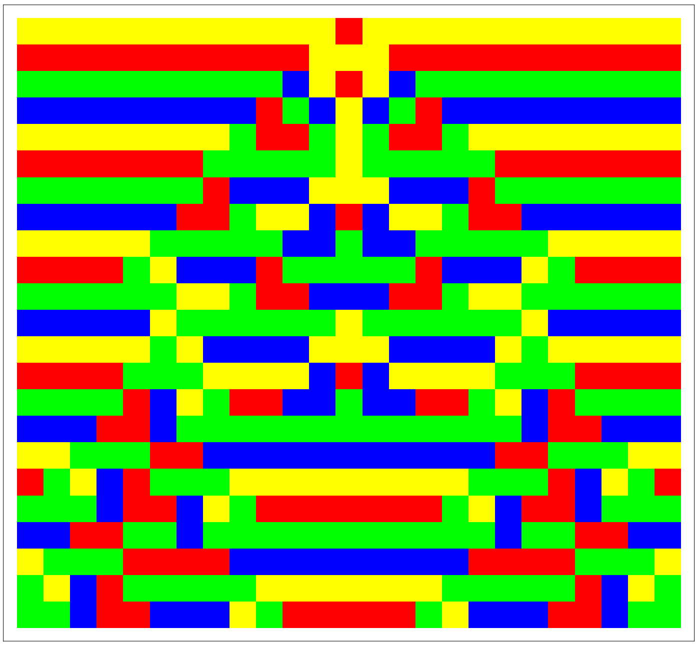




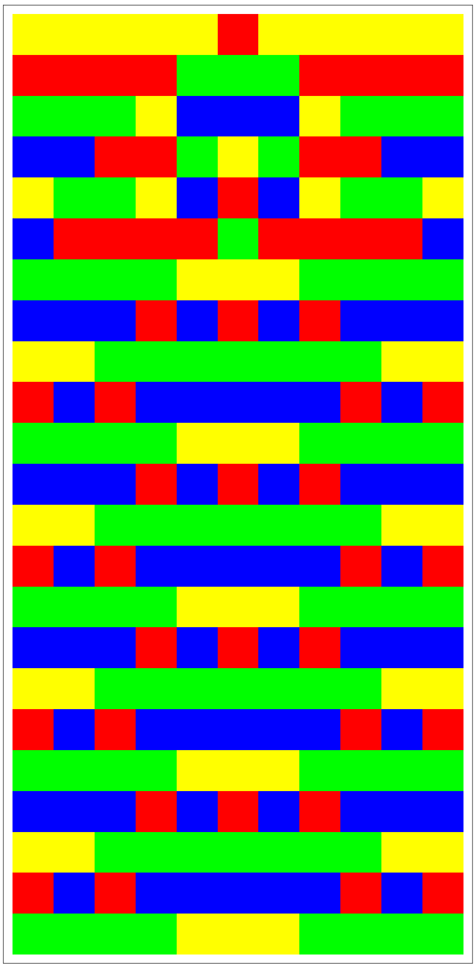


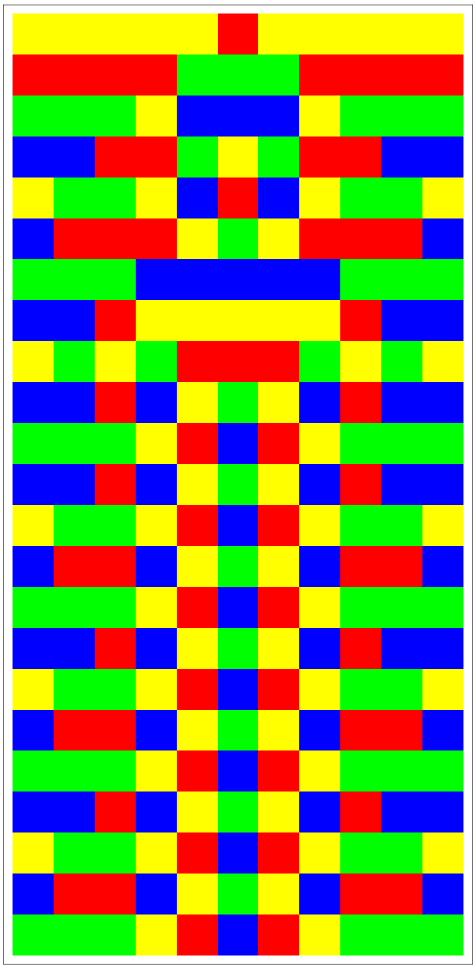




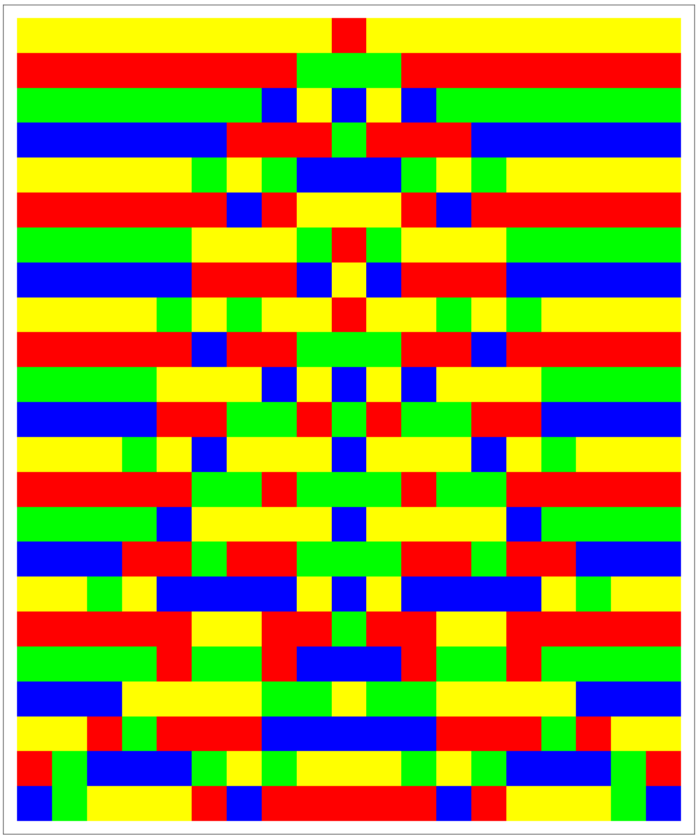


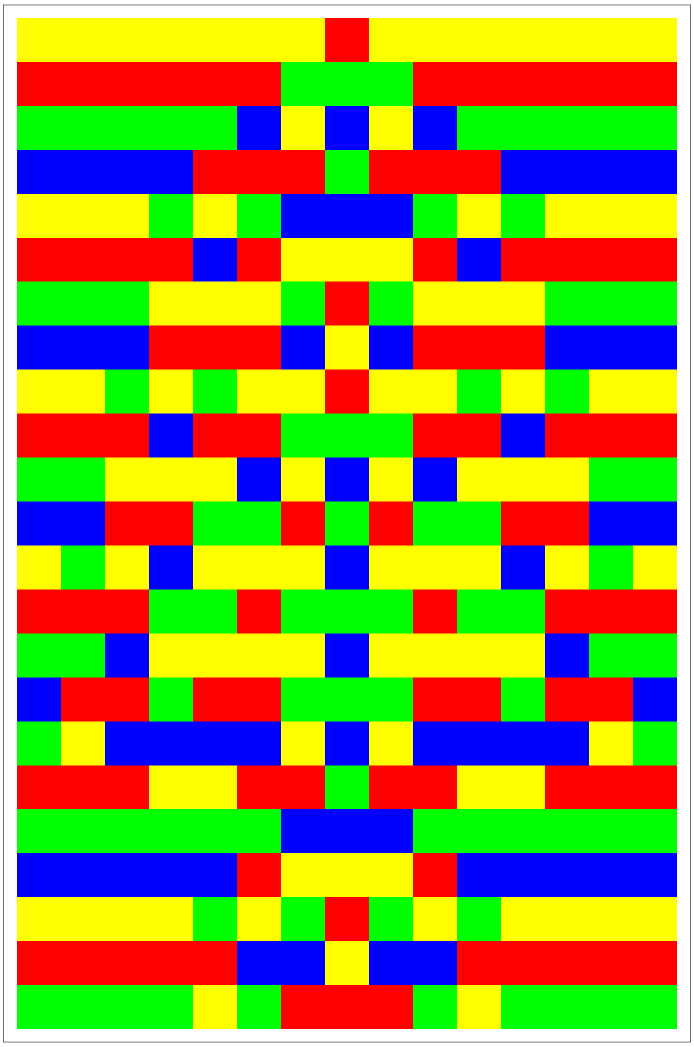
Total trans-contextural patterns

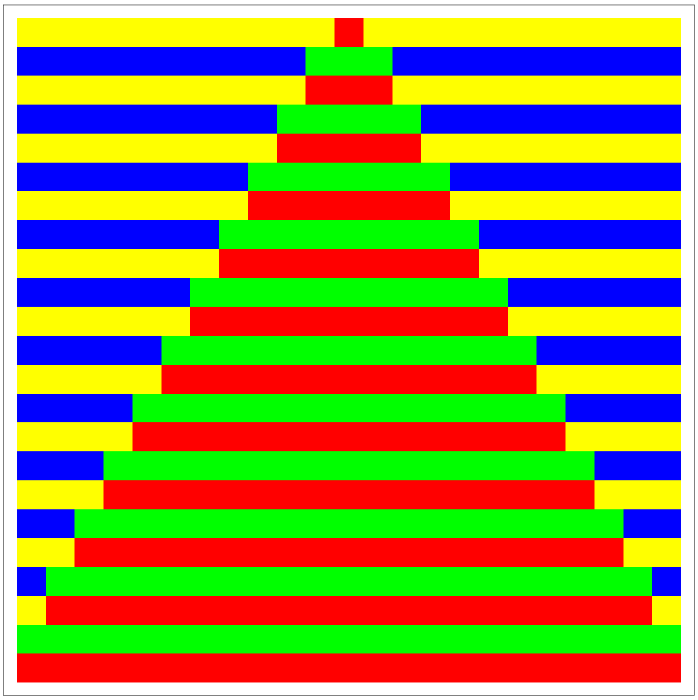


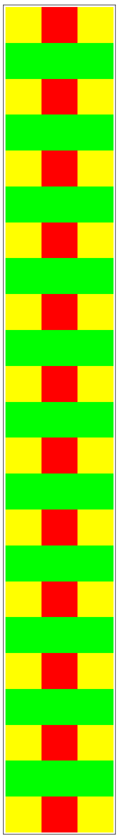


alternative transcontextural



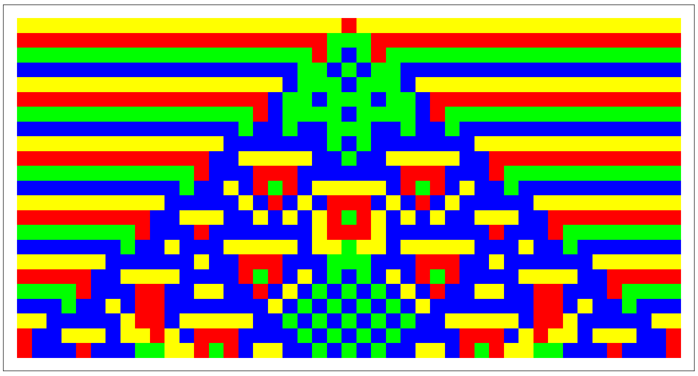




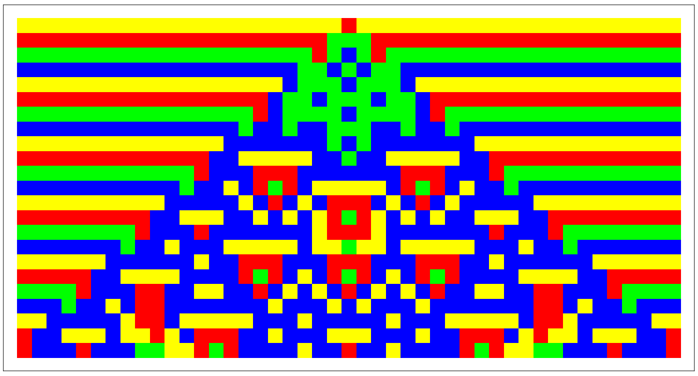


Deviant forms, overcomplete

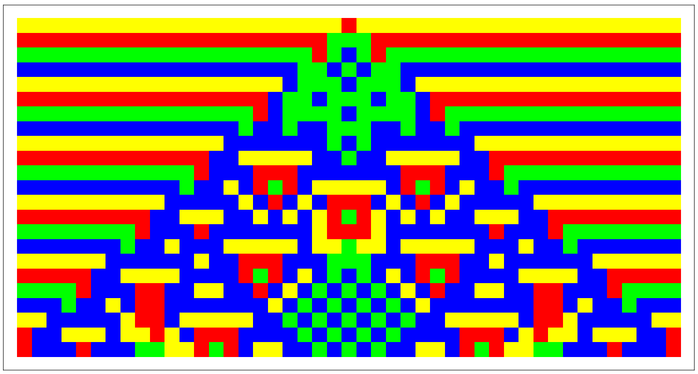
deviant



correction, one



correction, two = first



Examples of complete forms: Second: Asymmetric indCA$^{(3,4)}$??

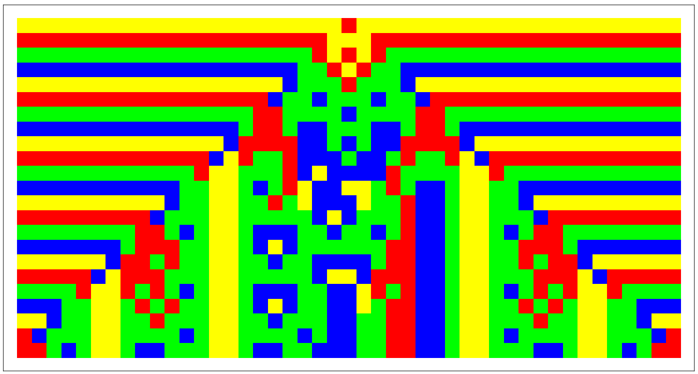
It seems that there are no asymmetric patterns for CA$^{(n,m)}$, n = 3, m ≤ 4.
IndCA$^{(3,2)}$ is trivially symmetric. This corresponds to the Laws of Form.
IndCA$^{(3,3)}$ is in a not trivial sense symmetric (this has to be proven).
IndCA$^{(3,4)}$ is structurally symmetric too.
It contains interesting asymmetric patterns only as defects.

Are there genuine assymmetric patterns in a non – interactional and non – interventional calculus?

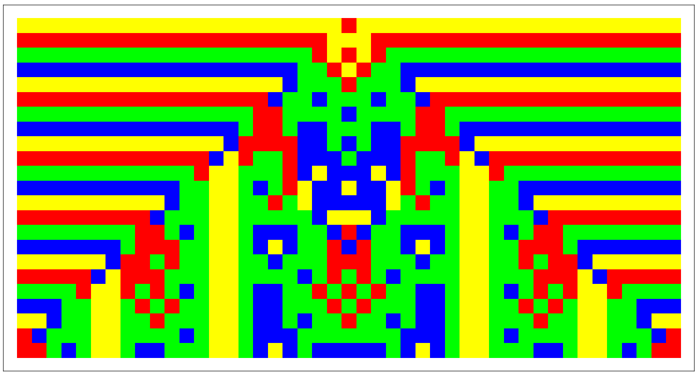
Variations of the same asymmetric pattern: defect cases

Same head

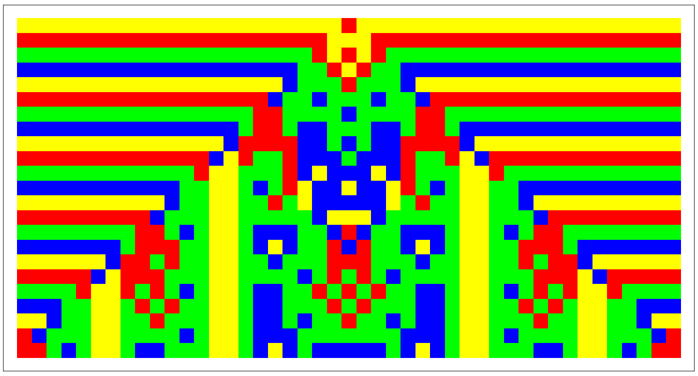
Asymmetry with defectious function :
{3, 3, 1} → 2 instead of {3, 3, 1} → 3



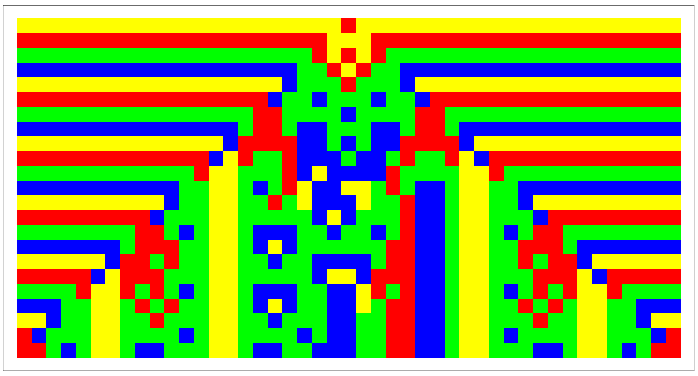
symmetry, corrected mutations



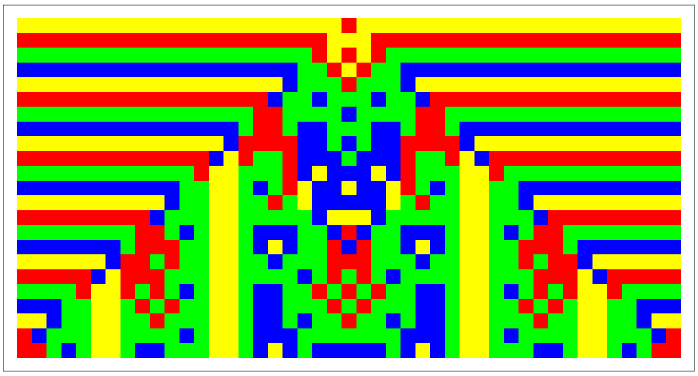
Corretion : {3, 3, 1} → 3, (*{3,3,1}→2,*)



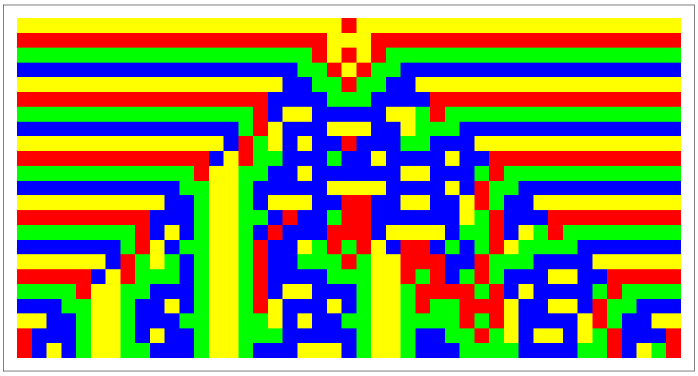
disturbed symmetry, same head



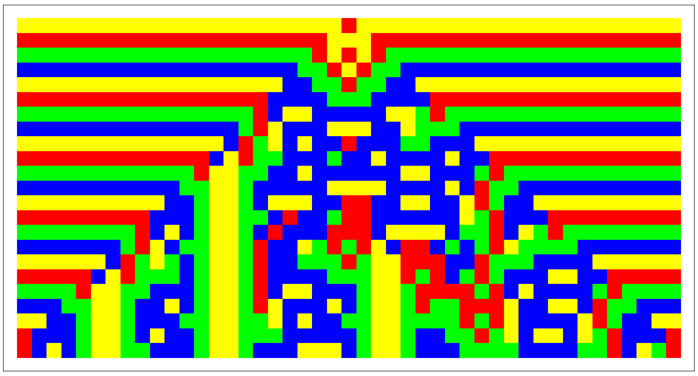
correction : {3, 3, 1} → 3, (*{3,3,1}→2,*)



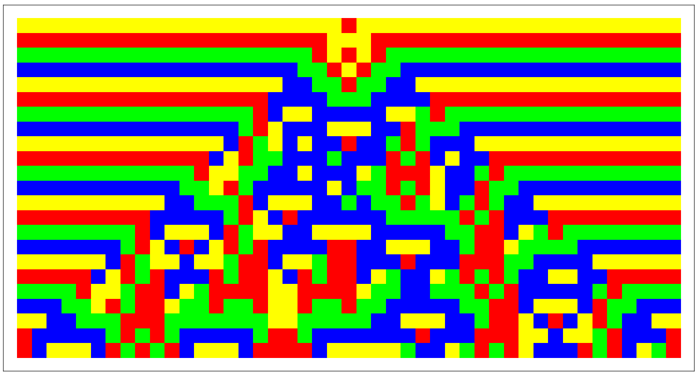
examples for same head asymmetric patterns



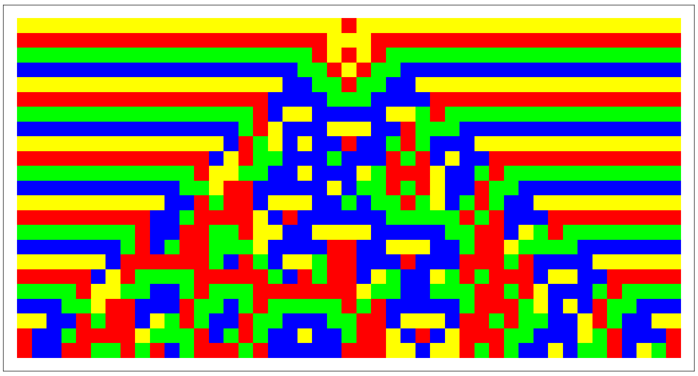
different code of example, same output

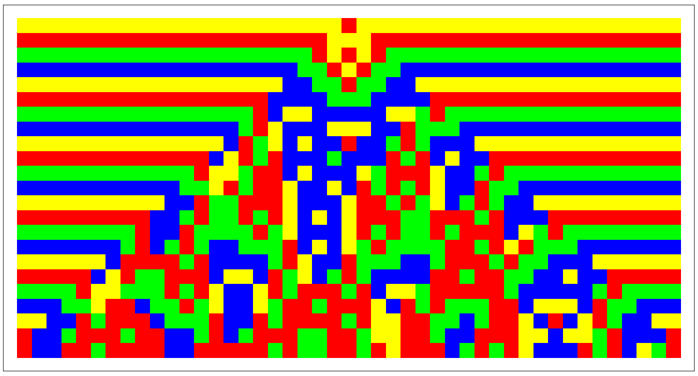


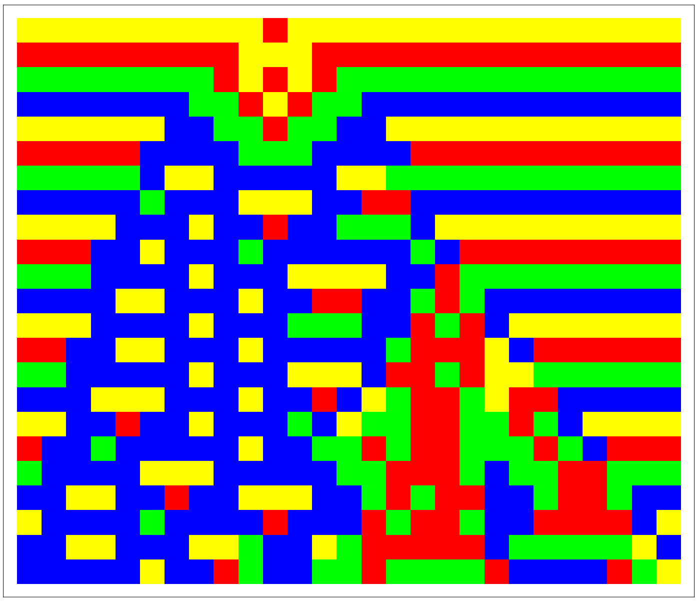
another different code



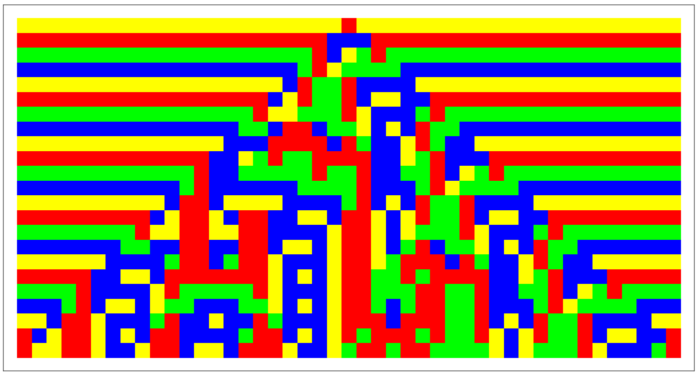
further different examples

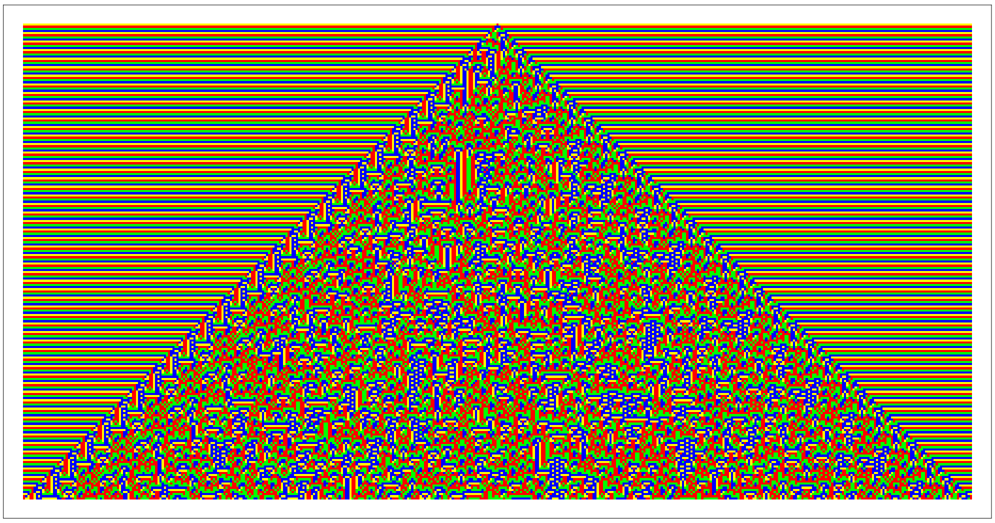




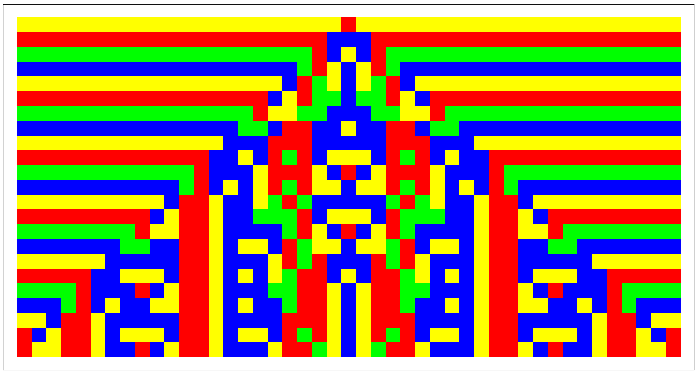


Different head
and defectuos asymmetry

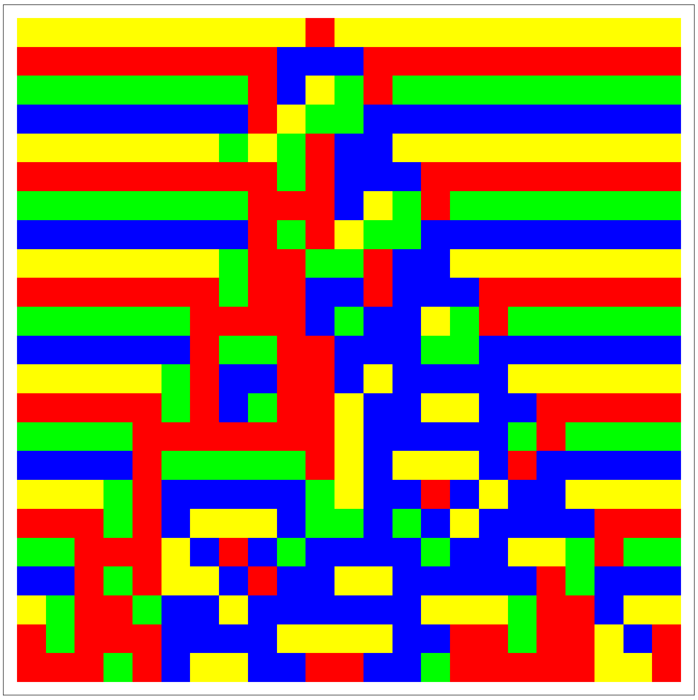




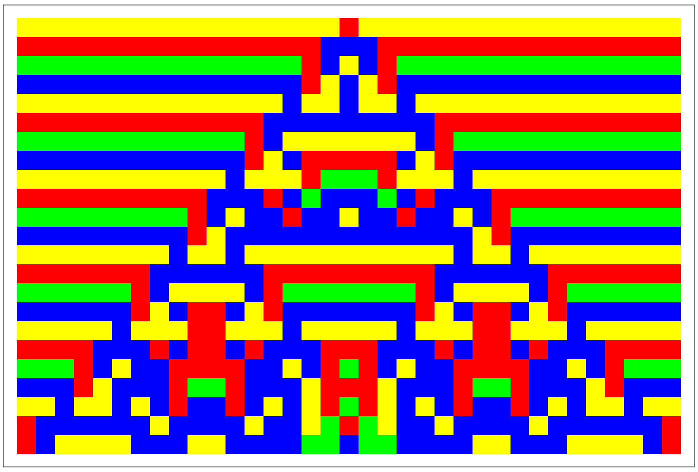
correction : {3, 3, 1} → 3, (*{3,3,1}→2,*); symmetry



defect, asymmetric pattern

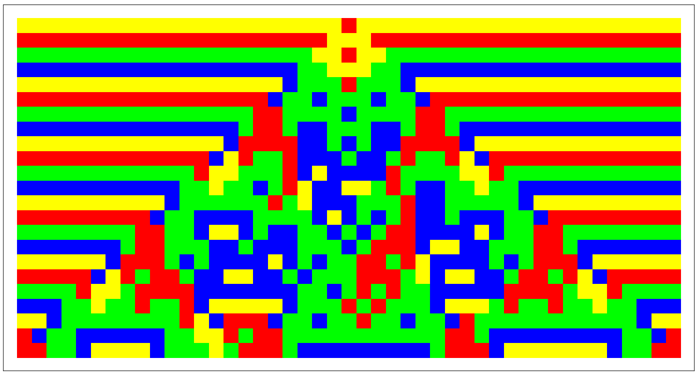


correction : {3, 3, 1} → 3, (*{3,3,1}→2,*)



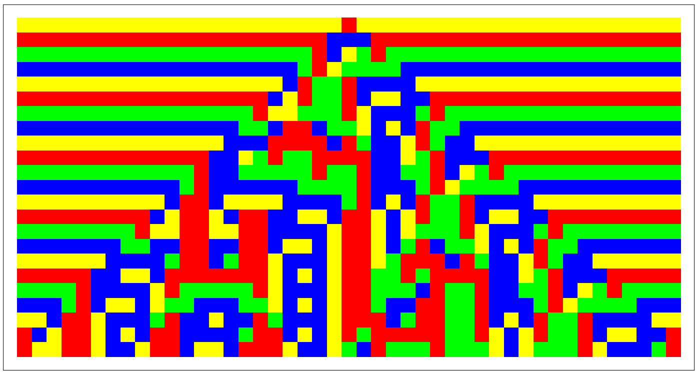
Non defect asymmetric pattern???
Overdetermination as defect

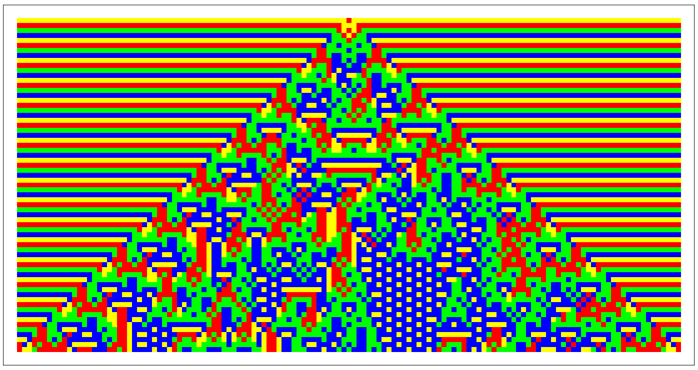
`Asymmetry` by `redundancy`



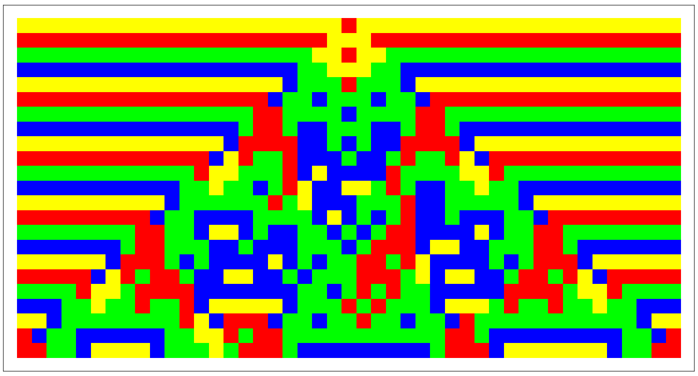
Asymmetric mutation by defect

one

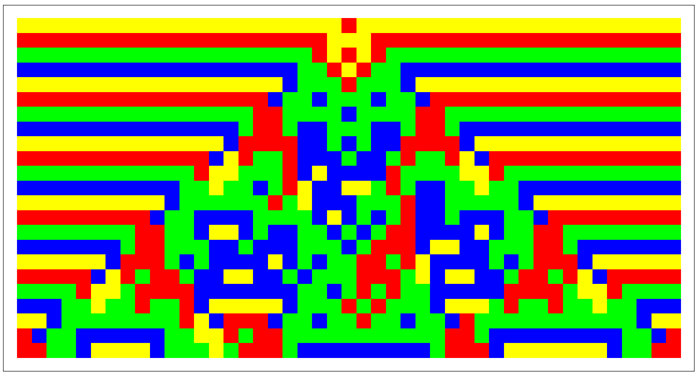




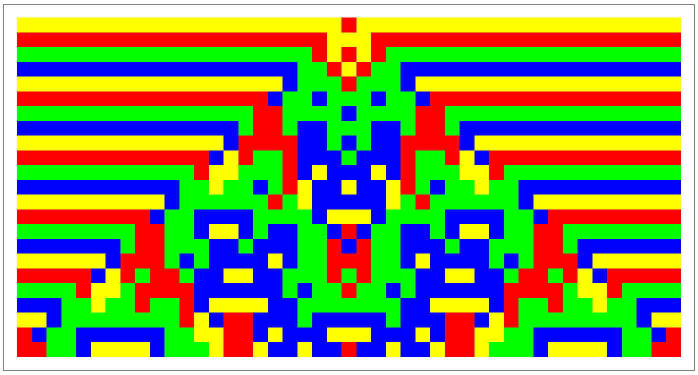
two

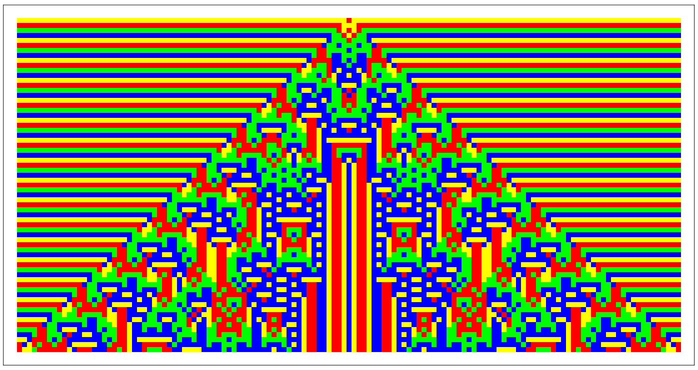


three, equal two



corrected, {3, 3, 1} → 3, (*{3,3,1}→2,*) !! !, symmetric





Symmetric variations