— vordenker-archive —

Rudolf Kaehr

(1942-2016)

Title

A Kind of a New Music Box

Rudy's PolyParadigm Sound Visualization Box PPSVB x.11

Archive-Number / Categories

3_54

Publication Date

2015

Keywords / Topics

Morphograms, Celluilar Automata, Semiotics

Disciplines

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

Abstract

Rudy's "PolyParadigm Sound Visualization Box PPSVB x.11" is a "Tool Box for sonic and visual deep-structural adventures by comparing classical, indicational and morphic CA"

Citation Information / How to cite

Rudolf Kaehr: "A Kind of a New Music Box", www.vordenker.de (Sommer Edition, 2017) J. Paul (Ed.), http://www.vordenker.de/rk/rk_A-Kind-of-New-Music-Box_2015.pdf

Categories of the RK-Archive

A Kind of a New Music Box

Rudy ś PolyParadigm Sound Visualization Box PPSVB x.11

Dr. phil Rudolf Kaehr

ISSN 2041-4358
(work in progress, vs. 0.4, Febr. 2015, Aug. 2014)

A Tool Box for sonic and visual deep-structural adventures by comparing classical, indicational and morphic CA

If words are not going to be listened to and notions are not going to be understood, there is still a chance of showing some pictures and making some noise, with an offer to motivate to enjoy or to conceive their differences and also to try to overcome the attitude of blind and deaf denial of conceptual thinking.

This little exercise concentrates on just 3 fundamentally different ways of writing as they are involved in the general theory of writing and formalization, developed in the attempts of *graphematics* which is paradigmatically surpassing the semiotics of formal and programming languages.

These 3 selected paradigms of writing, the *classical*, *indictional* and the *morphogrammatic*, are confronted with the formal concept of 1 D cellular automata (1 d CA).

There is not yet any 'deconstruction' of the classical concept of CAs as finite state machines and dynamic systems intended.

Classical CA

> Jaime Rangel-Mondragón, A Catalog of Cellular Automata
> *"A one-dimensional CA processes in parallel the simultaneous change of state of each individual cell forming a collection arranged in a line. These individual changes are performed in accordance to the states of each one of their neighbors including itself through the application of a local transition rule. It is remarkable the fact that from such a simple mechanism we are able to generate a behavior of profoundly intrincate complexity, all this in contrast to the general impression that effects are always preceded by causes of at least comparable complexity."*
> http://library.wolfram.com/infocenter/MathSource/505/CAcatalog.nb
>
> More at:
> http://plato.stanford.edu/entries/cellular-automata/supplement.html

Hence, three categories of paradigmatically different kinds of CA are studied. The first is well known and gives a contrastive background to the two newly introduced kinds of CA.

Firstly, the classical CA,
secondly, the indicational CA,
thirdly, the morphogrammatic CA.

Therefore I introduce a new kind of an epistemological *comparatistics*, i.e. a comparison of graphic and sound systems in respect to their *paradigmatical* structures based on different kinds of writing systems and their own intrinsic CA rule sets.

In a first approach, aspects of the *dynamics* of CA systems in general are studied by the presentation of their intrinsic *transition* graphs.

Transition graphs for classical CA had been introduced and exhaustively studied by Andrew Wuensche. The used implementation in this paper utilizes the published version of Stephen Wolfram.

"Cellular Automaton State Transition Diagrams" from the Wolfram Demonstrations Project
http://demonstrations.wolfram.com/CellularAutomatonStateTransitionDiagrams/

Furthermore, the proposed approach to a formalization of morphogrammatic CA is still more a *simulation* than a genuine implementation of the very concept of morphogrammatic CA.

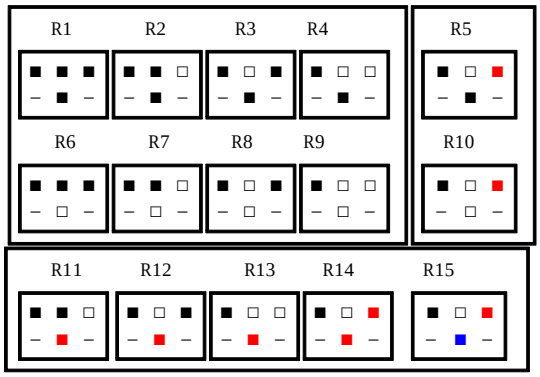
The *classical* paradigm is well studied, mathematically and philosophically, and got a decisive elaboration with the Opus Magnum of Stephen Wolfram's *A New Kind of Science* (NKS).

Here, I refer just to some 1D CA examples within the set of CA rules. Accessible by *special CA* of the CA box.

A special topic of compararison is the representation of the single archetypical figures of *Sierpiński triangles* in the 3 modi of formalization (classical, indicational and morphic).

RuleBox of the ReLabel-based rules

MorphogrammaticRuleDefinitions



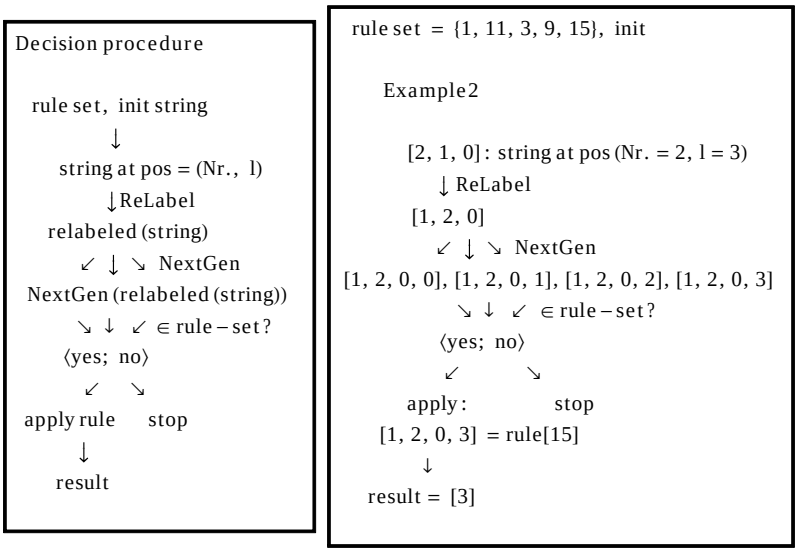
Relabeling

StaticMorphoRuleDefinitions

DynamicMorphoRuleDefinitions

DynamicMorphoRuleSchemes

Diagram of the separation procedure



http : // memristors.memristics.com/CA - Compositions/Memristive %20 Cellular %20 Automata %20 Compositions.pdf

Further explanation of the procedure

Given a morphoCA$^{(m,n,l)}$ defined by its complexity, how does it work?

The automata defined by morphoCA$^{(m,n,l)}$ are at first closed systems entailing their internal rules defined by the morphograms and the supposed complexity.

A classic automaton is a machine with input and output.

A classical CA is defined by its *rules* and by its *elements*. The elements are as much pre-given as the rules. Both are mathematically ternay functions over an alphabet, typically of two elements, 0 and 1.

In contrast to a classical CA, the elements of a morphoCA are not pre-given. What is given, i.e. specified are the rules conceived as morphograms and the complexity of its compositions.

Morphograms are playing a double role: they are *rules* and *data*, i.e. operators and operands, at once. In this sense, morphograms are functioning as a double look-up device for rules and data together.

With that, a morphoCA has no output yet. Simply because it has no informational input to manage.

Hence, a morphoCA has to be understood as being embedded in an environment that contains itself all kinds of automata. Such an environment is enacting data as a disturbance of the closed morphoCA.

Off whatever kind of elements or patterns of elements the disturbance consists, the disturbance has to be interpreted by the automaton according its own internal structuration by the complexity of the composition of its morphograms that are the deep-structural rules and patterns of the morphic automaton.

Therefore, if a disturbance, realized by a pattern of elements occurs, the machine has to decide if this structured disturbance corresponds to an internal rule/morphogram system or not.

Such an interpretation, necessary to decide an application of an internal rule is realized by a 'normalization' function that is defined on different levels of organization/abstraction.

On the trito-level of abstraction which is constitutive for morphoCAs, the normalization function is realized by the so called ReLabel procedure for a linearly ordered interpretation of events or the e/v-abstraction for the general case.

If a disturbance occurs as a string of whatever elements at a position of the 'history' of the previous activities of the automaton, it has to be checked if the string/pattern corresponds structurally to a rule/morphogram of the morphic automaton.

This is done 1) by a *relabeling* of the semiotic event and 2) by a *comparison* of the relabeled/normalized pattern with the existing morphograms that are involved in the particular morphoCA.

If there is a correspondence to a morphogram, the procedure of the automaton can continue with an implementation of the new string/pattern which is delivered by the disturbance.

At first it is convenient to use the normalized pattern instead of the original pattern that might contain very different signs than the ones that had been in the previous history of the automaton.

Therefore, if e.g. a pattern like [2, 1, 0] appears, the relabeling produces a normal form and this normal form of the string can be compared with the existing morphograms of the rule-set of the particular morphoCA.

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

```
ReLabel[{5,7,a,b,■,©,ESC,£,■}]
```

```
{1,2,3,4,5,6,7,8,5}
```

ReLabel[2,1,0] delivers [1,2,0] and this corresponds to the head of some morphograms , i.e. morphic rules.

The application of the NextGeneration rule produces a set of successors of the pattern:
NexGen[1,2,0] = {[1,2,0,0],[1,2,0,1],[1,2,0,2],[1,2,0,3] }.

Hence, there are 4 possible continuations offered by the morphic successor operation, NextGen, on the 'incoming' pattern [1,2,0].

In other words, there are no sign events that are not accepted in principle by a morphoCA.

Classically, if a CA is defined over the set {0, 1}, any other sign sequence that is not defined over this set gets will not be recognized. This lack of recogntion is not yet a active rejection.

The decision of the machine which one of the 4 possible successors is chosen depends on the definition of the specific morphoCA.

From a general point of view all possibilities are accepted as new applications of the implemented rules.

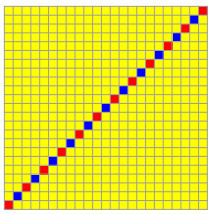
In concreto, a procedure of a machine is defined by a finite set or system of non-conflicting rules.

If the machine *morphoCA* is defined, e.g., by the rule set = {1,11,3,9,15}, i.e. the ruleM[{1,3,9,11,15}], then just the successor [1,2,0,3] of the previous [1,2,0] input is accepted by the machine. It corresponds to the rule [15] of the machine definition ruleM[{1,3,9,11,15}]. Therefore it fits into the rule definition of rule [15] and is accepted for an application of rule [15] delivering a prolongation as a next step in the development of the algorithm of the machine.

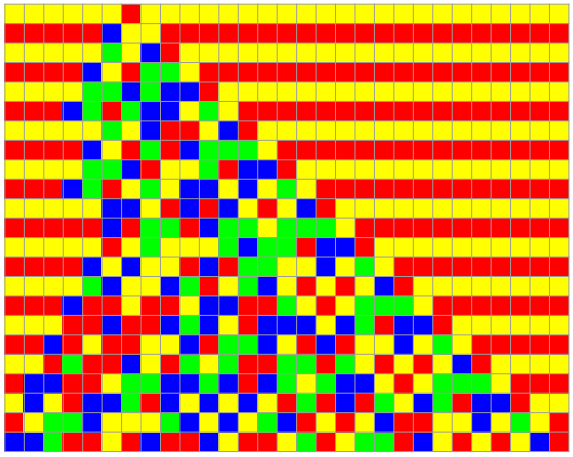
This prolongation of the CA process is defined by the application of *rule[15]* that is generating the element "3" on the constellation of the new level of the process.

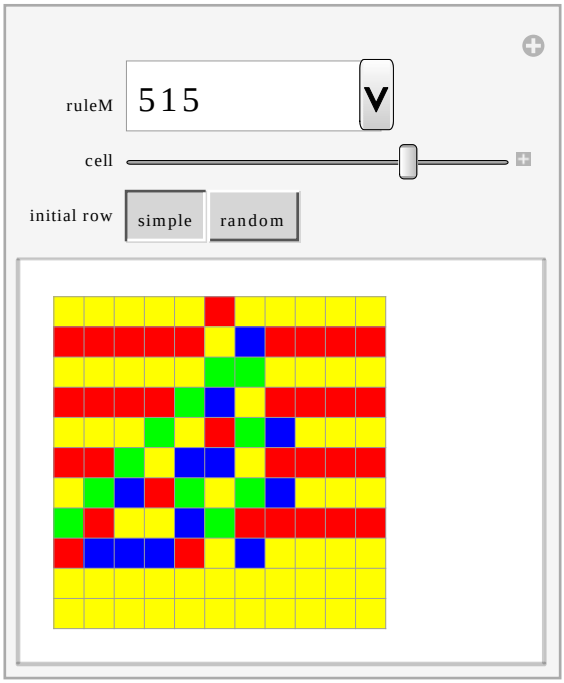
morphoCA example

```
ArrayPlot[CellularAutomaton[
 ruleDM[{1,3,9,11,15}],
 {{1},0},22],
ColorRules->{1->Red,0->Yellow,2→ Blue, 3→ Green},
ImageSize→400,Mesh→ True]
```



```
ArrayPlot[CellularAutomaton[
 ruleDM[{6,3,9,11,15}],
 {{1},0},22],
ColorRules->{1->Red,0->Yellow,2→ Blue, 3→ Green},
ImageSize→400,Mesh→ True]
```





With this quite complicated maneuver that builds the "Vorwissen" of a computation, two crucial aims are achieved.

First, an abstraction from any sign repertoire.

There is no need to implement into a morphoCA a set of pre-given elements that are defining the range of the rules as a sign repertoire.

Second, interaction of *acceptance* and *rejectance* of environmental events.

All elements, patterns, events of an *interaction* or disturbance of a machine with/by its environments can be handled by the machine as the modi of *acceptance* or *rejectance* of the physical or semiotic event.

A sign event from an environment gets rejected by the machine if it doesn't fit into the rule set of the actual machine.

Nevertheless, the rejection is not denying the status of the sign event as a reasonable object of perception. It just doesn't have the form to be accepted to be involved into further computation by the actual machine and its

rules..

Hence, there is therefore still the possibility left that the rejected event gets saved into an *internal* memory that is not the memory of the actual machine and might be used simultaneusly by another machine or for later applications by a further machine. Also the mentioned *inner memory* is an external memory for the actual running machine it nevertheless belongs to the complex machine as such.

To contrast the situation to the classical model it is clear that an event or an information that is not accepted by a classical machine simply doesn't exist for the machine at the time of the interaction. It might be classified as *disturbance* in a negative sense.

The categories of acceptance are:
1) semiotic,
2) morphogrammatic (trito-level),
3) deutero-grammatic and
4) proto-grammatic.

Other possibilities to categorize sign events in general are a) the *indicational* and b) the *Mersenne*-abstraction of events.

A rejectance might be utilized to change the structural definition of the existing machine. Hence this would enable the possibility of a structural evolution of the machine that will change the rejectance to an acceptance of the events.

In contrast to the interactional and operational approach, the papers published up to now are not generating their data and their rules but are depending on a pre-given set of data and concrete rules over the data. Thus, this approach is based on pre-defined look-up tables and is not yet *interactively* producing its operational outer and inner environments.

Those rules are defined abstractly as general functions in a traditional way. Only in the case of applications, the functions are defining the concrete rules for the given formal context.

As an example, the *deuteroRule* dckd[{11122}] is defined by the following function. The realization of the deutero rule dckv[{11122}] for a constellation of 5 values is given by the list of concrete functions.

The same function, dckd[{11122}], might be embedded in an environment with a higher or with a lower complexity depending on the existing embedding of the rules.

Thus, any occurrence of the listed concrete rules is accepted by the morphoCA in question. Other constellations remain rejected until the range of the rule is adjusted.

Look-up is depending on a selected alphabet and the functions over it.

The *operational* approach is not depending on a pre-given set of rules, organized as a look-up table, but is operationally creating its operands over any symbolic input.

This happens on a 'look-up' list of meta-rules.

The meta-rule for *dckd[{11122}]* has 4 entries and is producing 80 functional rules that are applied for the automaton.

```
dckd[{11122}] =
Flatten[{
Table[
{
Mod[{i,i,i,i+j},5] → Mod[ i+j,5],
Mod[{i,i,i+j,i},5] → Mod[ i+j,5],
Mod[{i,i+j,i,i},5] → Mod[ i+j,5],
Mod[{i+j,i,i,i},5] → Mod[ i+j,5]
},
{i,5}, {j,4}]}]
```

A shorter example is defined by just one meta-rule that is defining 20 rules for the definition of *dckv[{11112}]*.

```
dckv[{11112}] =Flatten[Table[Mod[{i,i,i,i},5] →Mod[ i+j,5],{i,5}, {j,4}],1]
```

On the level of a functional implementation of the morphic cellular automata, like in this paper, the genuine concept of morphic *interaction* is replaced by a corresponding set of 'pre - given' functions in *a look-up table* of

functions.

Albeit the presented implementations of morphoCAs are based on the meta-rules and the functions produced by the meta-rules the intended *interactional* implementation is not yet fully realized.

That means, that such a situation is understood as non-interactional and not yet involved in any disturbance, interpretation and codification. The automata are simply running an the base of thir given rules as they are defined.

For questions of modeling and constructing living systems, this is a very strong restriction of possible dynamics that has to be surpassed.

The *look-up* concept relates to *memory*, while the operational approach is producing its concrete 'operands' by the application of its operators.

What defines of morphoCA is a set of *meta-rules* (operators), symbolized by morphograms, and not a set of values and functions. Certainly, the meta-rules have to be stored. With that a kind of look-up list for meta-rules has to be implemented.

If the realized rules are memorized by the machine, this memorization can be utilized secondarily as a *look-up table*. With that, the look-up table approach is not denied but has not to be presumed for the functioning of the machine. The machine is producing its own look-up table.

This sketch of the concept of morphic cellular automata opens up some interesting questions that are not yet answered. And which probably don't occur in classical context.

Where are the input-elements of the disturbance from?

How is the topology of the automaton affected by such structural interactions?

Obviously, classical CAs are interaction-free. Therefore they don't have an environment that could disturb its architecture. Interactions between 'autonomous' CAs are not included in the framework of CAs.

CAs may model the interaction between traffic entities but they are not enabled to model conceptually the interactions between CAs.

The common operations on and of CAs, like union, products etc. of automata, are not touching their very concept of 'interaction-free' closeness.

Helpful insights and concepts to the approaches to a new paradigm of interactional/reflectional computing can be found in the work of the cybernetician Gordon Pask (Pangaro).

Motivation and definition of the rules

Epistemological orientation

On the level of a functional implementation of the morphic cellular automata, like in this paper, the genuine concept of morphic interaction is replaced by a corresponding set of 'pre - given' functions in *look-up table* of functions.

The look-up concept relates to memory, while the operational appraoch is producing its 'operands' by the application of the operation.

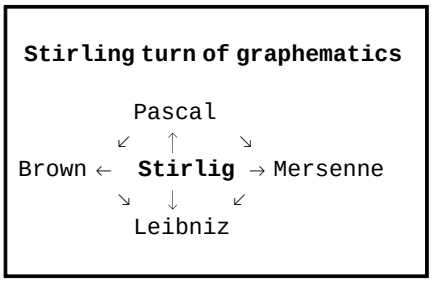
Look-up is depending on a selected alphabet and the functions over it.

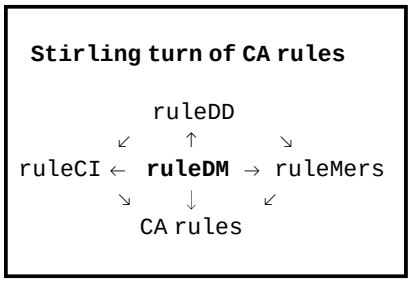
The *operational* approach is not depending on a pre-given set of rules, organized as a look-up table, but is operationally creating its operands over any symbolic input.

What is presumed is a set of *rules* (operators), symbolized as morphograms, and not as a set of functions.

Hence the system of morphograms is presented as a system of operations.

Paradigmatic system of CA rules





The Stirling turn indicates a reversion/subversion of the known logical order of semiosis. In this conceptual graph, semiotics turns from the fundamental center - position to a derived mode of writing with trito - grammatics playing the role of the initial position of the graphematical writing system.

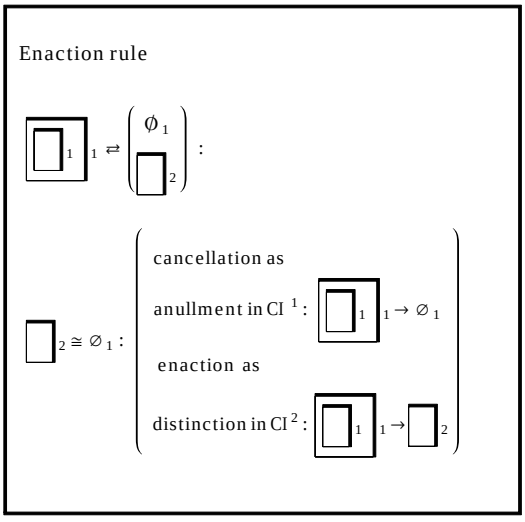
ruleDD : deutero – grammatic rules of morphic CA, (not included in this paper)
ruleCI : indicational CA rules, CI, CIR, CIRT,
ruleM : morphogrammatic rules of the trito – level, M, DM, N, MNP, M42, (static version)
ruleDM : morphogrammatic rules of the trito – level, M, DM, N, MNP, M42, (dynamic version)
ruleMers : Mersenne rules (not included),
ruleCA : classical CA rules.

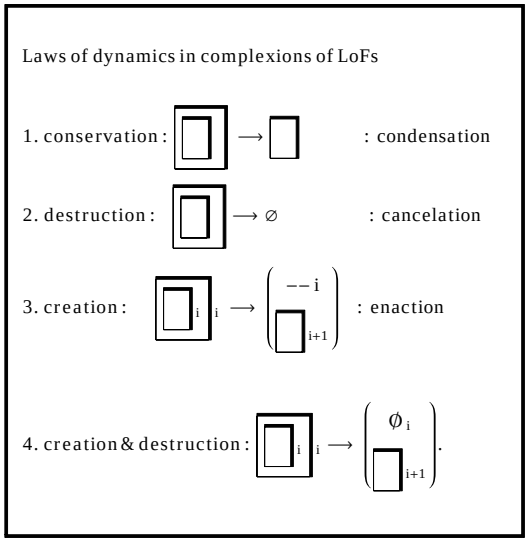
Laws of Form

The *indicational* way of writing and thinking, introduced by George Spencer Brown (GSB) with his *Laws of Form* (LoF), published 1969, is not specially well elaborated and got a very controverse reception from the side of mathematicians and logicians.

One of the most crucial intuition of the CI is the "*topology invariance*" of its terms of distinctions. Therefore, the indicational CA introduced here are based on this "*topology invariance*" of the distinctions only and are not yet including the internal CI rules of the Laws of Form (condensation, cancelation).

A distinction of a distinction is conceived in the *reflectional* CI, i.e. CIR, and CIRT, as both at once: as *annullment* and as *reflection* (enaction). Therefore, *annulation* is eliminating and destroying distinctions while reflection as *enaction* is not only creating new distinctions but also a new domain, i.e. world of distinctions, in which the new distinction and its further applications is realized.





A calculus of the *formation of forms* , i.e. the form and process of *structuration*, might include at least the 4 principles of distinction dynamics :

1. conservation,
2. destruction,
3. creation,
4. creation & destruction.

http : // www.thinkartlab.com/pkl/media/Diamond %20 Calculus/Diamond %20 Calculus.html

Morphogrammatics

The *morphogrammatical* way of writing and thinking, introduced by Gotthard Gunther with his *"Cybernetic Ontology and Transjunctional Operations*,(1962), is as such well elaborated by the publication *"Morphogramatik"*, Kaehr, Mahler (1993), and got a some controvert reception from the side of mathematicians, logicians, semioticians and cyberneticians.

Morphograms had been introduced by Gunther as pre-logical patterns (morphé) of trans-classical logics. As pre-logical and pre-semiotic figurations morphograms are in fact *figures* and *rules* at once, i.e. as figurations or structurations. The dynamic aspect of morphograms as rules is implemented in the paradigm of morphoCA.

In his theory of cybernetic self-reflection, Gunther classified the 15 basic morphograms into 3 classes:

1. Acceptance: *objective* reflection represented by the morphograms $MG^{(4,2)}$,
2. Rejectance: *subjective* reflection represented by the morphograms $MG^{(4,3)}$,
3. Refutation: *self-reflection* as reflection on 1) and 2) by $MG^{(4,4)}$.

In the context of cybernetic studies of dynamic systems, this classification had also been interpreted in the 1960s by Henz von Foerster as order principles.

Cybernetic Dynamics
1. Order from order,
2. Order from disorder,
3. Order from noise (order and/or disorder).

A new graphematic principle of order and creativity might be added as:
4. Order from (neither order nor disorder).

Because this cybernetic research into automata theory happened in the early 1960s, it wasn't mainly computer supported.

Definitions

How GSB sounds might be listened at the register CI, is orchestrated by *ruleCI*.
The further CI concepts, CIR, RCI and CIRT, had been freely introduced by the author in previous papers.

The rule set CI, defines the objectional definition of the Calculus of Indication (CI), a reflection on CI generates a second-order CI, i.e. the RCI, with its genuine new rules.
Further reflectional thematizations of the CI generates a third-order calculus RCI, and its augmentation CIRT.

Indicational CAs are thus divided into:
a) first-order indCA (in the sense of the Calculus of Indication (CI), ruled by the set of *ruleCI*,
b) as second-order (enactional) rules *ruleCIR*,
c) as third-order rules *ruleRCI*, and
d) the rules of *ruleCIRT*.

Rule schemes for indicational CA

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

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}]}]

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}]
  }]

ruleCIRT[{a_}] :=
  Flatten[{
   ruleCIRT[{a}]
```

```
}]
```

Morphogrammatic CAs are divided into:

a) classical morpho CAs, $CA^{(3,2)}$, ruled by *ruleCl*, with complexity 4,

b) trans-contextural CAs, $CA^{(3,3)}$, ruled by static *ruleM*, with complexity 5,

c) trans-contextural CAs, $CA^{(3,3)}$, ruled by dynamic *ruleDM*, with complexity 5,

d) trans-contextural CAs, $CA^{(3,4)}$, ruled by *ruleMN*, with complexity 6 and

e) trans-contextural CAs, $CA^{(3,4)}$, over-dtermined, ruled by *ruleMNP*, with complexity 7,

f) trans-contextural CAs, $CA^{(4,2)}$, over-determined, ruled by *ruleM42*, with complexity 8.

Cl rule set = [1,2,3,4,6,7,8,9} over 4 places,
M rule set = { CL ⋃ {5,10,11,12,13,14,15} over 5 places,
MN rule set = { M} over 6 places,
MNP rule set = {M} over 7 places.

Rule schemes for morphic CA

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

ruleM[{a_, b_, c_, d_, e_}] :=
    Flatten[{rca[{a}], rca[{b}], rca[{c}], rca[{d}], rca[{e}]}]

ruleDM[{a_, b_, c_, d_, e_}] :=
    Flatten[{dca[{a}], dca[{b}], dca[{c}], dca[{d}], dca[{e}]}]

ruleMN[{a_, b_, c_, d_, e_, f_}] :=
    Flatten[{rca[{a}], rca[{b}], rca[{c}], rca[{d}], rca[{e}], rca[{f}]}]

ruleMNP[{a_, b_, c_, d_, e_, f_, g_}] :=
    Flatten[{rca[{a}], rca[{b}], rca[{c}], rca[{d}], rca[{e}], rca[{f}], rca[{g}]}]

ruleM42[{a_, b_, c_, d_, e_, f_, g_, h_}] :=
    Flatten[{rlc[{a}], rlc[{b}], rlc[{c}], rlc[{d}],
             rlc[{e}], rlc[{f}], rlc[{g}], rlc[{h}]}]
```

Motivations

After Niklas Luhmann got mesmerized by the performances of the magician Heinz von Foerster (Second Order Cybernetics), the *Laws of Form* nevertheles nurtured a whole movement over decades of mainly German theoretical sociologists.

Finally, the *morphogrammatic* endeavor is more or less unknown to the academic public. It goes back to Gotthard Gunther's work in the 1960s at the Biological Computer Lab, Urbana, Ill. on a reflectional theory of living systems capable of self-reflection and autnomous decision making.

It is certainly not my aim to enter into a 'debate' about 'deviant' or 'cranky' (Hao Wang) formal languages.

There is also no need to go back to the Ancient Chinese thinkers, Pythagoras or Kepler to connect conceptions with numbers and sound.

It suffice to reduce the apparatus to an elementary construct that has shown to be quite powerful.

Cellular automata have a long history but might have come to the public awareness only by computer-supported experiences.

The challenge of this exercise is to hear and see concepts, i.e. *structurations*, and not just to perceive sounds and pictures.

This seems to be possible mainly by a *comparison* of the (deep)structure of different sound and graphic systems based on specific cellular automata.

Epistemologically different perceptive systems are not properly defined by their internal differences but by the difference between their deep-structural differentiations.

A nice, and well developed structuration is established with the difference of *classical*, *indicational* and *morphogrammatic* writing systems as proposed in my texts towards a general theory of writing systems, called graphematics.

Also sounds of different graphematical systems are heard and pictures are seen, there is nevertheless a crucial difference in their *mode* of production.

The slogan, *What you see is what you get*, applies properly for *classical* approaches of a visualization of concepts and a way to present sound.

The *indicational* approach is still in the framework of the classical semiotic conceptualizations and modes of writing. Because the fundamental *'topological invariance'* of its mode of writing which is not directly percievable it demands for some combinatorial manipulations of the notation to be possible to perceive in a classical mode of perception its genuine abstraction properly.

The *morphogrammatic* approach is denying the obvious WYSIWIYG concept. Therefore, *What you see and hear is not what it is*. Morphogrammatics is a part of kenogrammatics. The term *'keno'* in kenogrammatics derived from the Greek *kenoma (κένωμα*, emptiness) says it all: there is nothing to tell.

The WYSHINWIS, *"What You See and Hear Is Not What It iS"*, challenges towards a new kind of perception: the cognitive perception, that is at once, a perceptive cognition.

The set of paradigms, the *classical*, the *indicational* and the *morphogrammatic*, is certainly not exclusive. There are other kinds of conceptualizations and writing systems to recognize too. This is explicitly developed in the research field of *graphematics*, the morphosphere(s).

Hence, the aim of this little *exercise* presented here, is to learn and to train the brain to recognize the difference between different conceptual systems as they are presented to our perception by sounds and graphics and not their intrinsic beauty or lack of it.

Neither are there any mathematical or philosophical considerations included in this proposal of an exercise.

One exercise might be concentrated mainly on the single form of *Sierpiński triangles*, and its variations in graphic and sonic environments. It then could be seen as a kind of a contemplation on Wolfram's ECA rule 90. In this case such exercises are restricted to strictly symmetric patterns of homogeneous or heterogeneous dynamics.

Technically I apply freely David Burraston's *"Music Box Toy with Elementary Cellular Automata"*.

What do we have to do to follow this invitation?

Compare, contemplate and analyze your experiences!

The rules of the presented CA are new and had been introduced by the author as an experimentation to deal with morphogrammatic and indicational CA.

This work of different paradigms of formal systems goes back to publications that did not yet use *Mathematica* as a programming environent.

David Burraston's program *"Music Box Toy with Elementary Cellular Automata"*.

"This Demonstration uses a simple and novel mapping of elementary cellular automata (CA) to single - voice musical sequences. The mapping is created by evolving a small CA through all its possible initial conditions for a number of generations, converting the cells to decimals and storing them in a table. This table is visualized using Mathematica' s built - in function ArrayPlot *with starting conditions assigned vertically and generations evolving horizontally."* (David Burraston)

More at: http://noyzelab.blogspot.com.au

David Burraston, *"Music Box Toy with Elementary Cellular Automata"*
http : // demonstrations.wolfram.com/MusicBoxToyWithElementaryCellularAutomata/Wolfram Demonstrations Project
Published : January 31, 2012

How to use the programs?

How to compare sound and visual events?

[Unfortunately, only with the formats CDF and HTML their is chance to enjoy the very fun of the programs.]

Again, my intention is not to simulate known kinds of anthropologically founded music and sounds but to let the algorithms, as far as possible, 'manifest' themselves thru the machine.

Hence, my not so humble intention seems to be to enable a project that evokes *"a kind of a new kind of music of no kind"*.

It might be a late contribution to *Bruce Stirling's* remark at the Academy of Media Arts, Cologne around 2000, that he preferred very much more to see/hear the machine manifesting itself instead of imitating, simulating and varying human achievements in music and poetry.

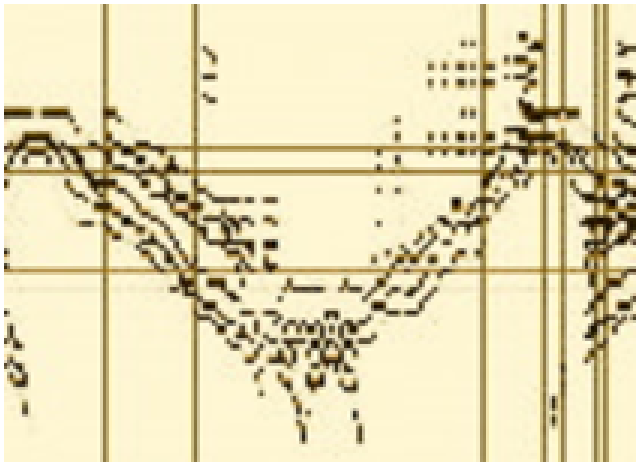
A different intention is expressed by:

Stephen Wolfram: *"How difficult is it to generate human-like music? To pass the analog of the Turing test for music?"*

http://blog.stephenwolfram.com/2011/06/music-mathematica-and-the-computational-universe/

Katarina Miljkovic's mixed approach

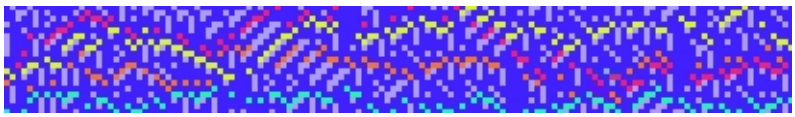
"Generative sound is always coupled with live performance of some kind, often improvisation, to contrast and enrich the mechanical aspects of electronic part."



http: // www.katarina – miljkovic.net /

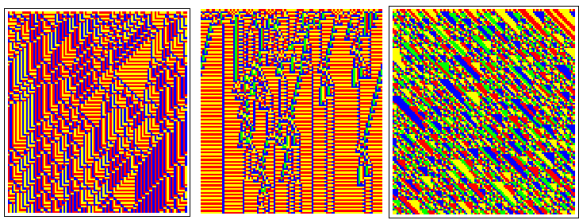
More CA sounds: Music simulations versus algorithmic sounds

Music by John Baez and Wolframtones, June 2, 2009



`http : // math.ucr.edu / home / baez / music / treq_lila / 4 _lalla _aisha _ii.mp3`

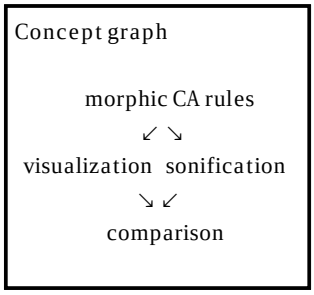
MorphicAlgorithms, Rudolf Kaehr, Glasgow 2014



MorphicAlgorithms are part of the project *"The Sound Box: A Kind of a New Music & Visualization Box"*
The aim is not to simulate music by algorithms, but to give a *sonic* impression of the structure of the morpho - cellular automata. The' pure' structure of the automaton might be distracted by the instrumental interpretation of its algorithms.
The Sound Box offers more than 500 different morpho - cellular automata and its sonifications. All the algorithms have a visualization too. A visualization of the sounds is added as the label of the track. The first specification of the track is the name of the algorithm, the second, the number of the' instrument'.

https://soundcloud.com/morphicalgorithms

Orientation thru different paradigms and categories



The sound and visualization programs are both based on the same morphogrammatic production rules for 1D cellular automata.

Therefore, the comparison of both types of events is strictly one-to-one.
That is, the category *ruleM* of the sound box corresponds directly to the category *ruleM* of the visualization box. And this holds also for all other categories.

The graphematic rules are classified in 3 categories:
1) *morphogrammatic:*
a) static rules: *ruleCl*, *ruleMN*, *ruleMNP*, *ruleM42* and
b) dynamic rules: *ruleDM*,
both rule sets have a *simple* or a *random* seed (init),
2) the *indicational* rules: *ruleCIR*, *ruleCIRT*, and
3) the *classical* CA rules: *CA rules*.

Additionally, there are some selected rules too: *special M* and *special CA*.

The offered direct graphic representations of the sound production, as shown by the sound box, is more complex than the underlying strict production rules for the cellular automata. Its complexity is defined by the complexity of *"all its possible initial conditions for a number of generations"* (Burraston).

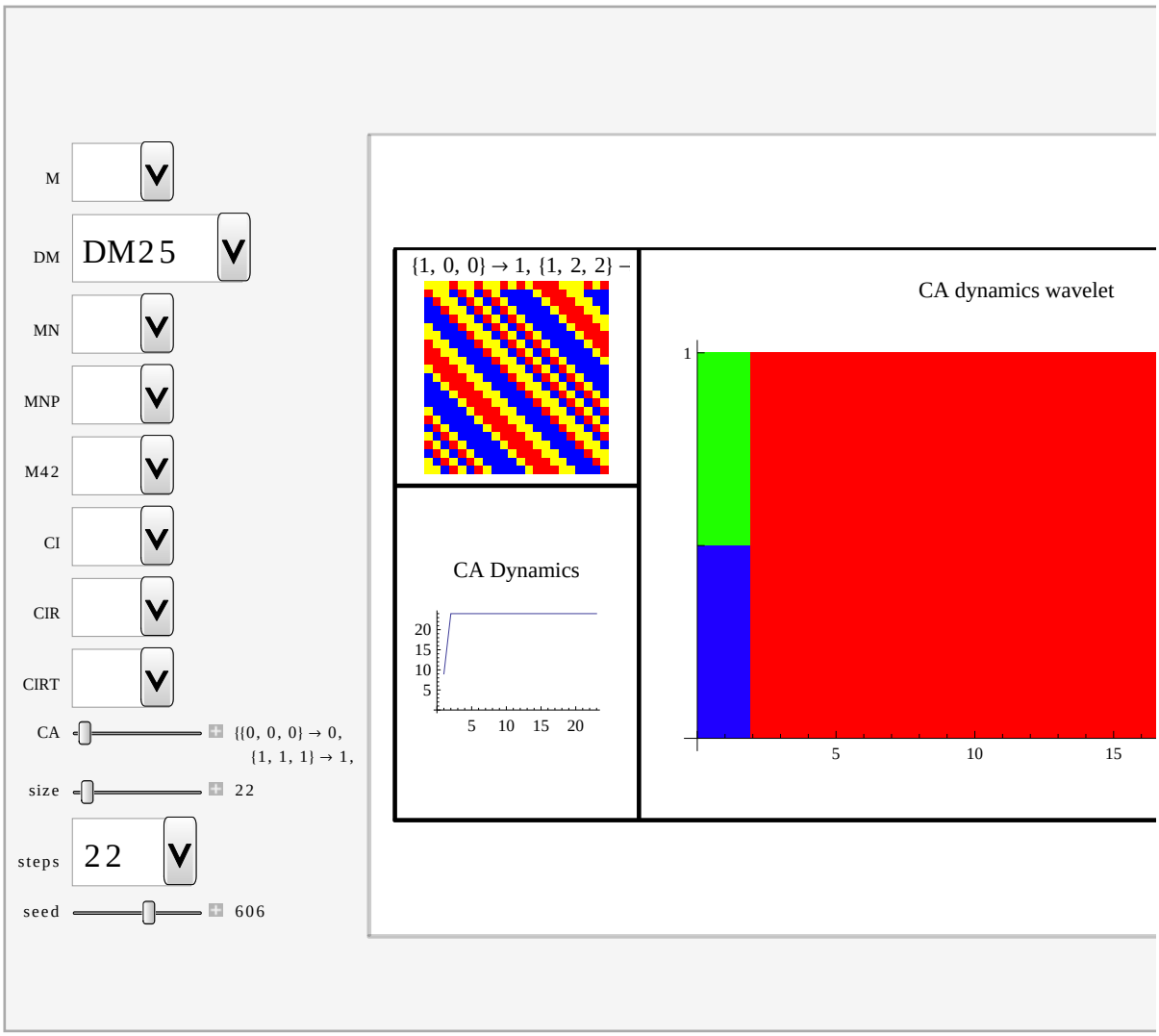
Therefore, a separate visualization of the direct production rules with a simple or a random initial condition, seed, is offered too.

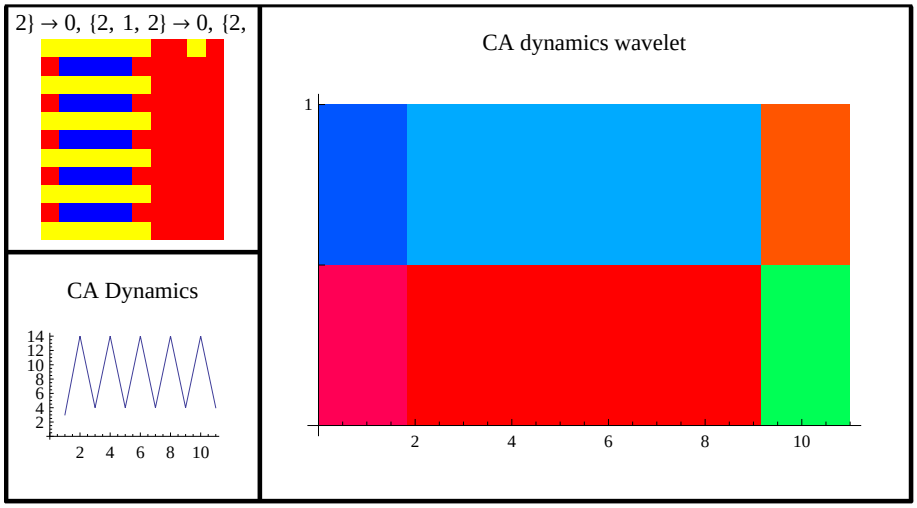
Comparisons

Comparison is part of the general project of *Comparatistics* that is comparing on a paradigmatical level different kinds of writing systems (symbolizations).

The topic *"comparison"* gets algorithmic support by several programs published by *Daniel de Souza Carvalho* at Wolfram Demonstrations Project.

A nice, slightly modified example for formal comparisons is given with Carvalho's *"Elementary Cellular Automaton Dynamics as Wavelets"*.





How to use the sound box?

A full description of the categories can be found in Burraston's paper.

• cells:

• generations:

- mapping:
- transpose:
- duration:
- instruments:
- show mesh:

How to use the graphics box?

- *n* (steps) glider for the number of 'generations'
- *steps* offers a set of fixed steps.

How to use the transition graph box?

- *width:* complexity of the environment
- *states:* set of the morphic states in symbolic form
- *icons:* set of the morpho rules in symbolic form

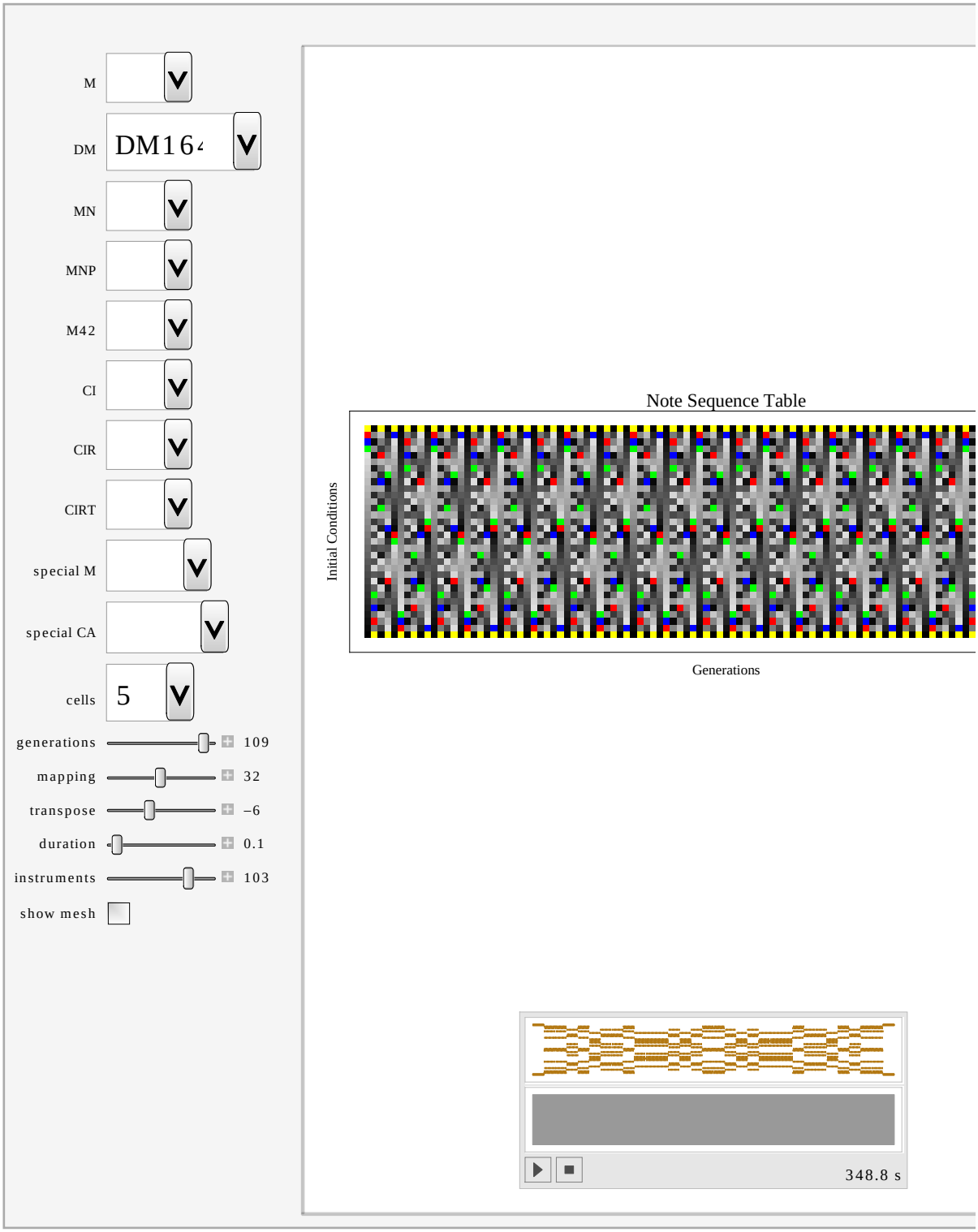
Initialization Code for PPSB-ruleSets

StaticMorphoRule Set

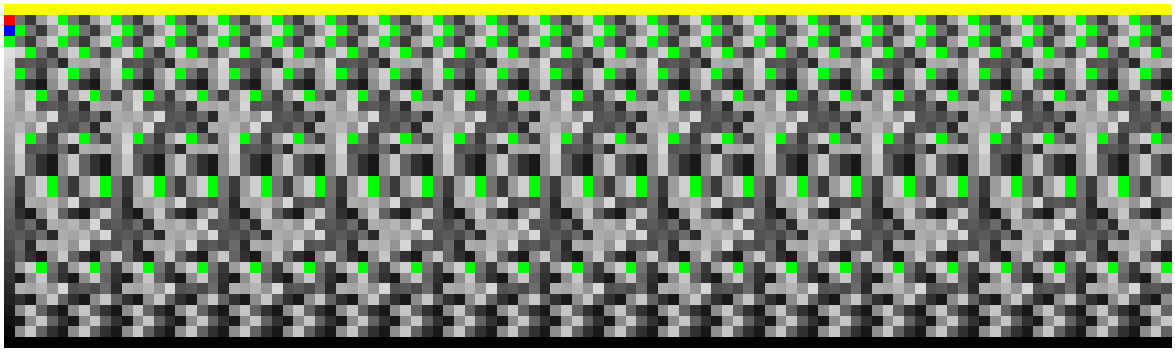
DynamicMorphoRule Set

IndicationalRule Set

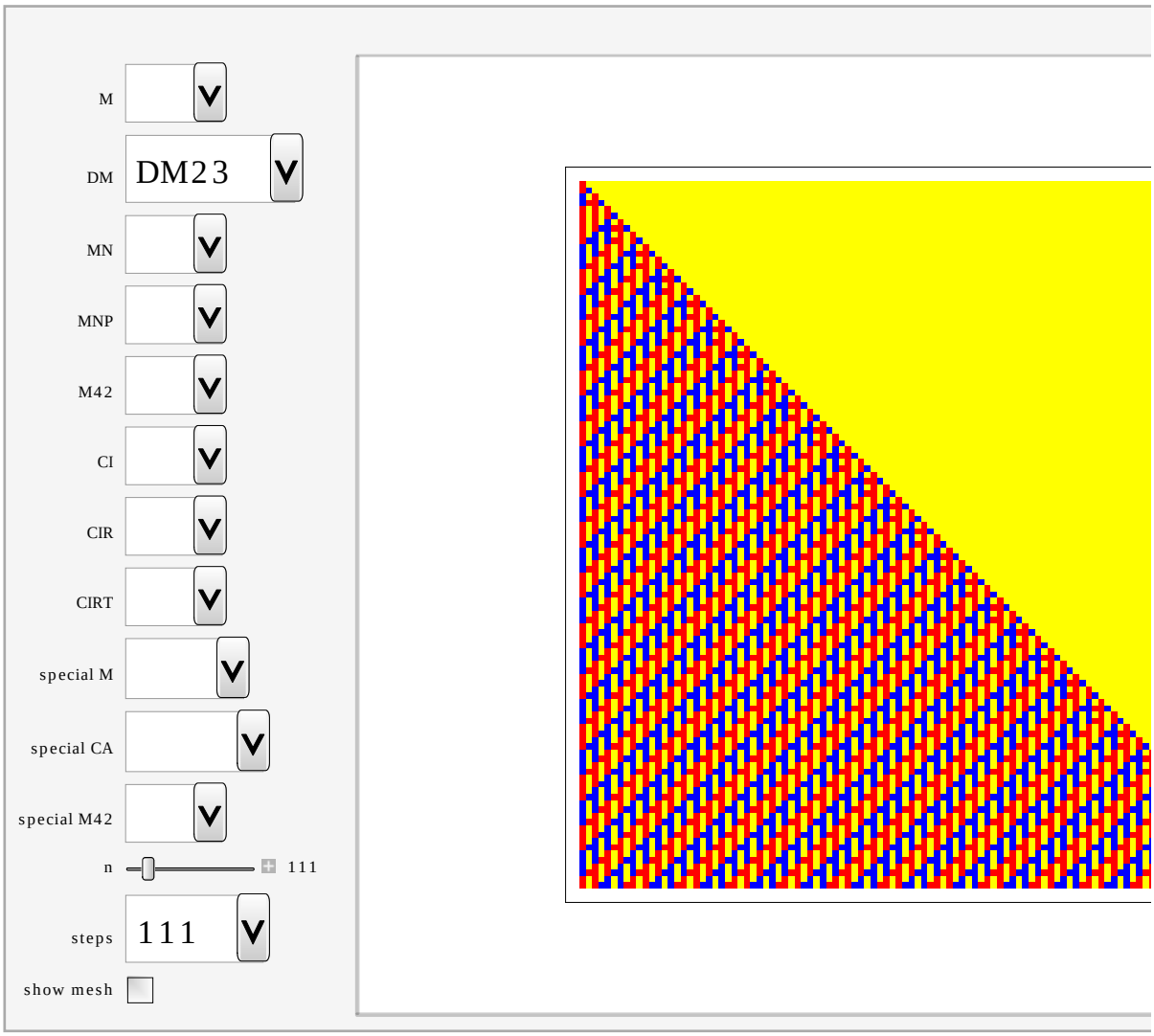
Dynamic Sonic Events with PPSB.x11



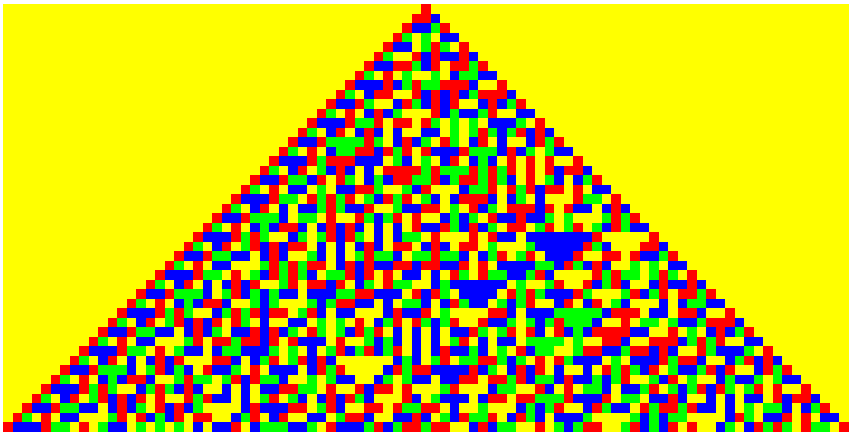
ruleDM 13, 5



Dynamic Visualizations of PPSB.x11 events

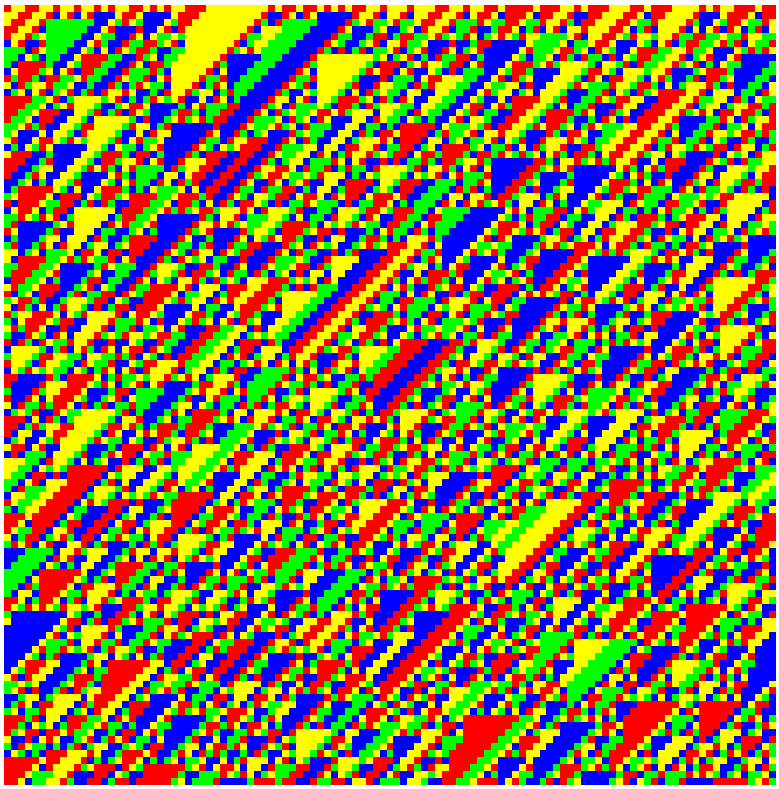


ruleDM 60, 44



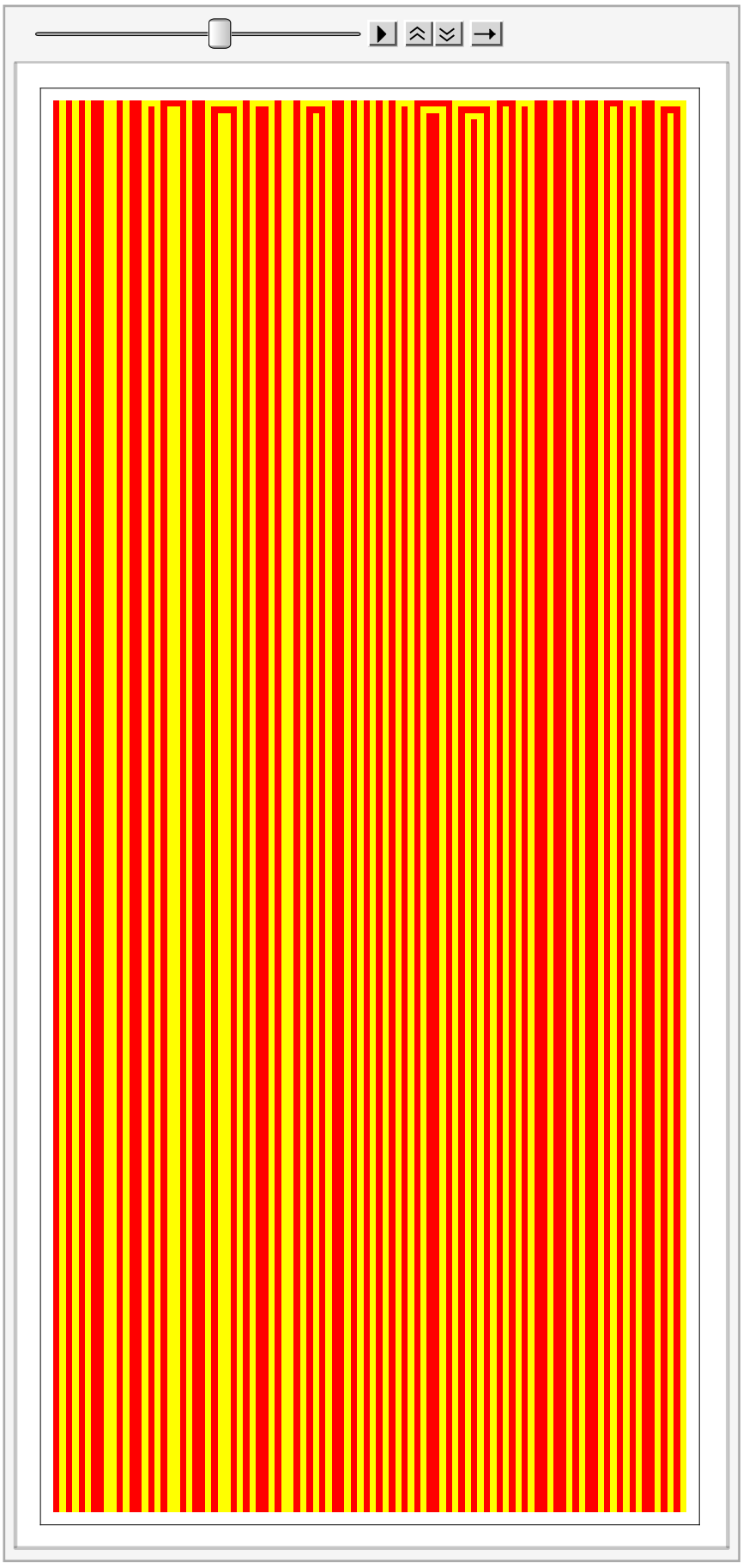
ruleM, static + ruleDM, dynamic, RandomInteger[1,111]

ruleDM 65, 111, Random

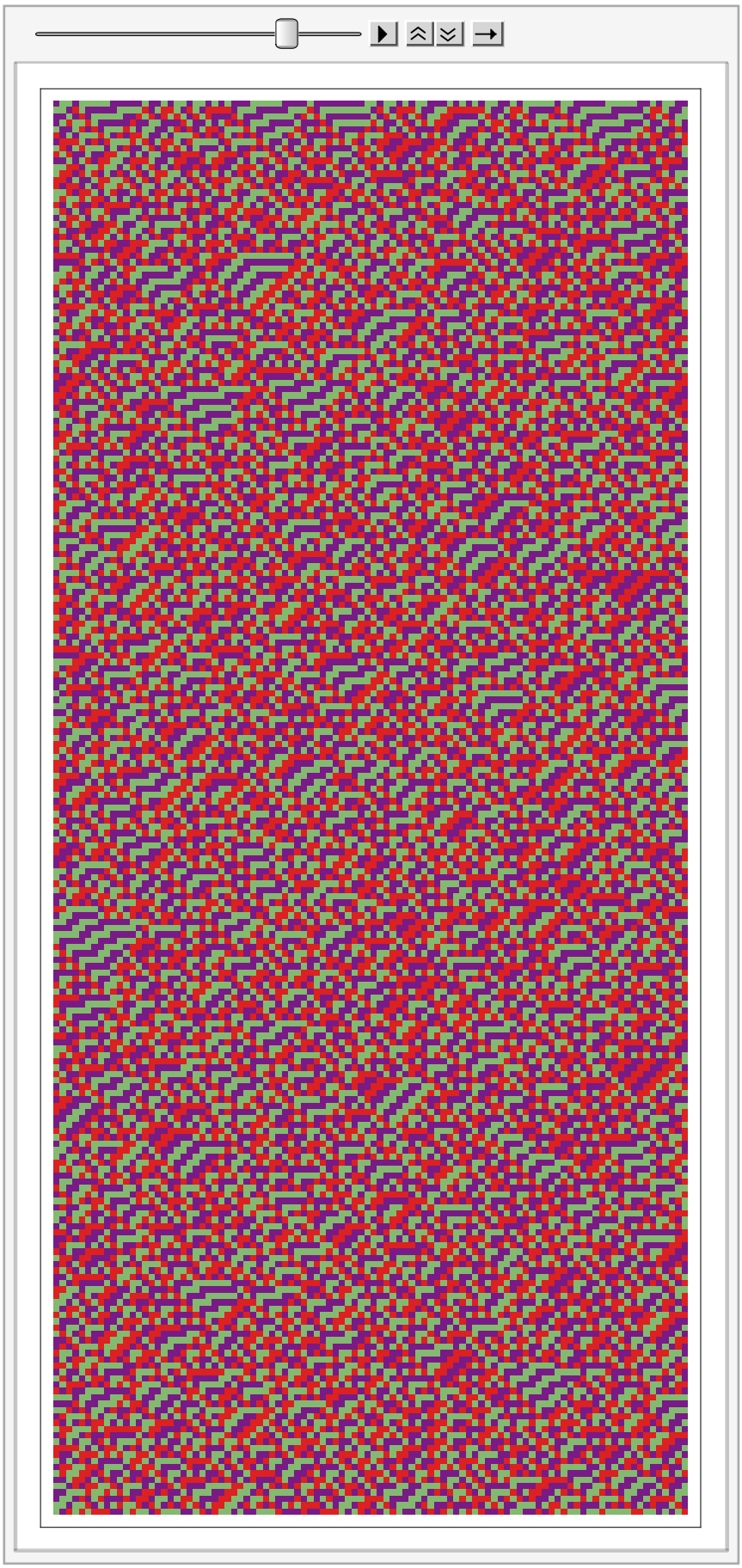


ListAnimate ruleDM, Random, 222

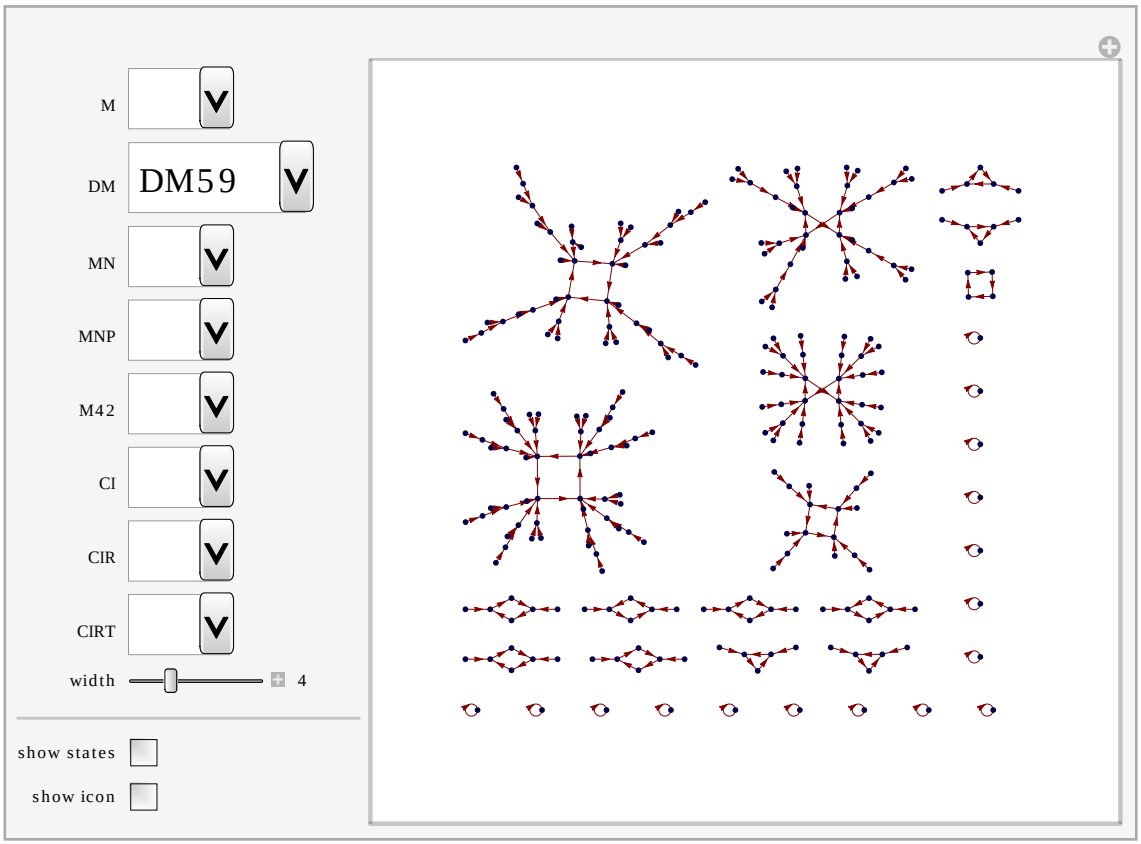
Discrete Colors



Rainbow Colors



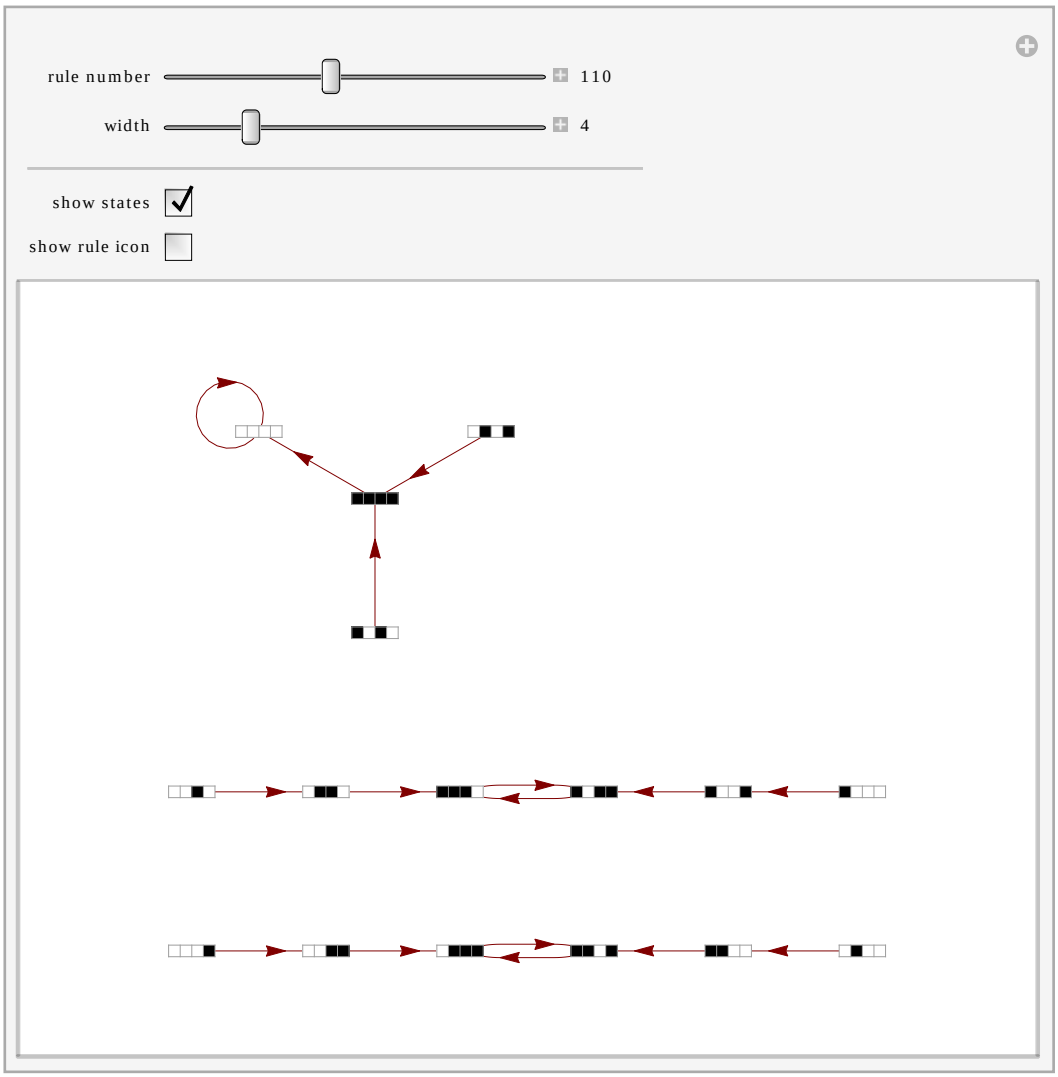
Transition Graph representation of PPSB.x11 events



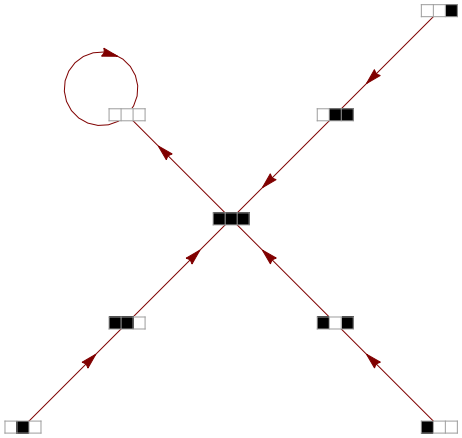
Transition graphs for classical CA

Stephen Wolfram, *"Cellular Automaton State Transition Diagrams"* from the Wolfram Demonstrations Project

http://demonstrations.wolfram.com/CellularAutomatonStateTransitionDiagrams/



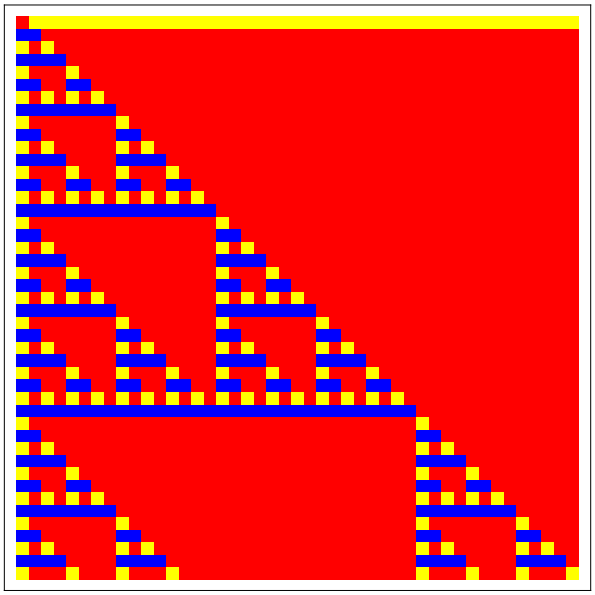
Transition graph of CA rule 110, width 3



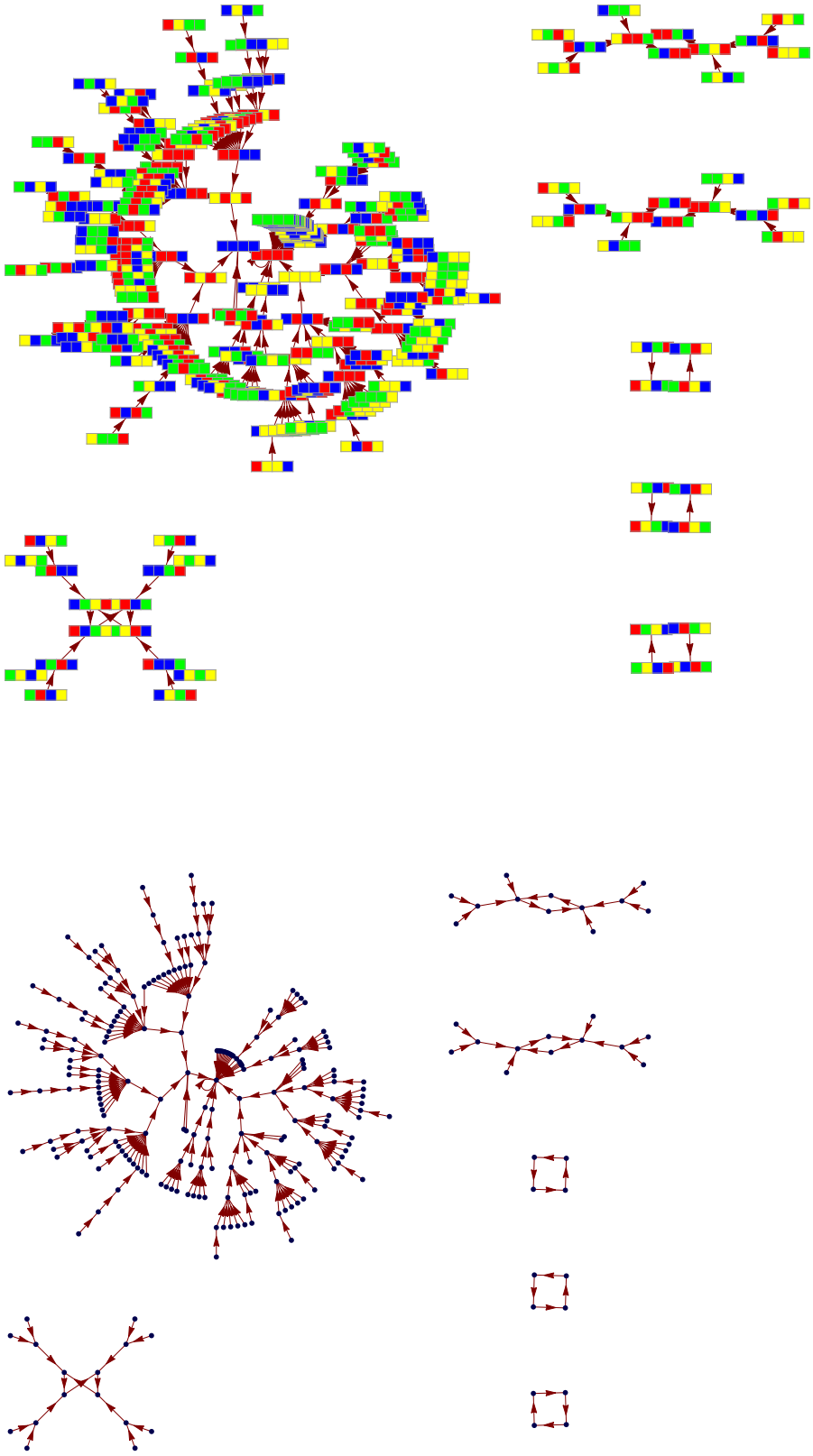
Comparatistics of picto-, diagrammatic and sonic manifestations of the ruleMN18

CA vizualitation of rule MN18, {9, 44, 47, 4, 0.1, 103}, witdth 4

```
ArrayPlot[CellularAutomaton[
  ruleMN[{1, 2, 12, 13, 5, 15}],
  {{1}, 0}, 44],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green}]
```



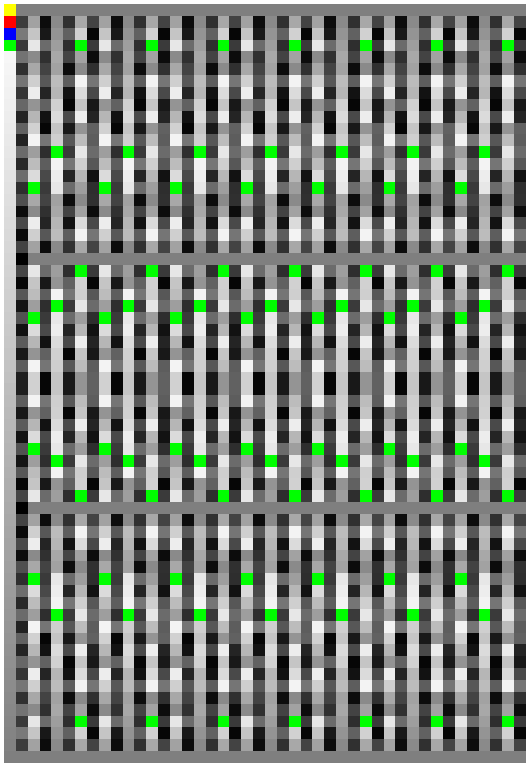
Transition graph for rule MN18, {9, 44, 47, 4, 0.1, 103}, witdth 4



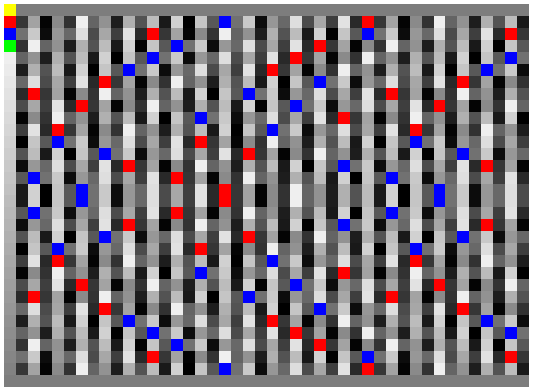
Sonic rule MN18, {9, 44, 47, 4, 0.1, 103}, 2252.8 s

Different number of cells

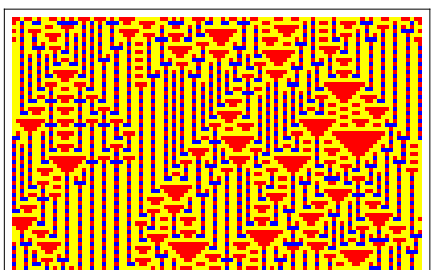
6 cells

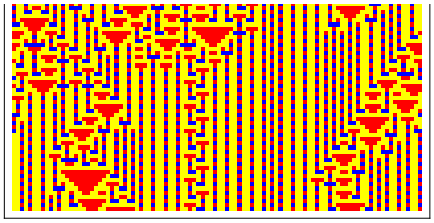


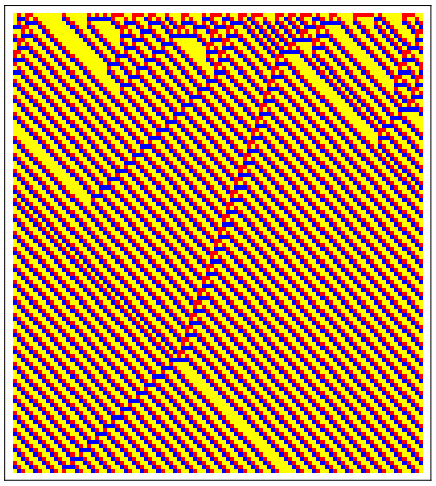
5 cells

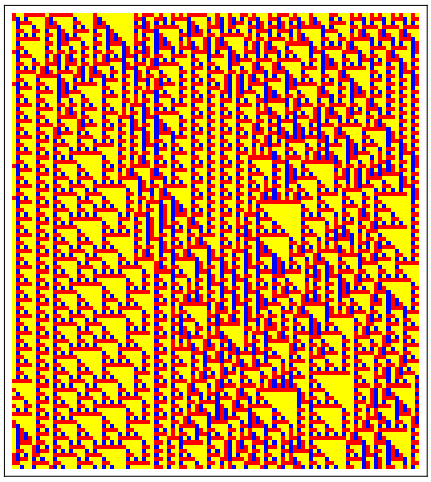


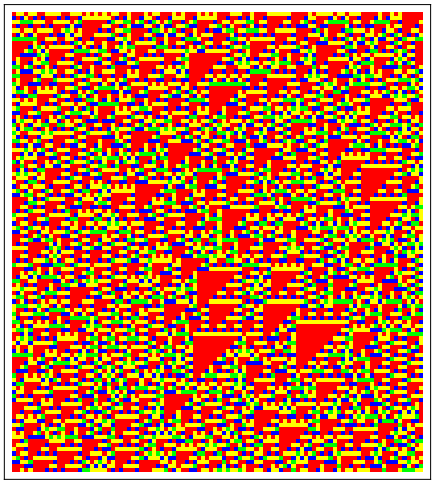
Examples of the plot table of vizualisations of the morphic *ruleM*, RandomInteger[1,100]

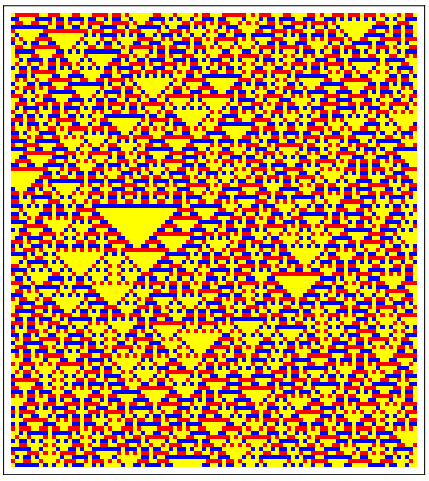




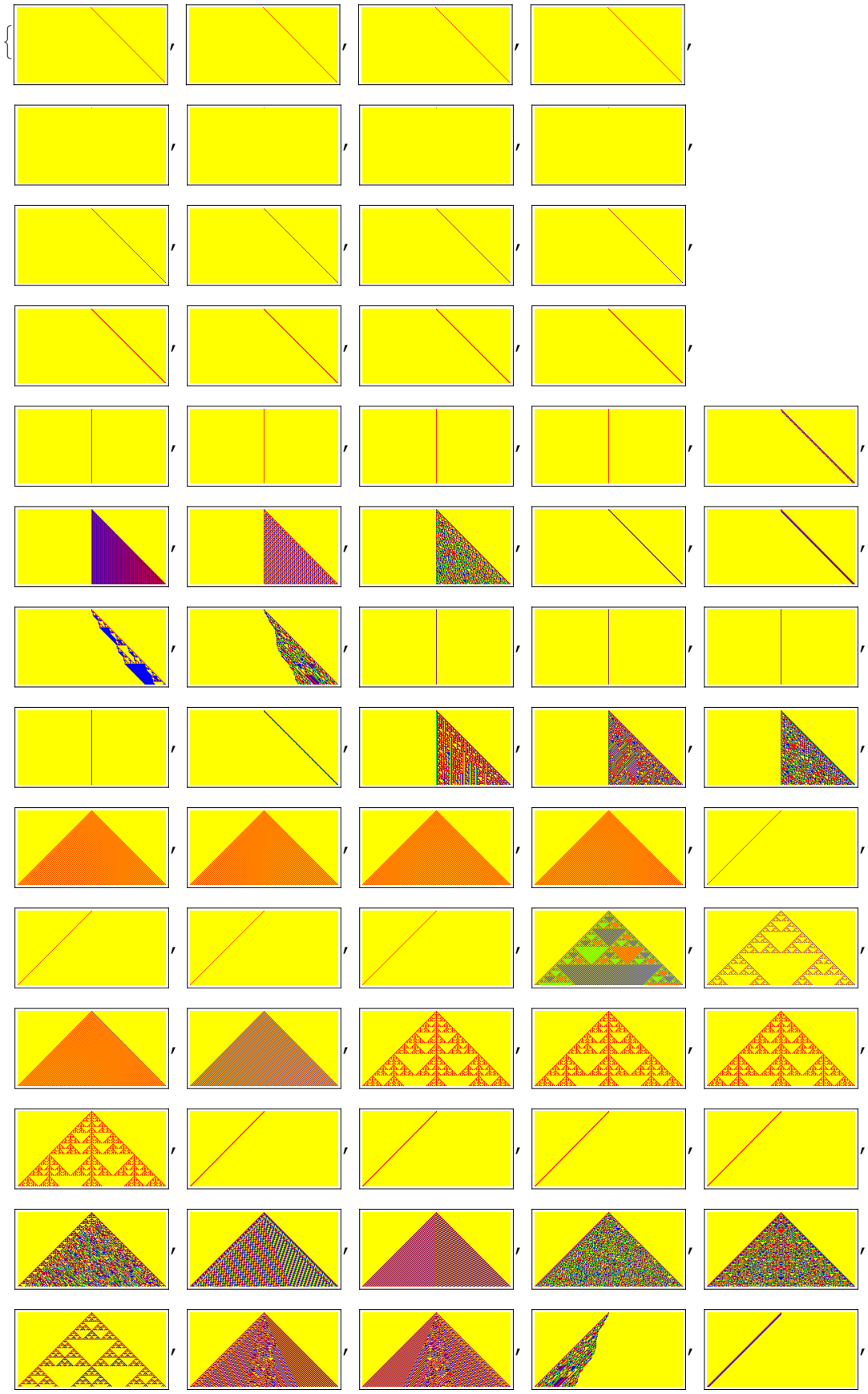


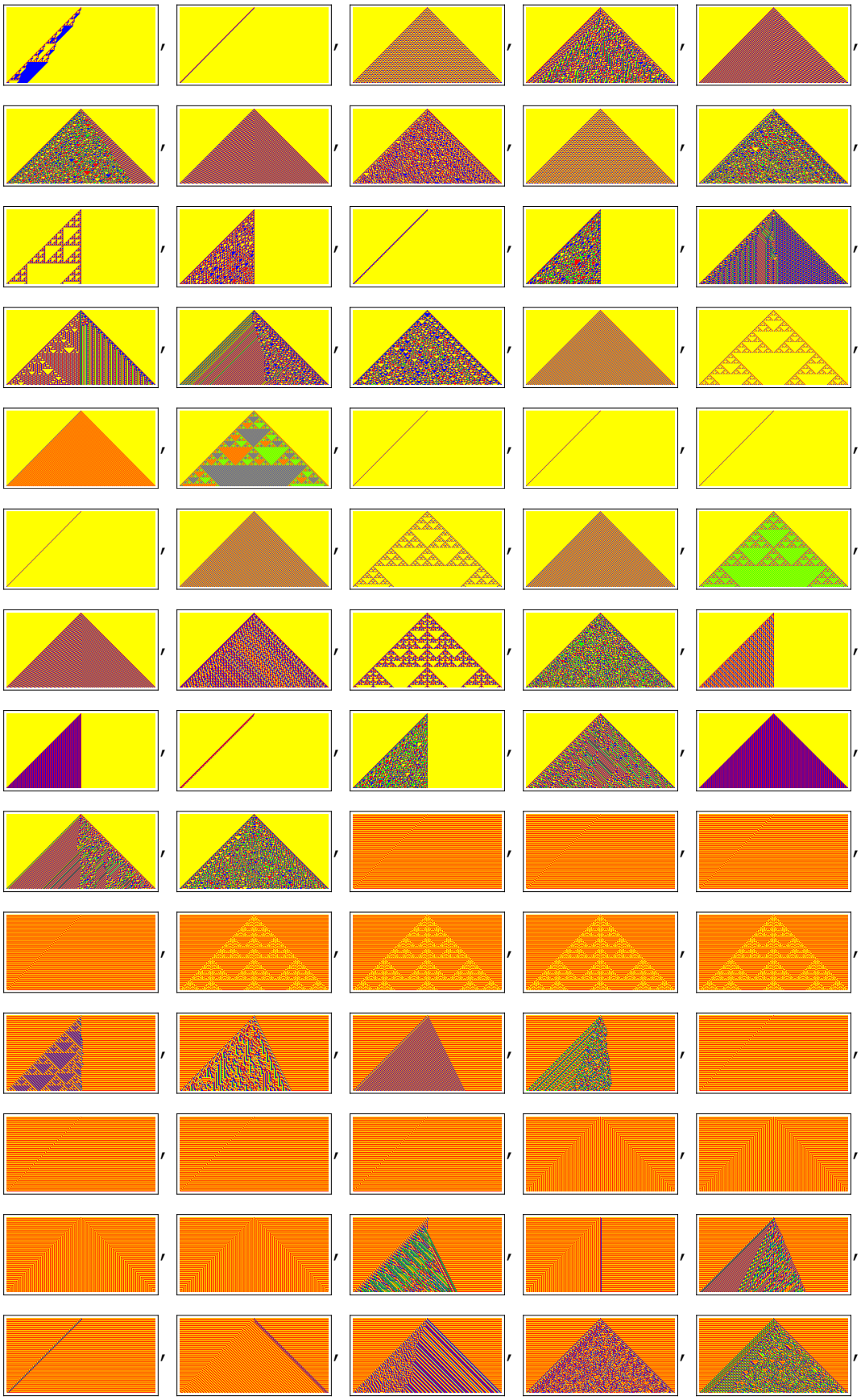


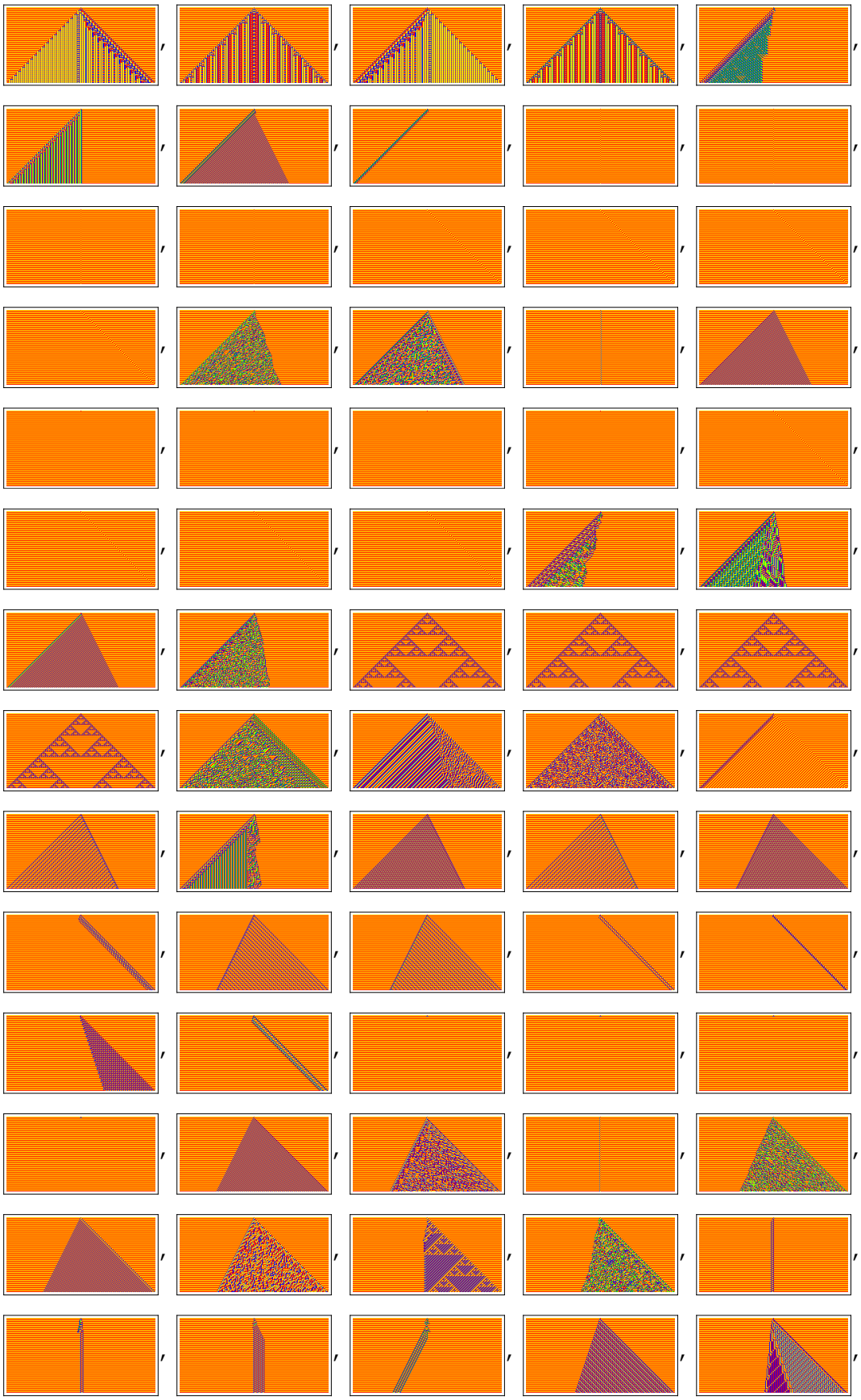


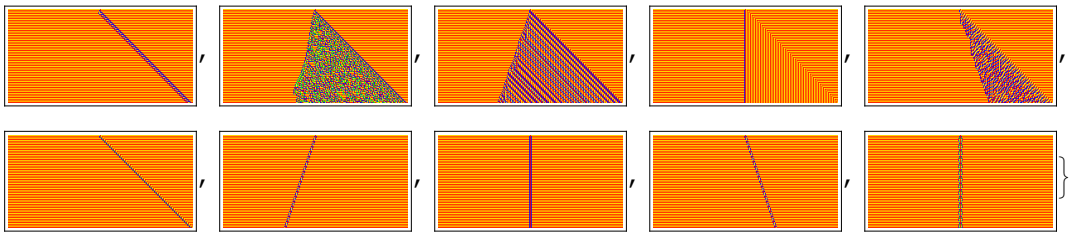


Complete plot table of vizualisations of the morphic rules *ruleDM*

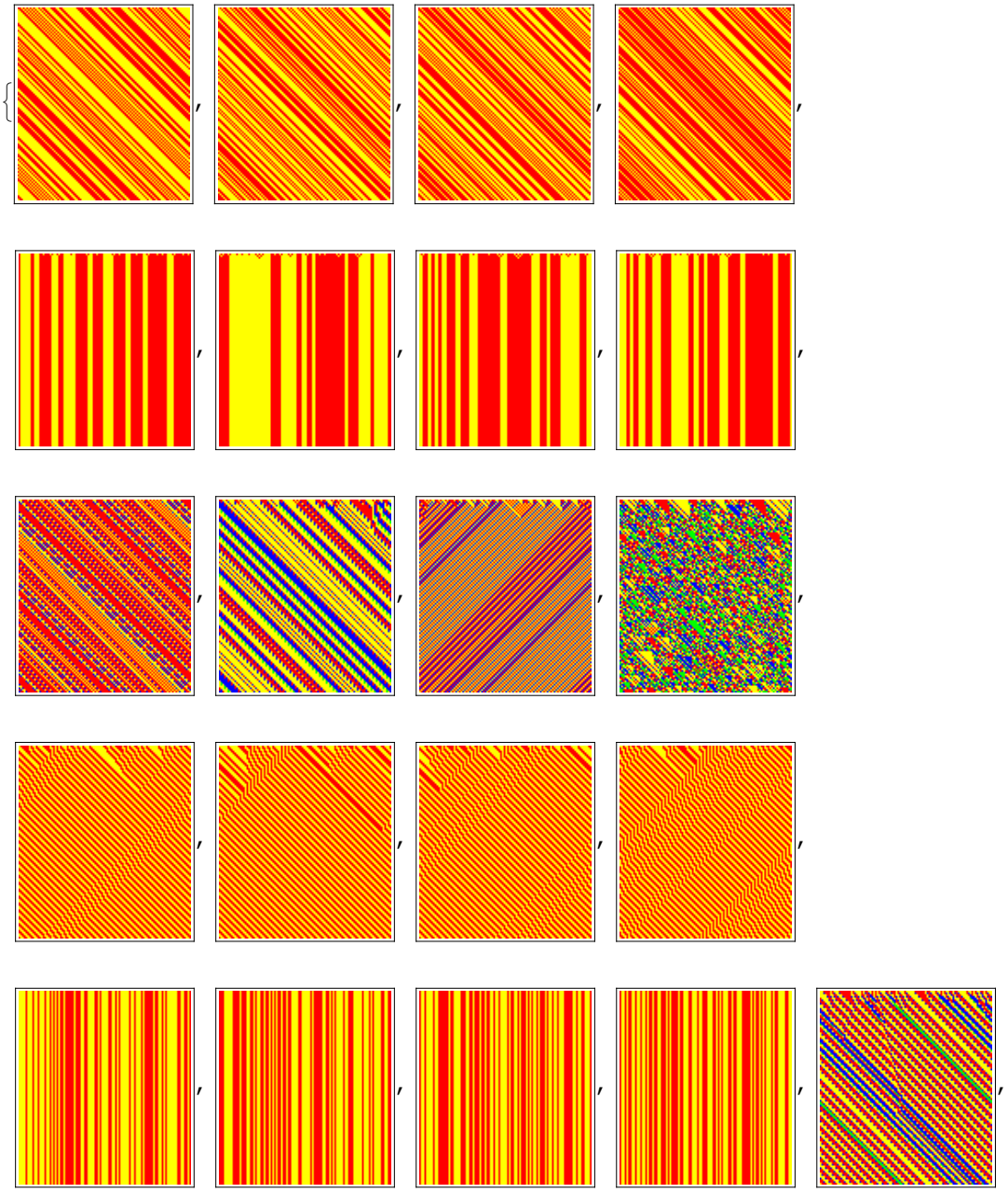


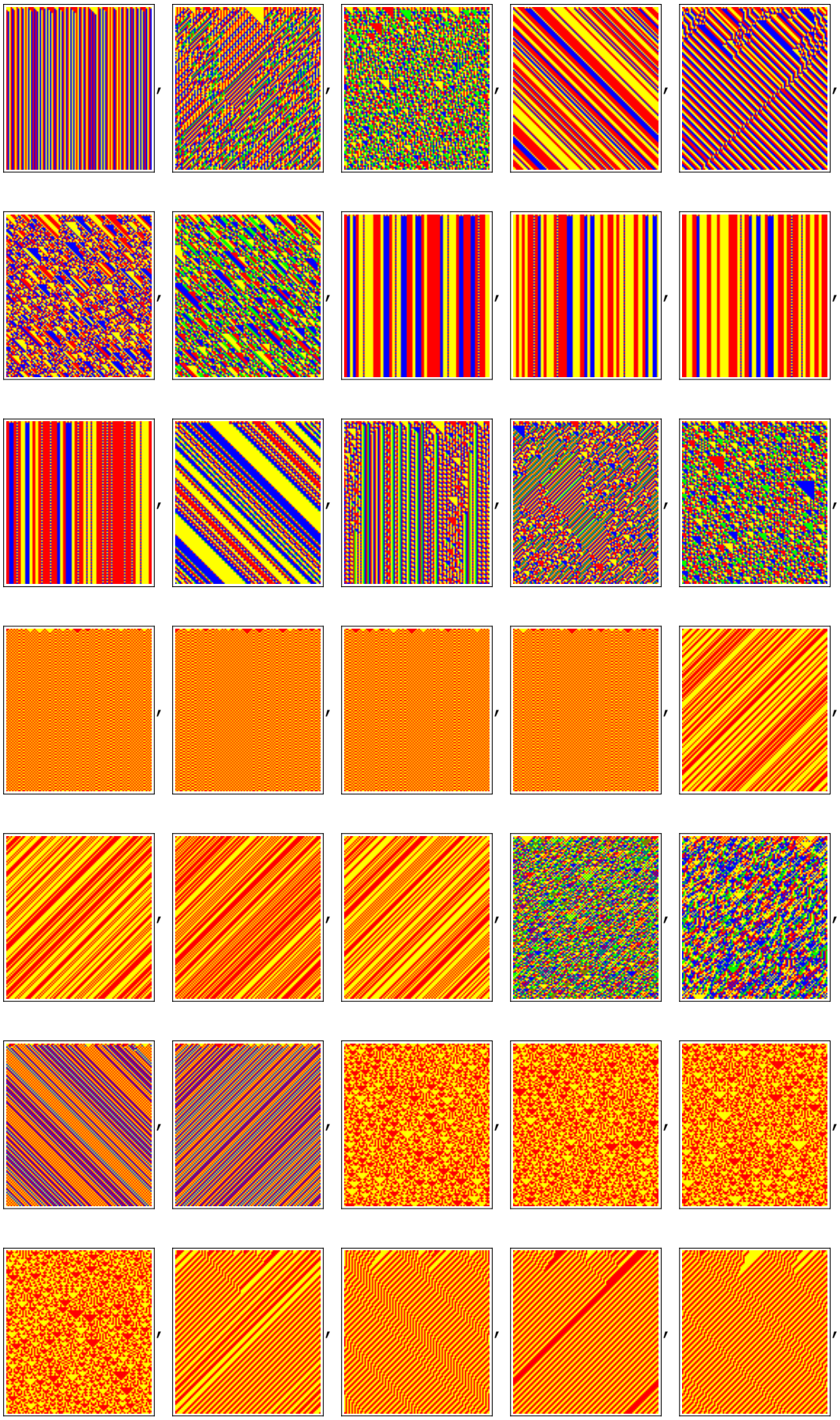


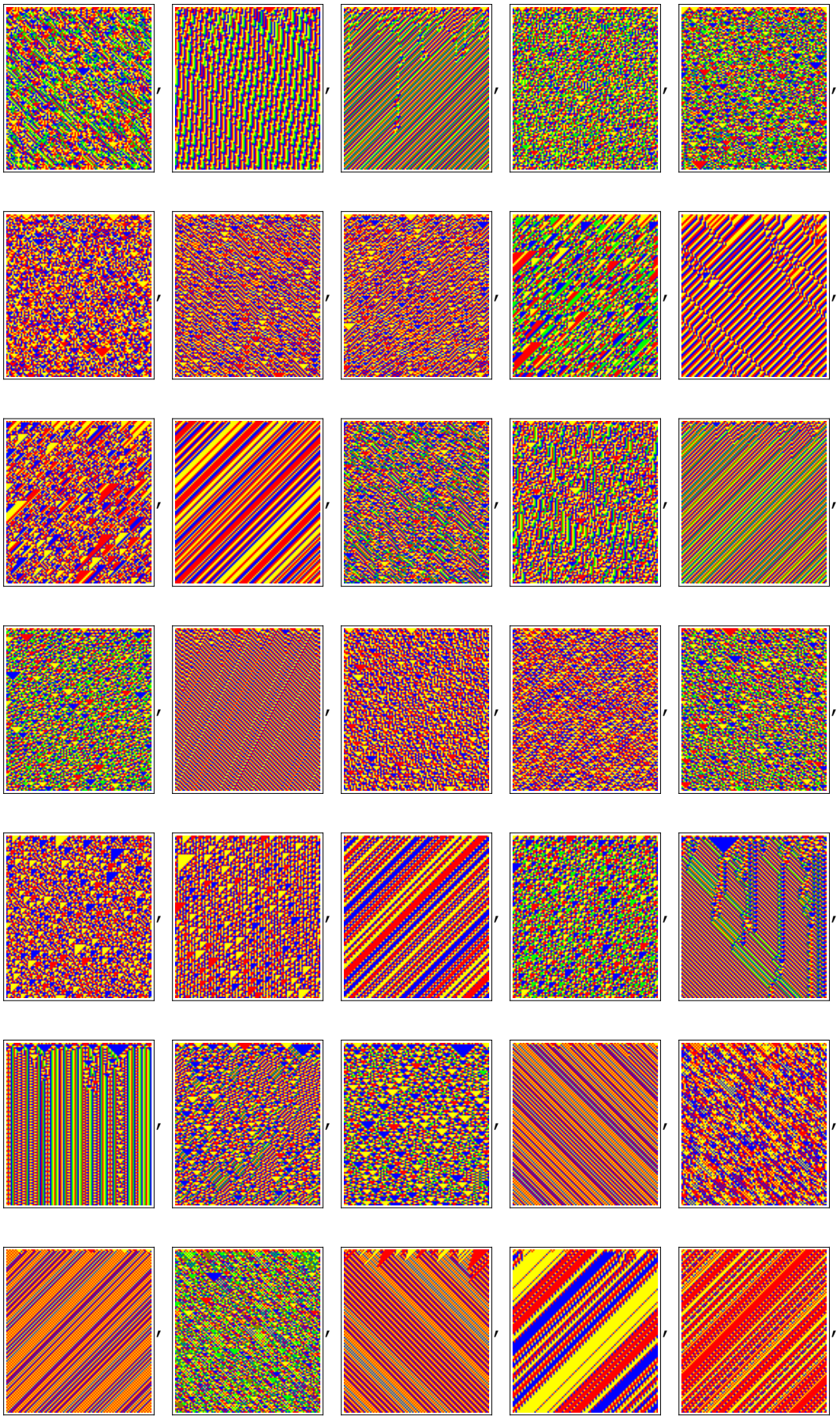


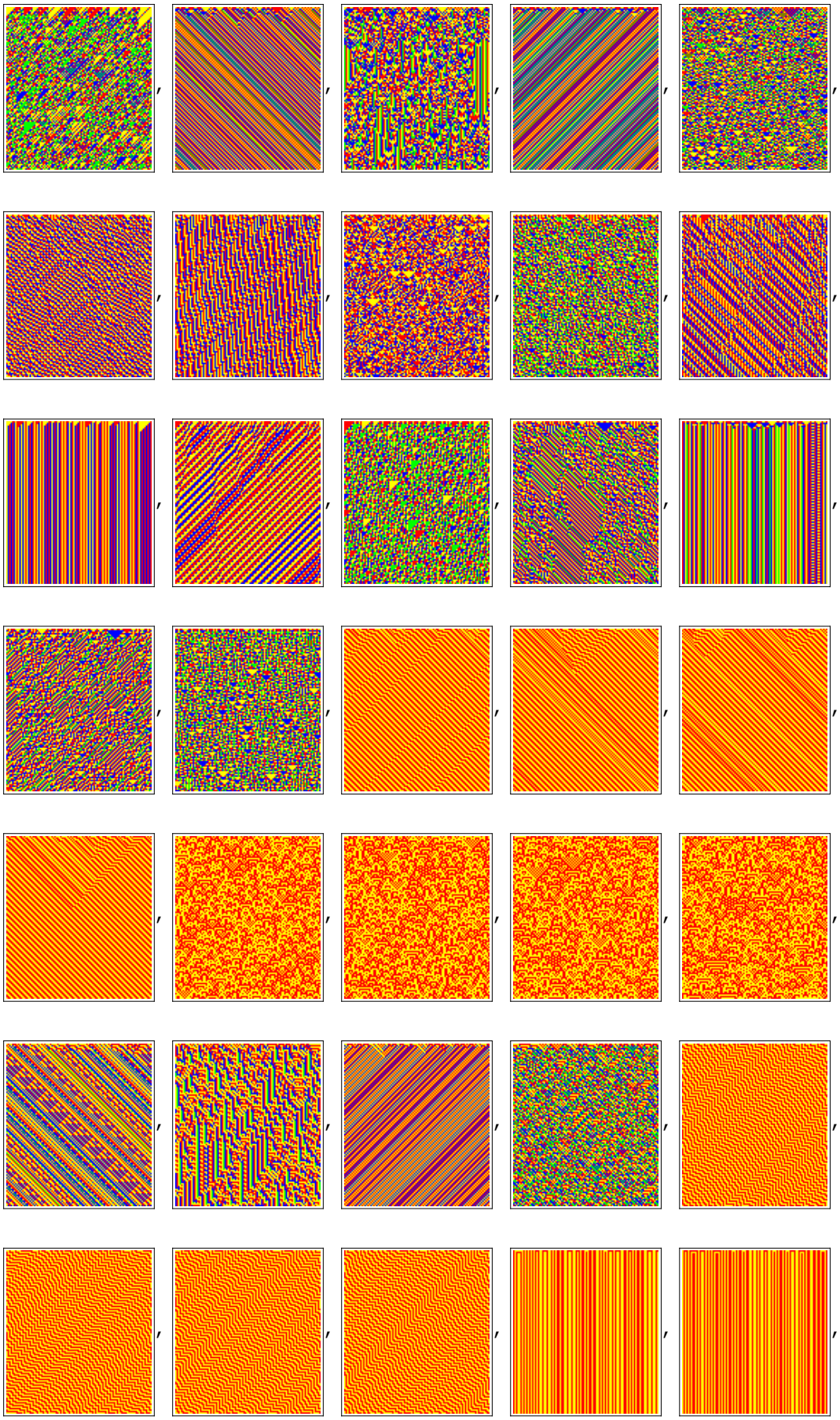


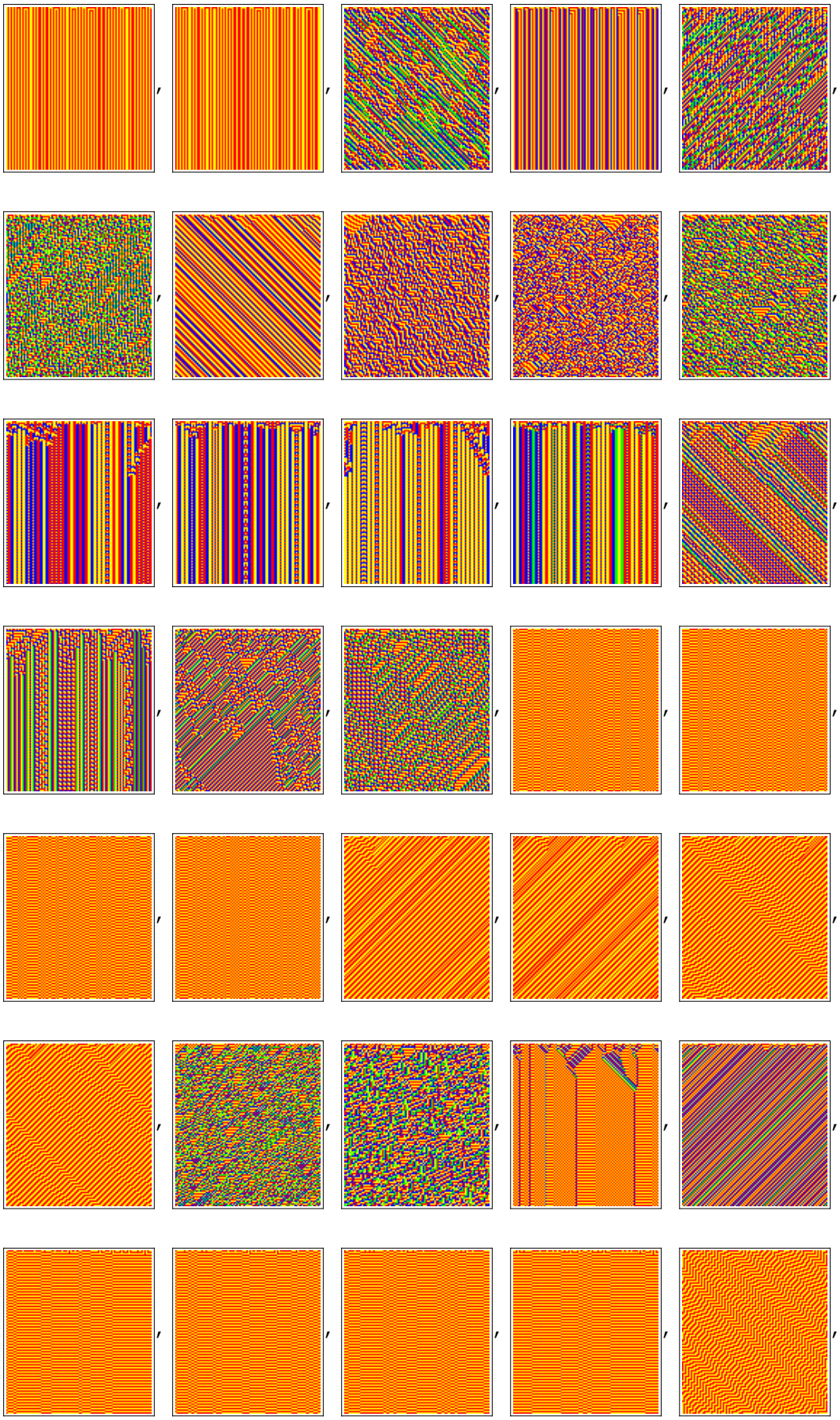
Complete plot table of vizualisations of the morphic rules *ruleDM*, RandomInteger[1,100]

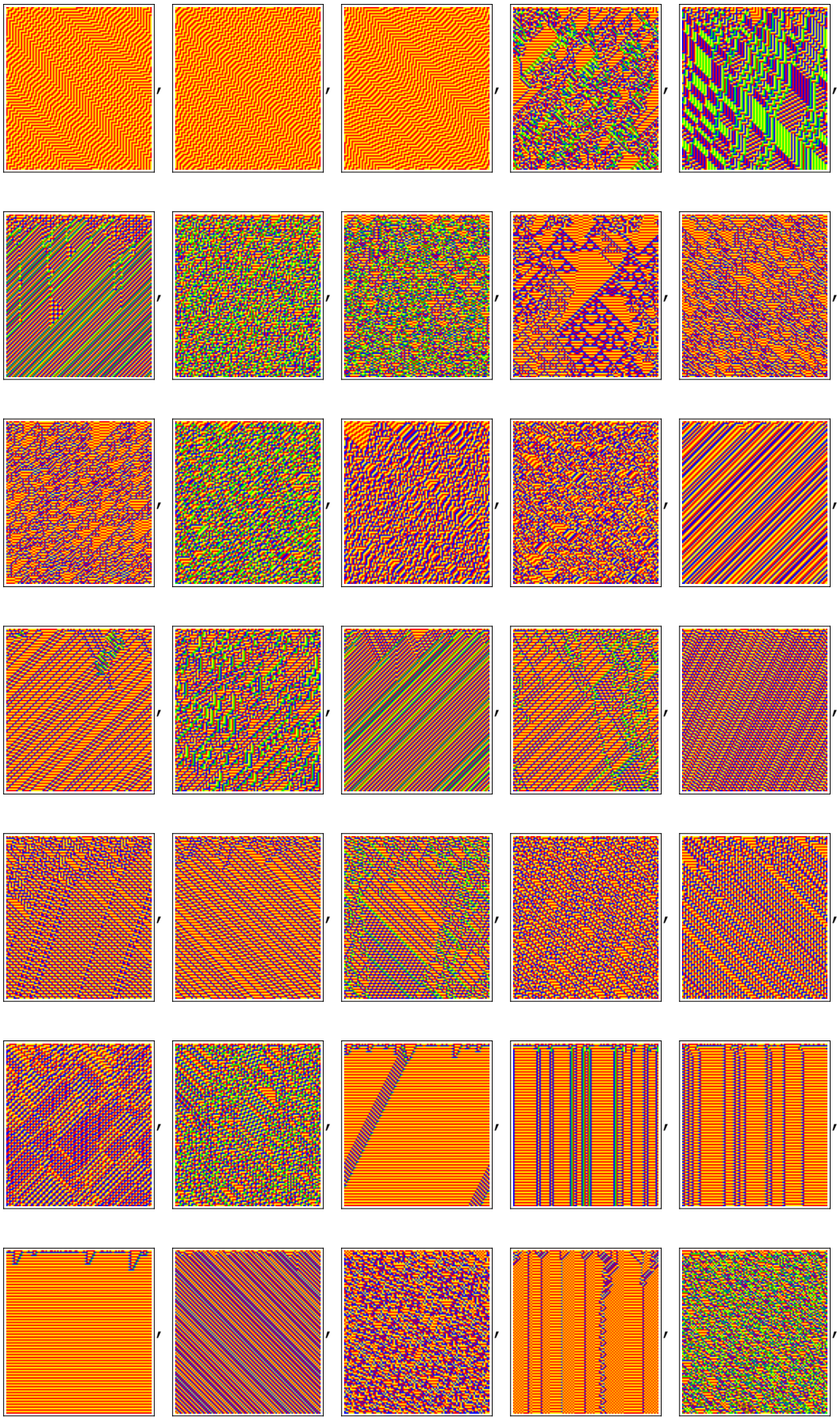


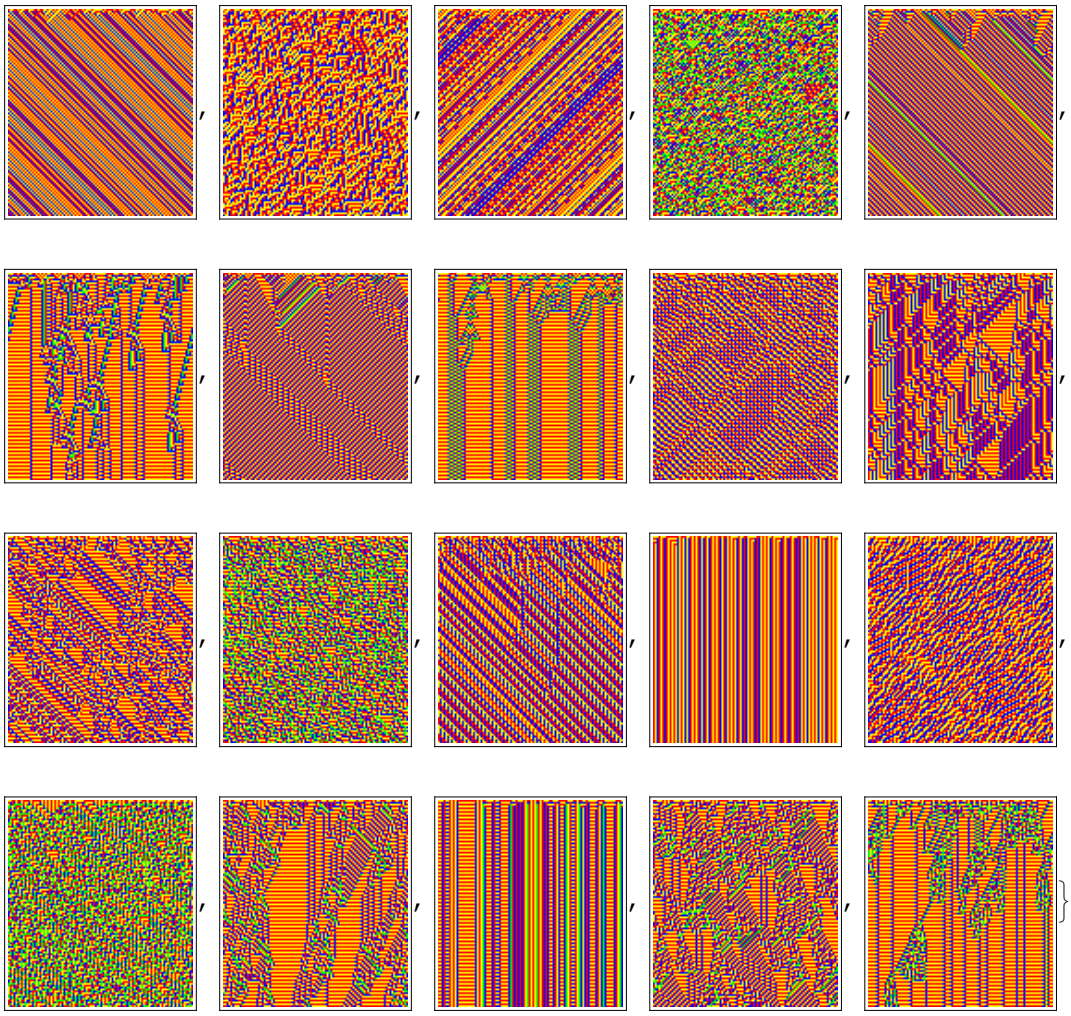




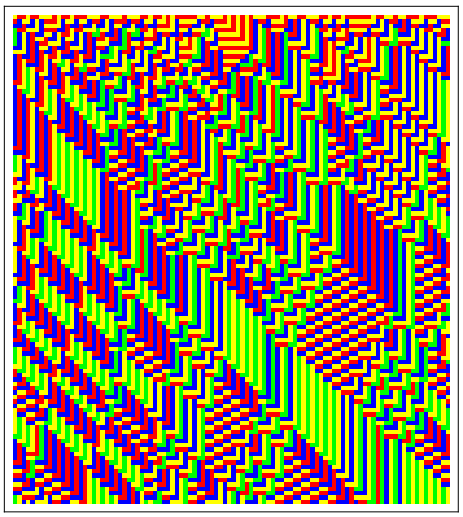


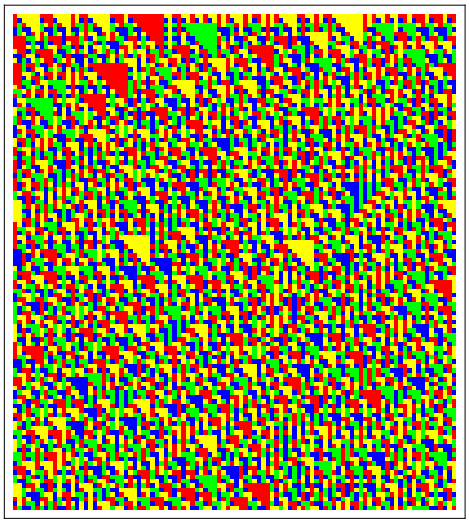


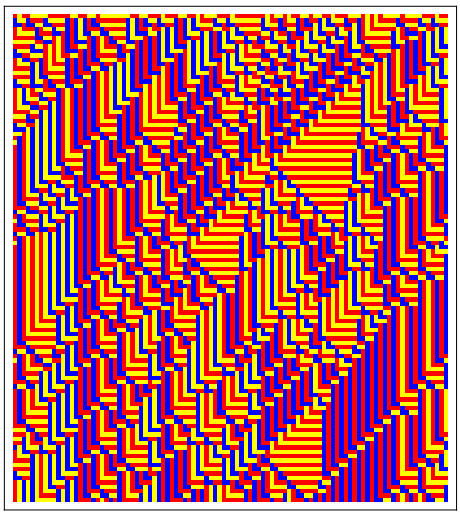


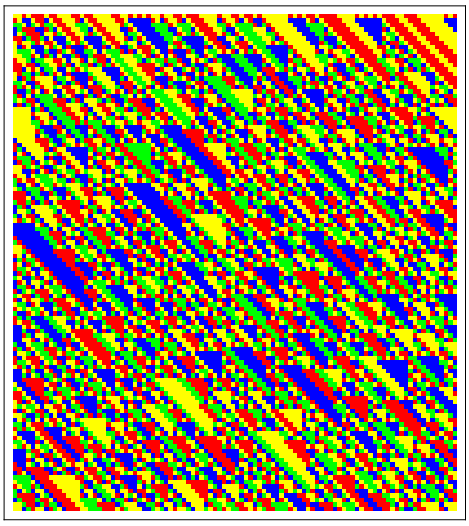


Special cases of ruleDM





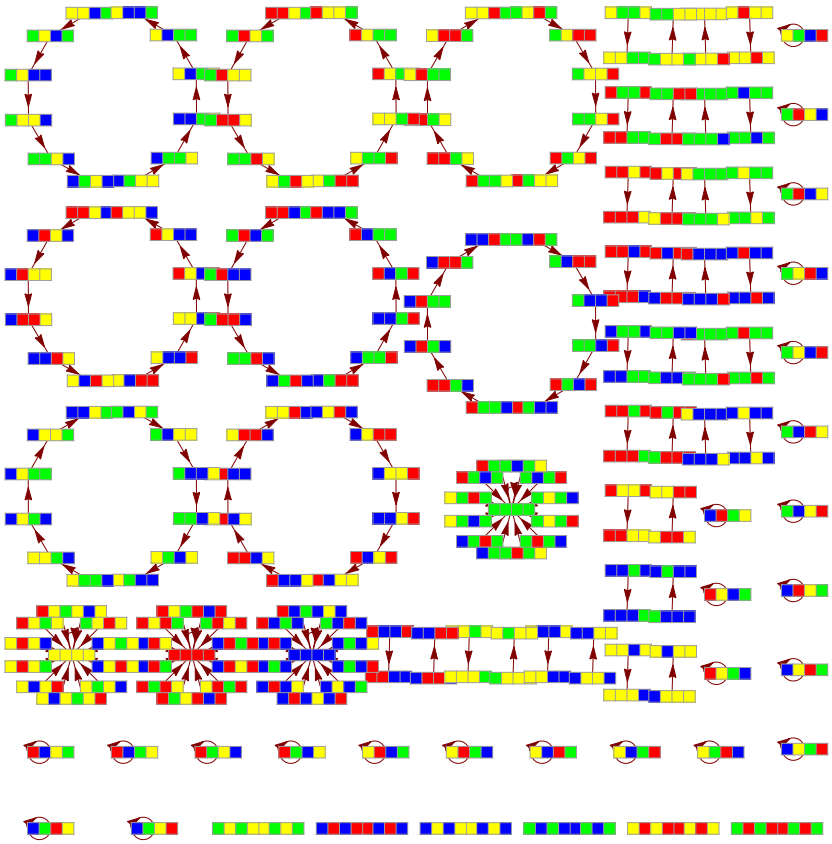




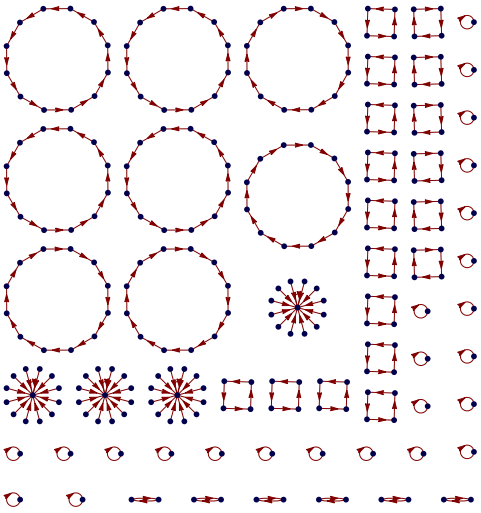
Transition graphs

Transition graph for ruleDM[{1,2,3,4,10}] -> “DM2”

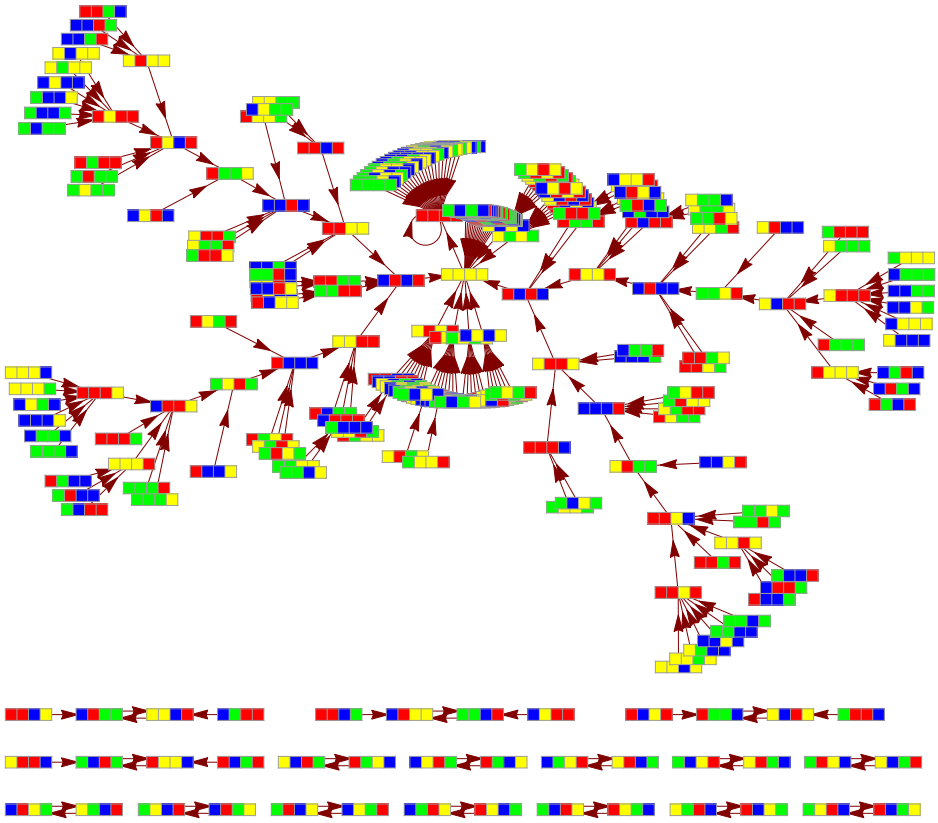
Colored transition graphs



Naked transition graphs



Transition graph for ruleDM 24 and ruleM 24, width = 4



ruleM 24, ruleDM 24, width = 3

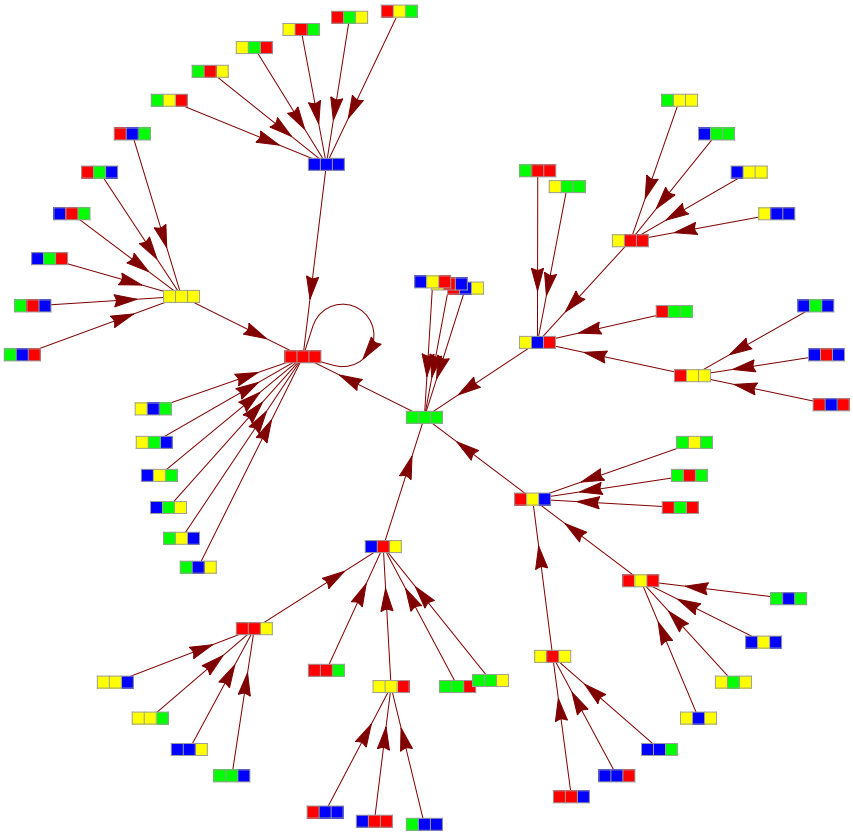
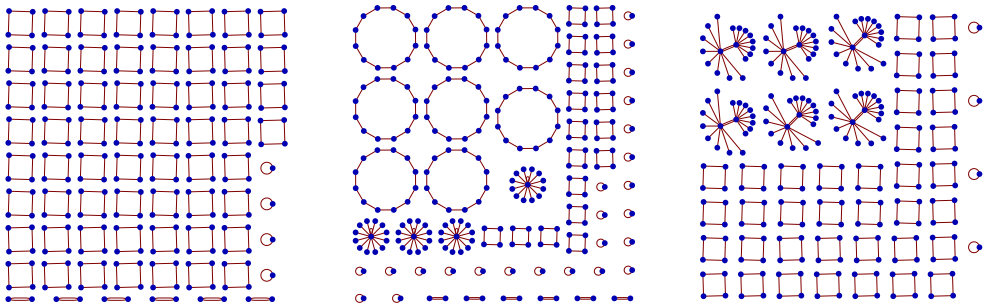
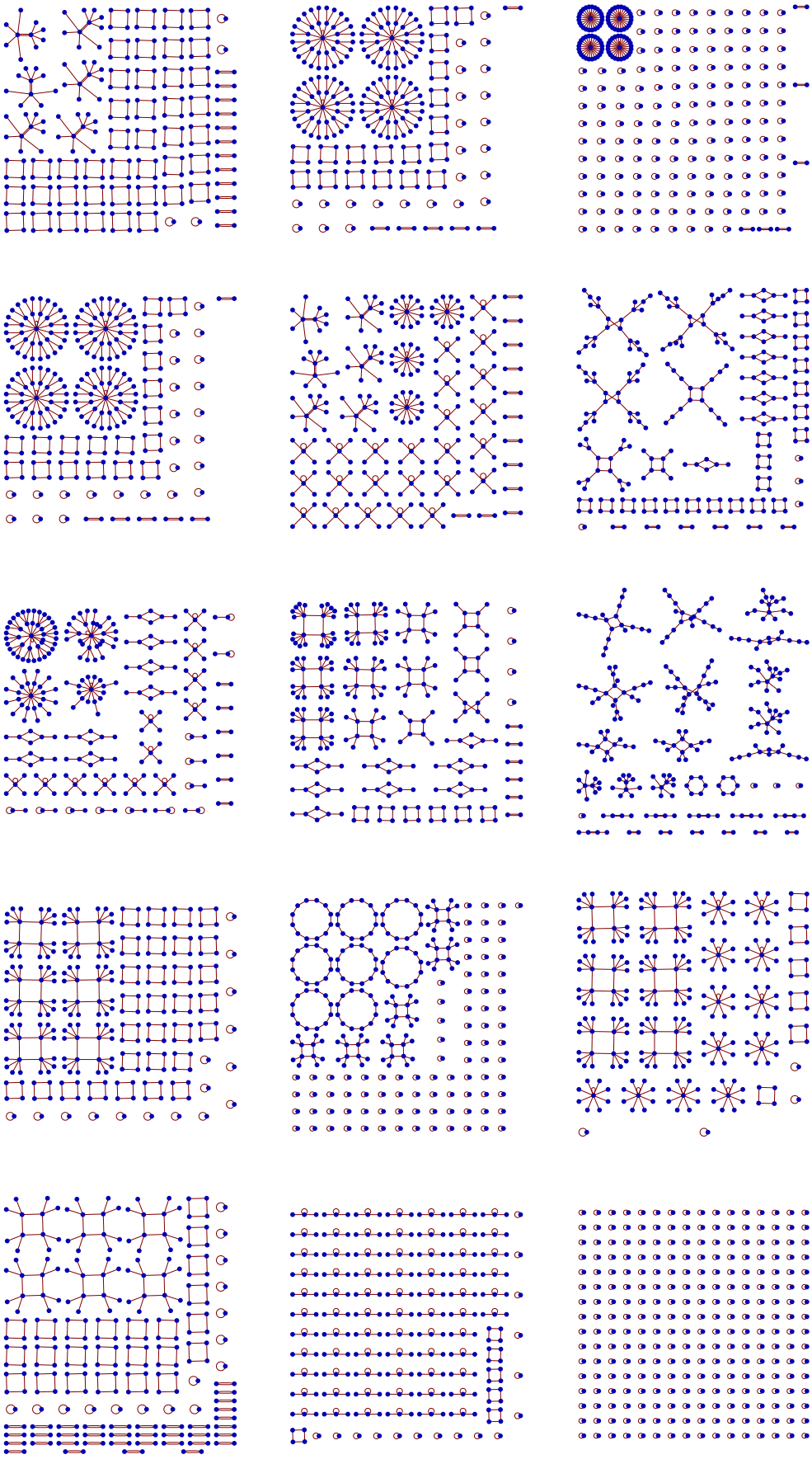
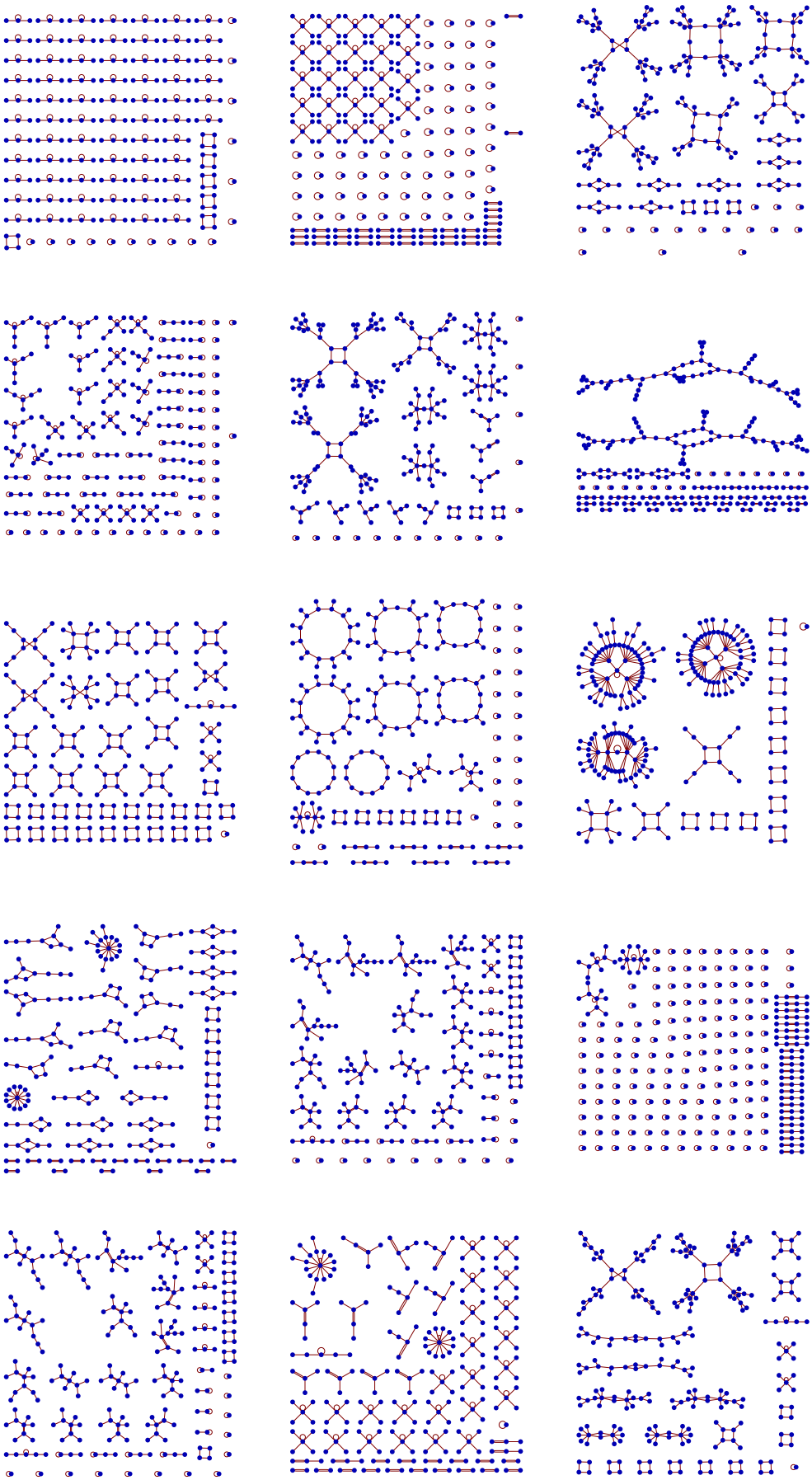


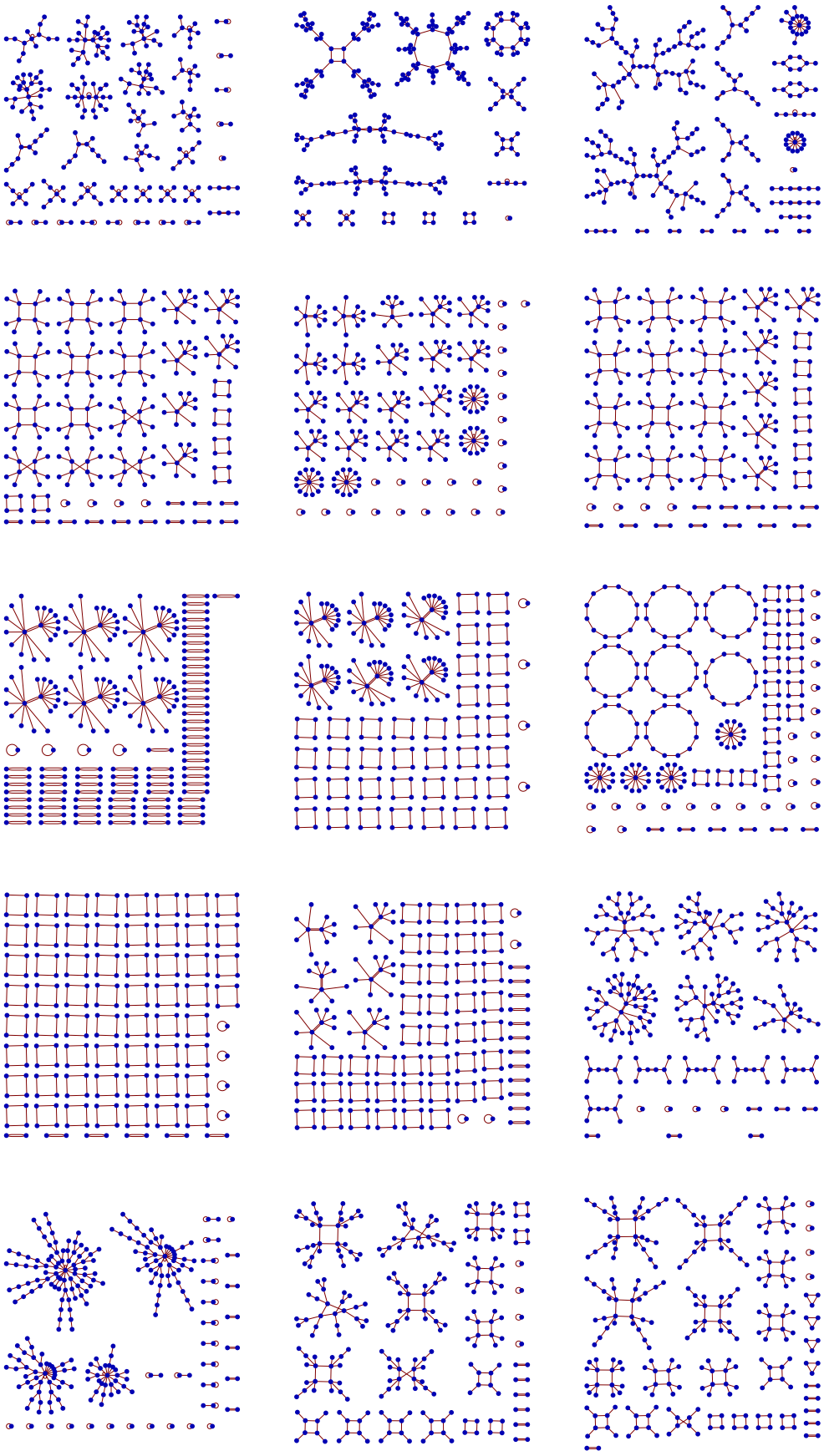
Table for ECA:
http : // atlas.wolfram.com/01/01/views/40/TableView.html

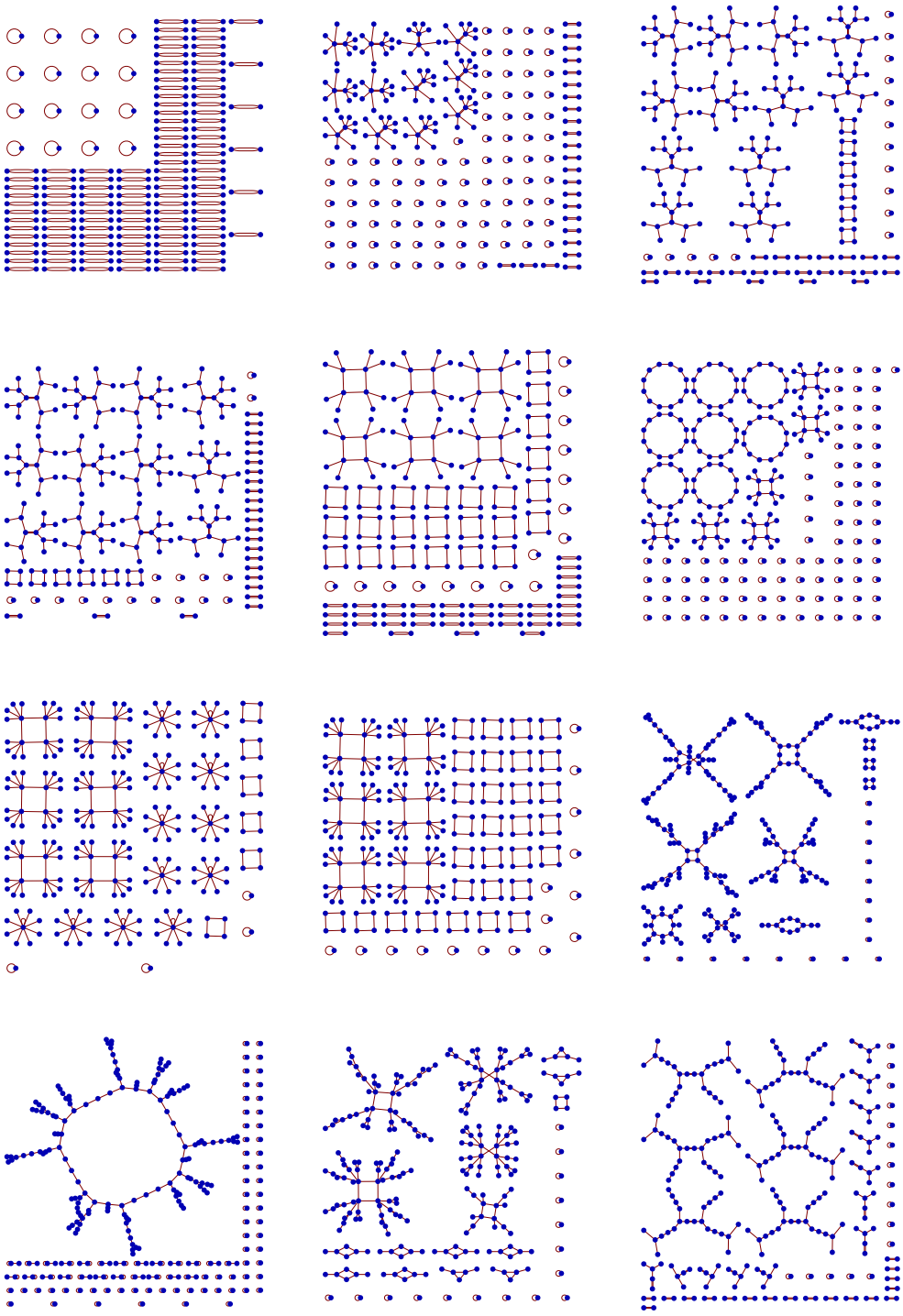
Transition table for ruleDM, w=4

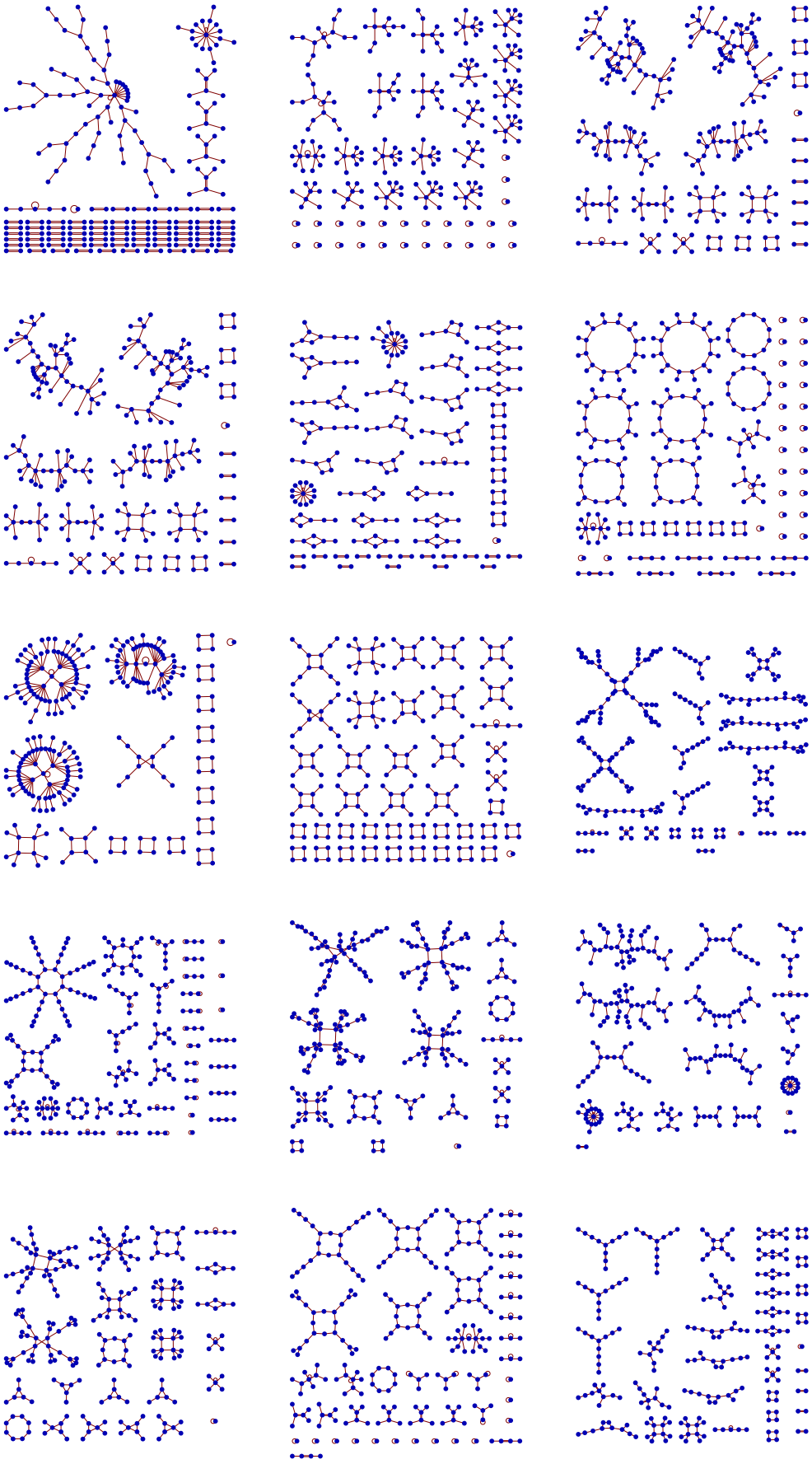


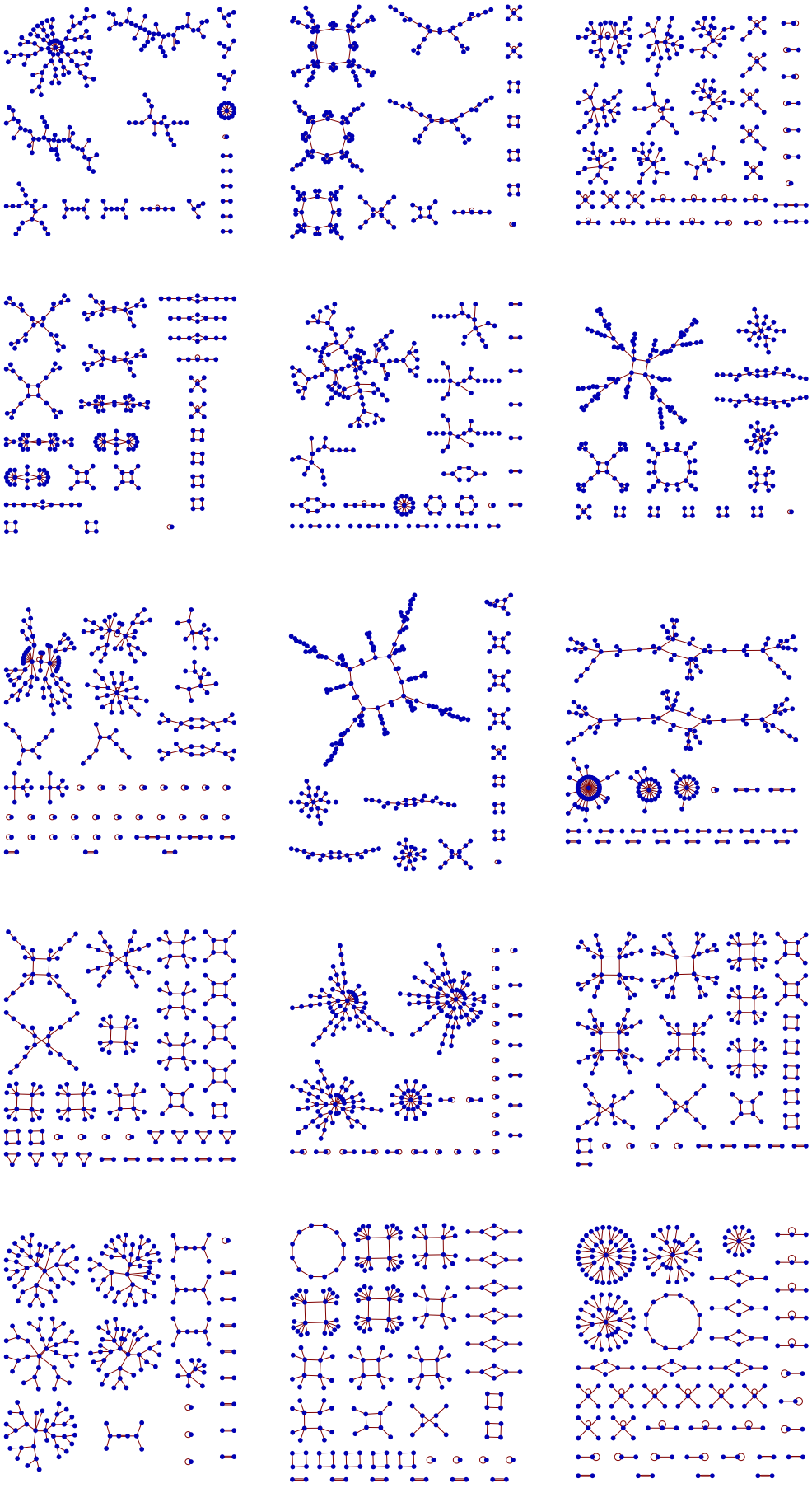


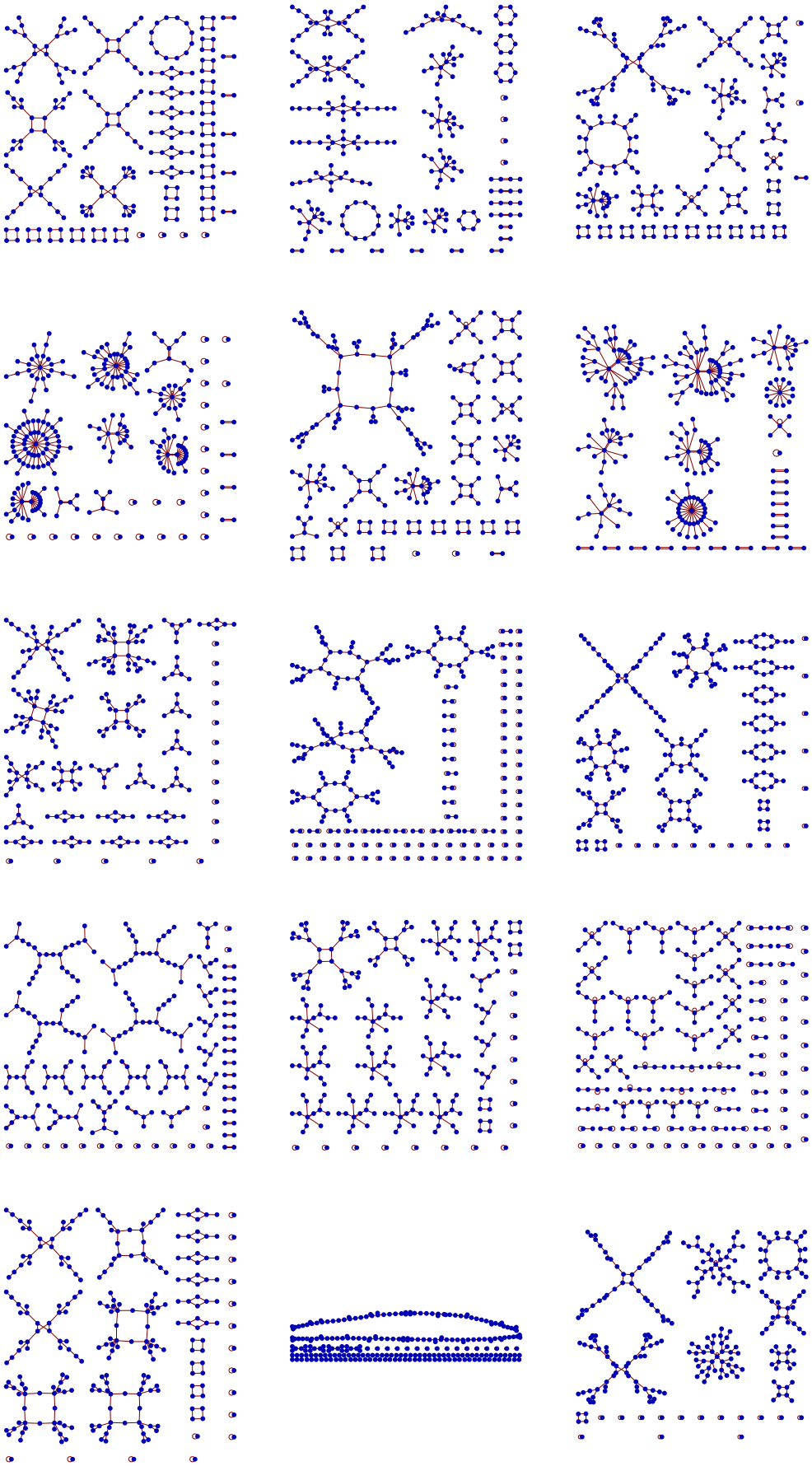


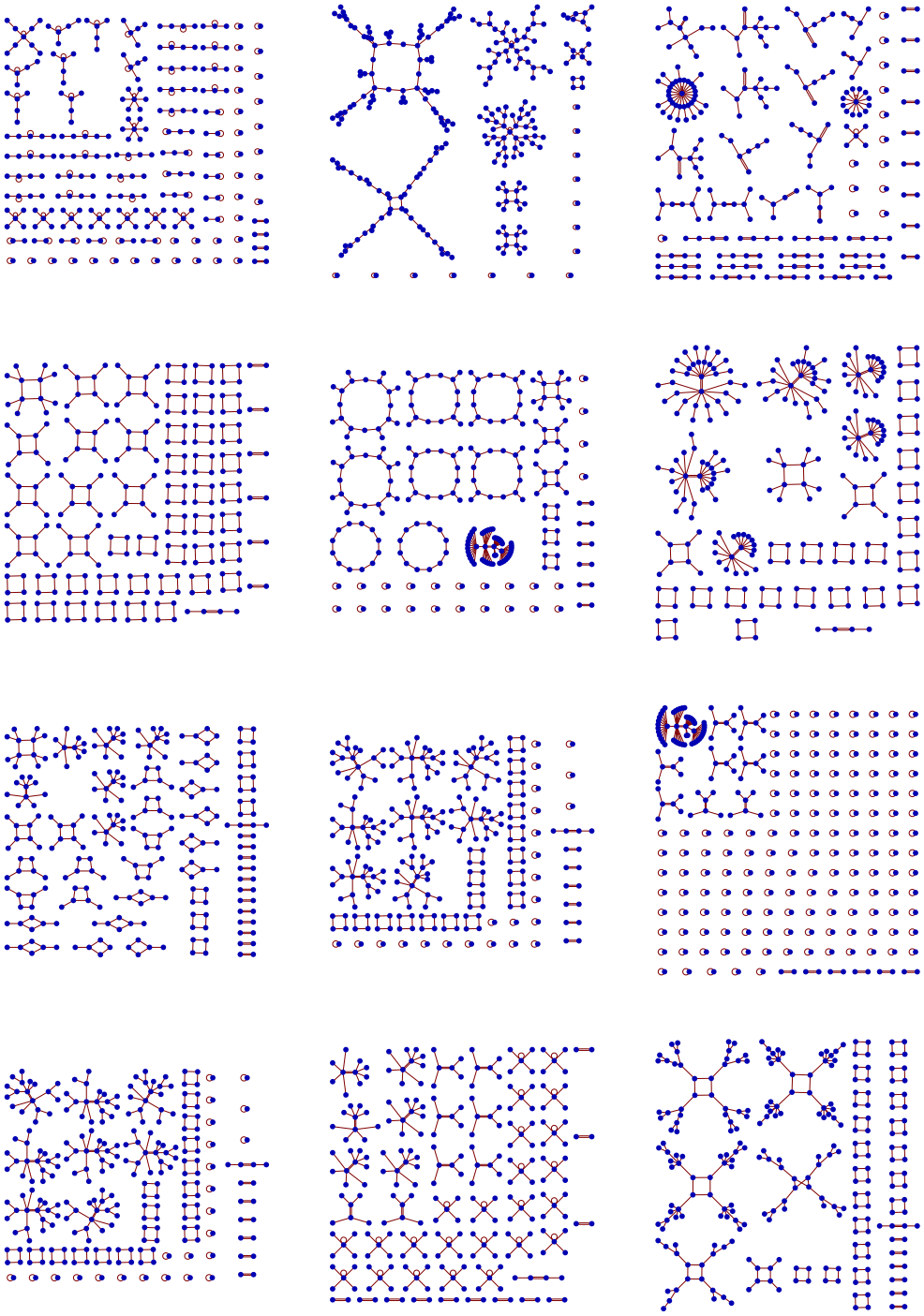


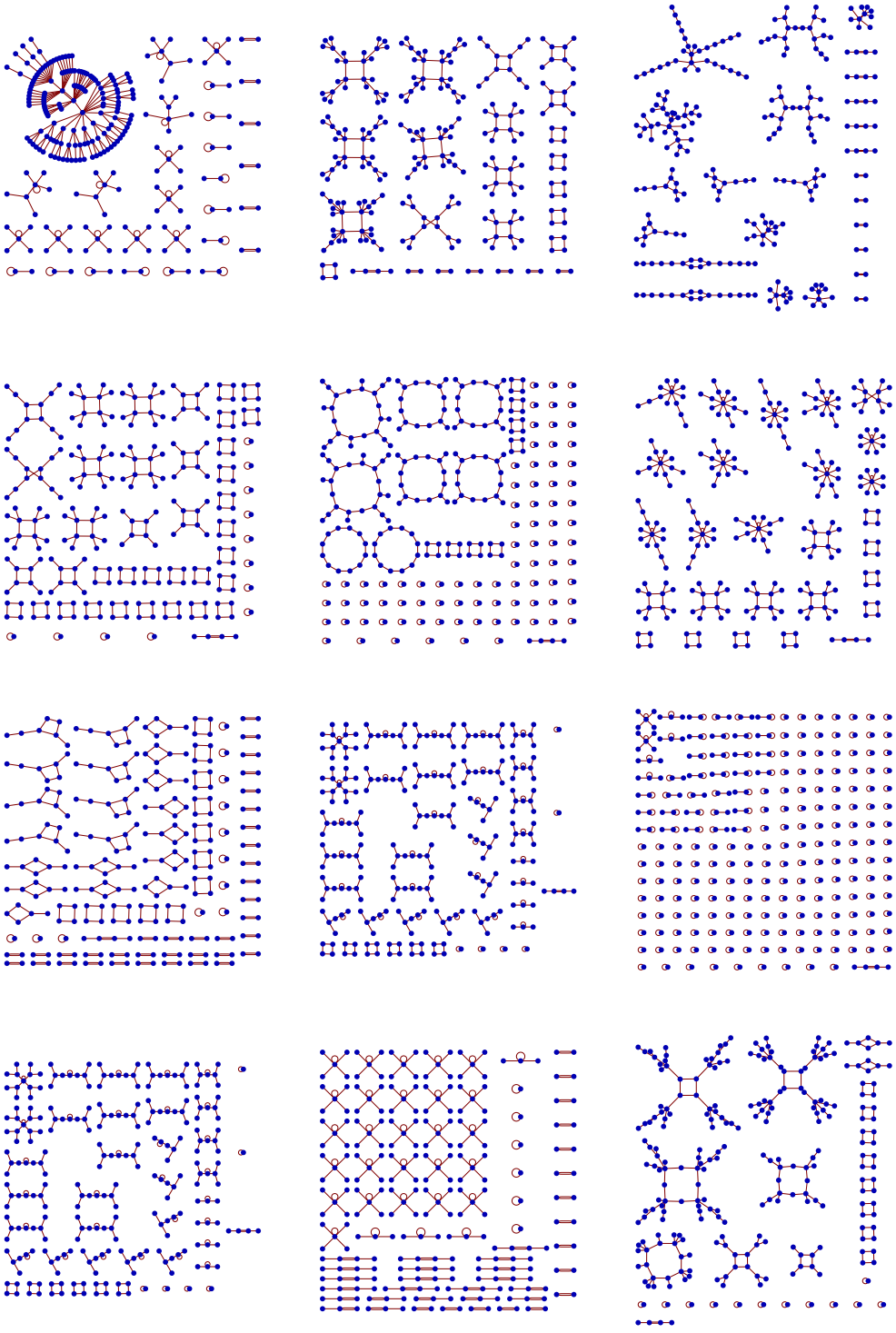


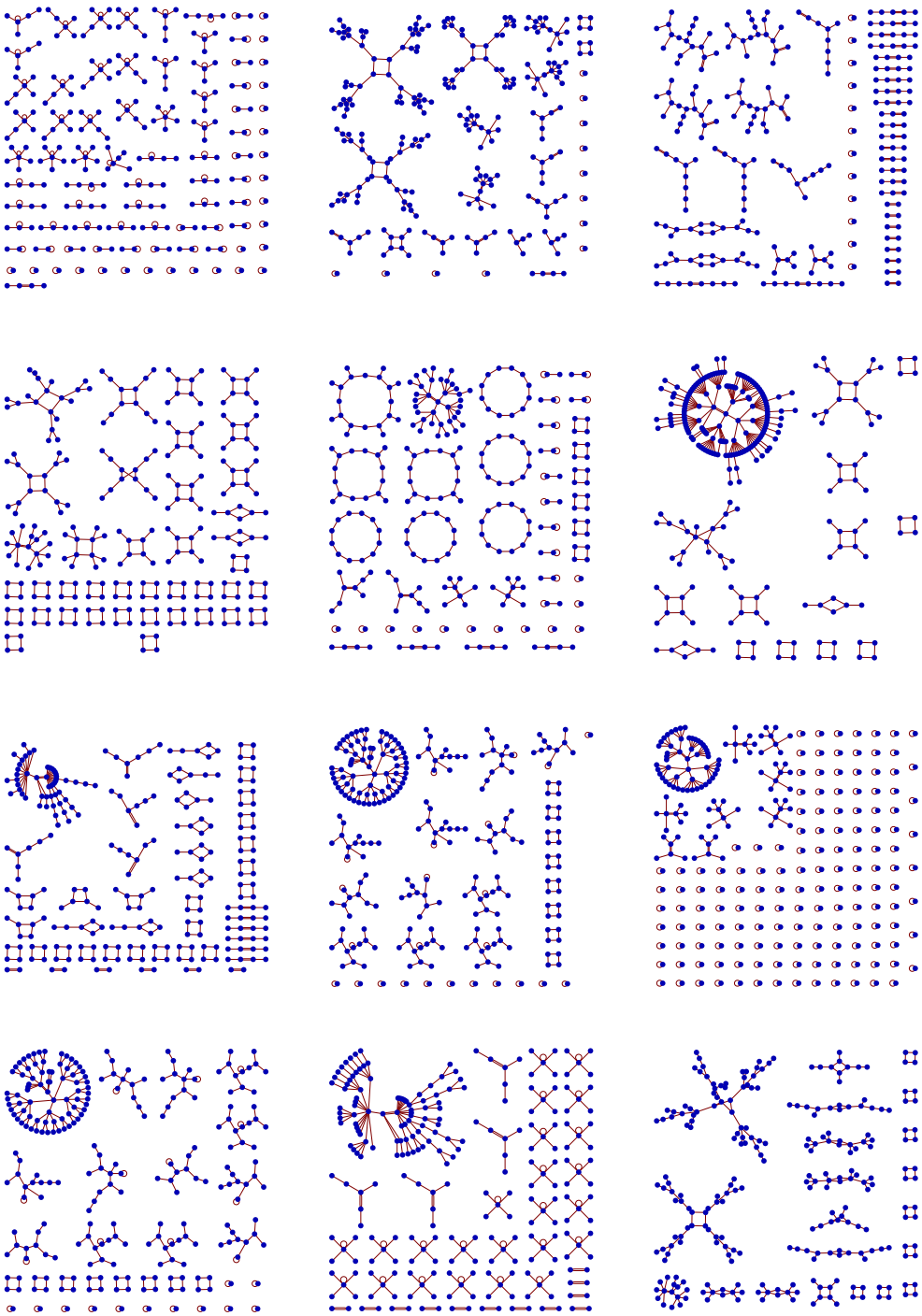


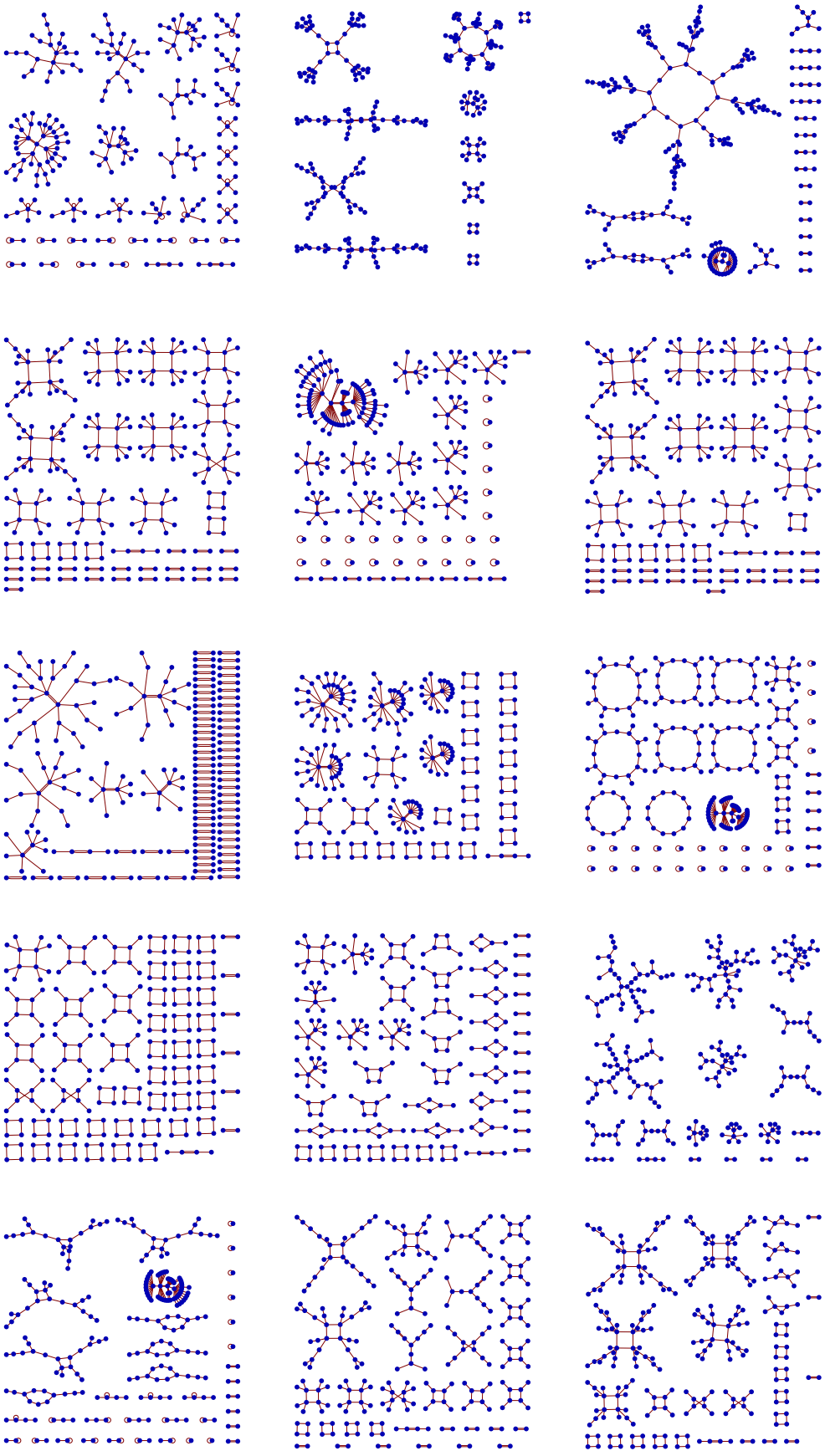


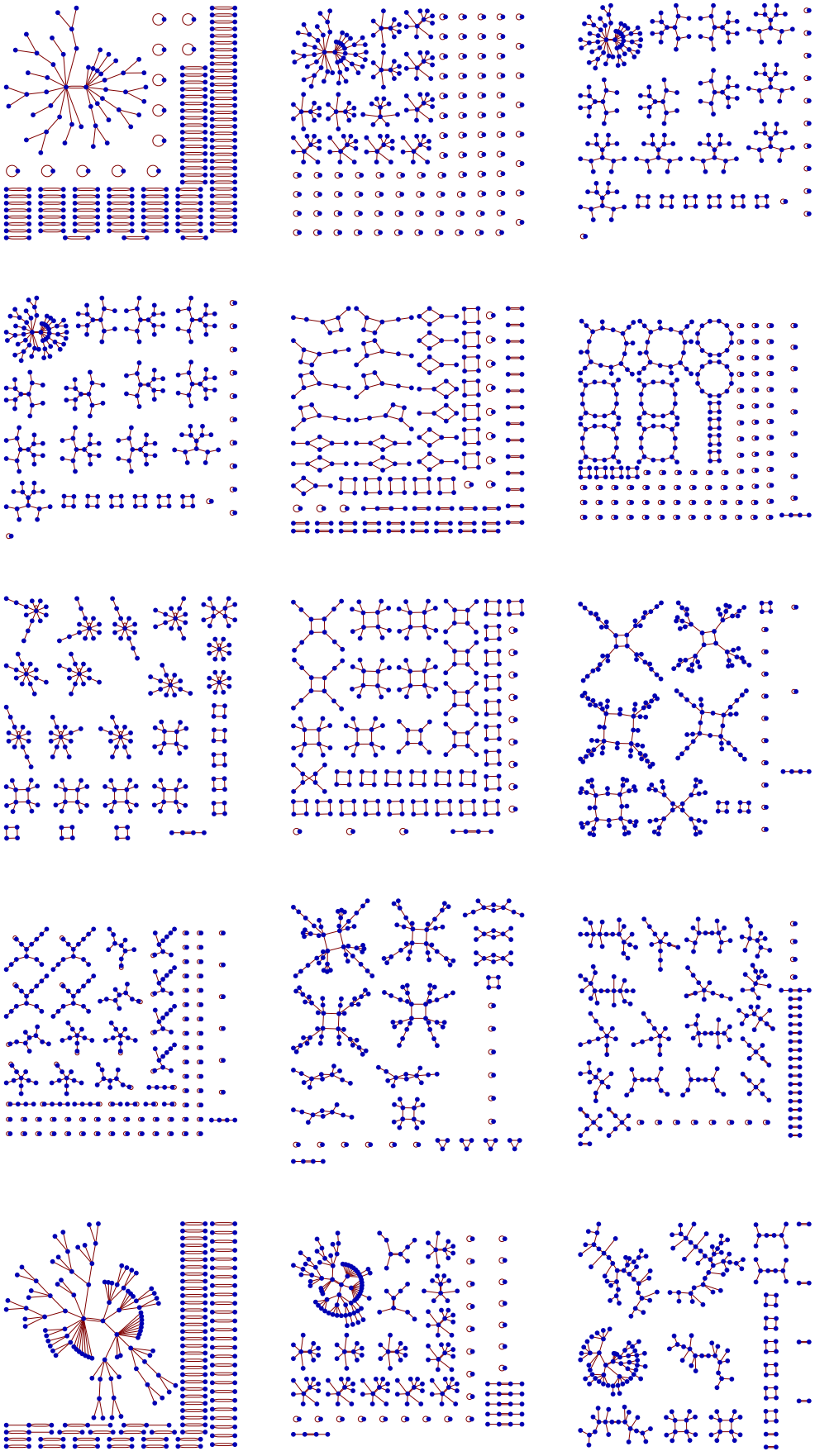


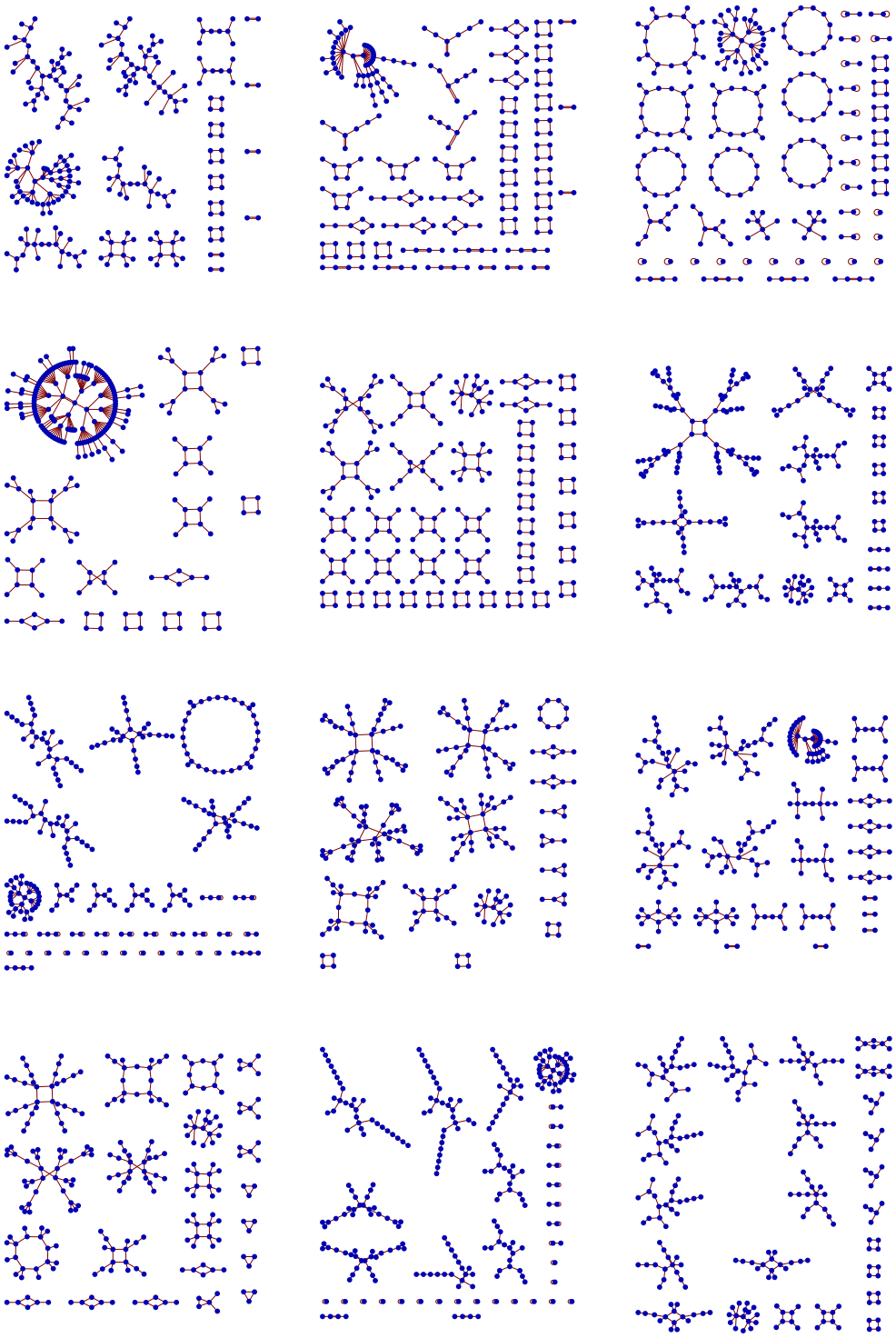


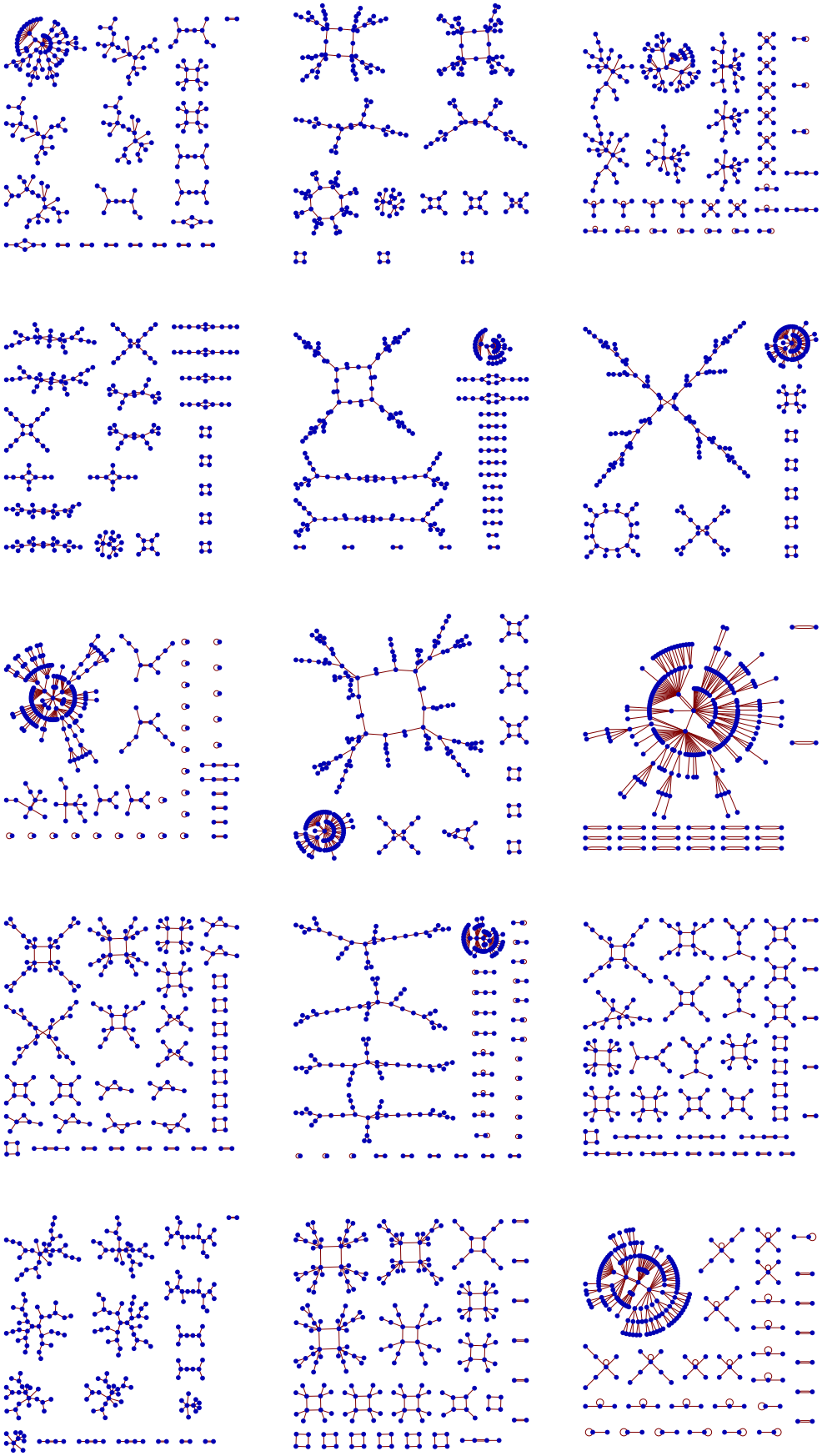


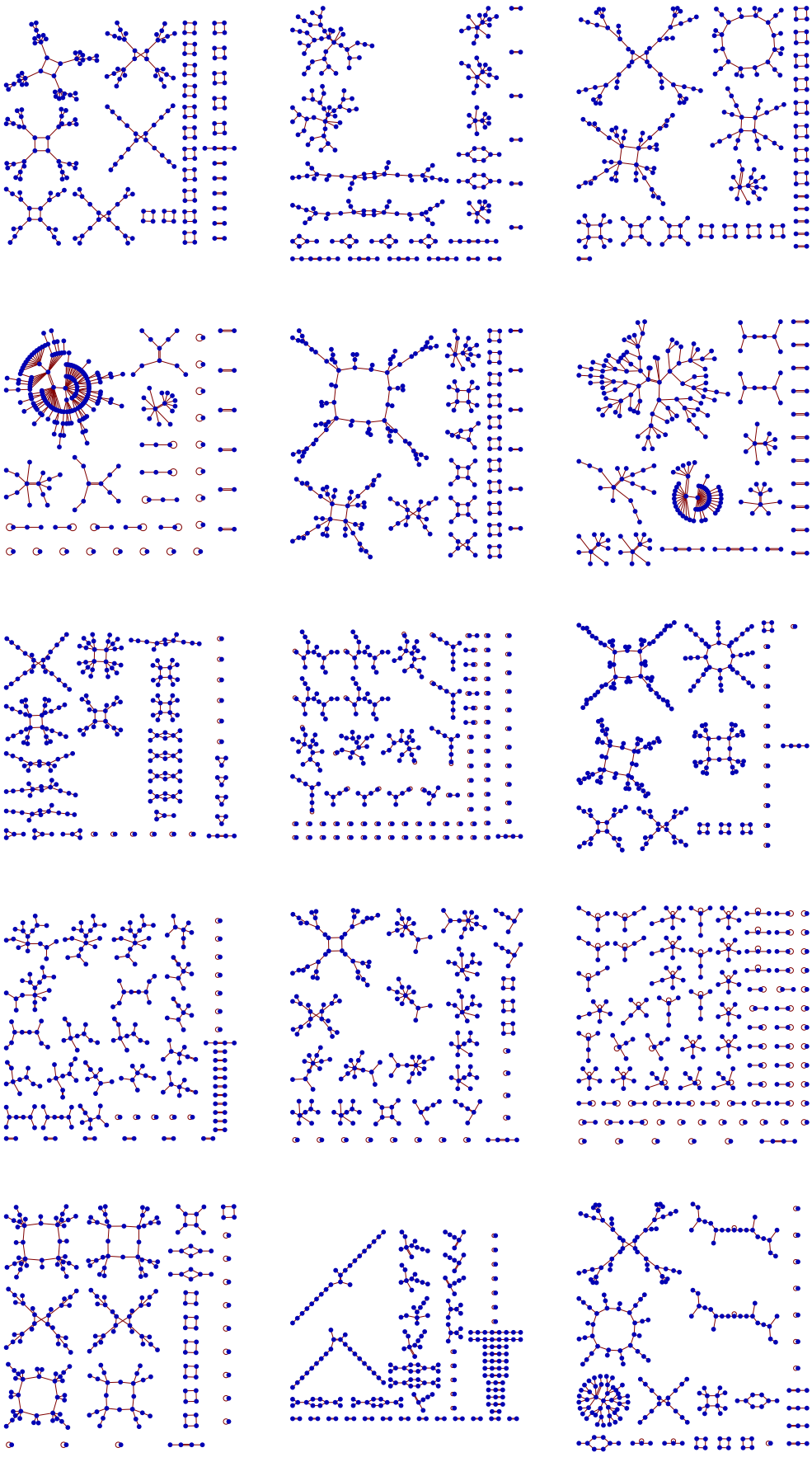


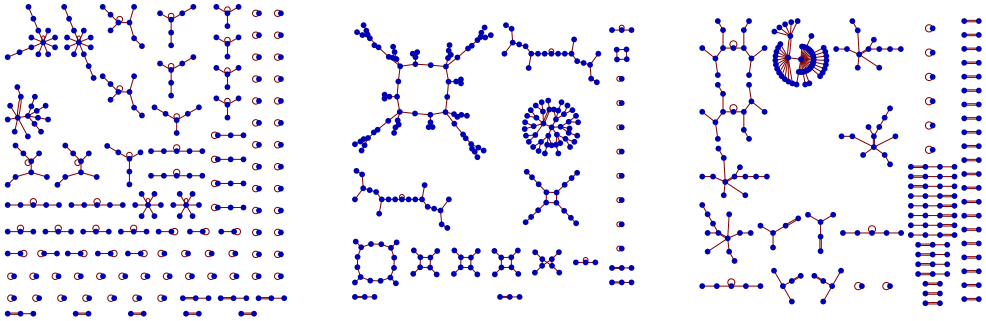




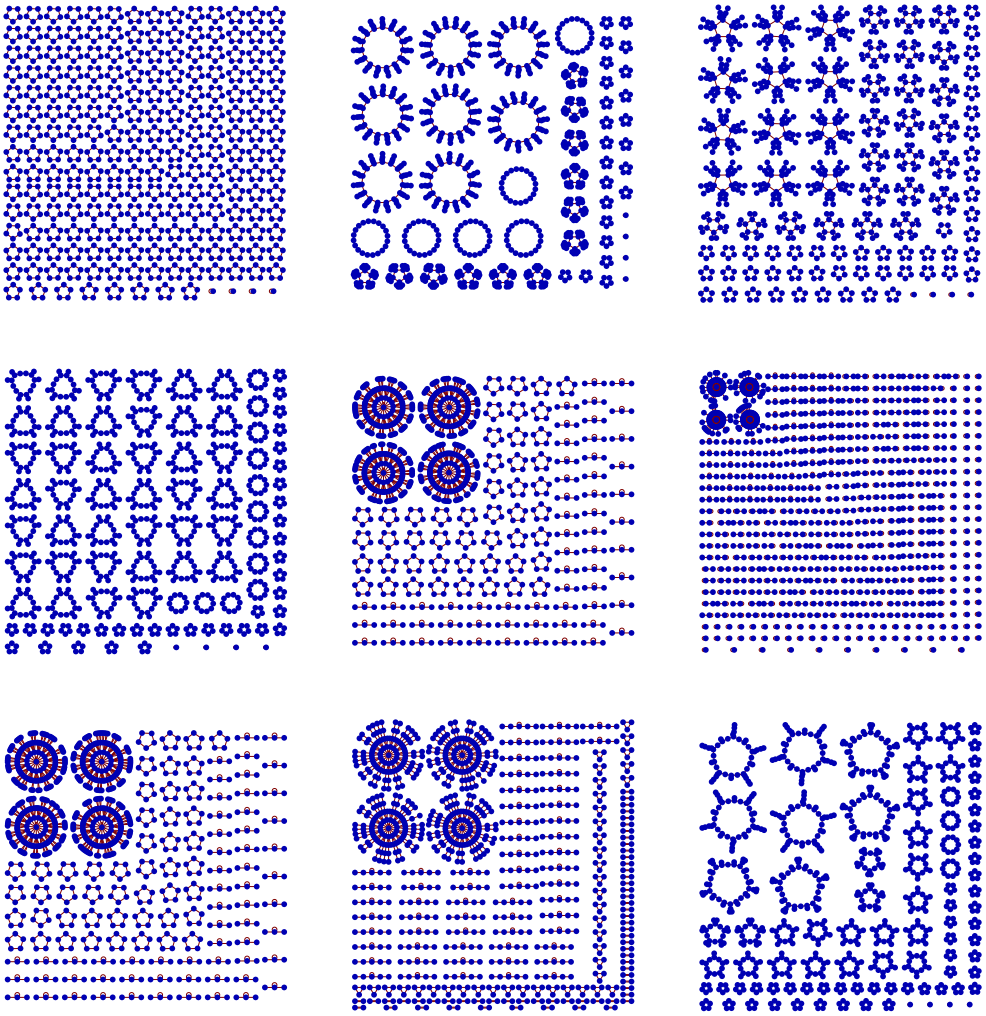


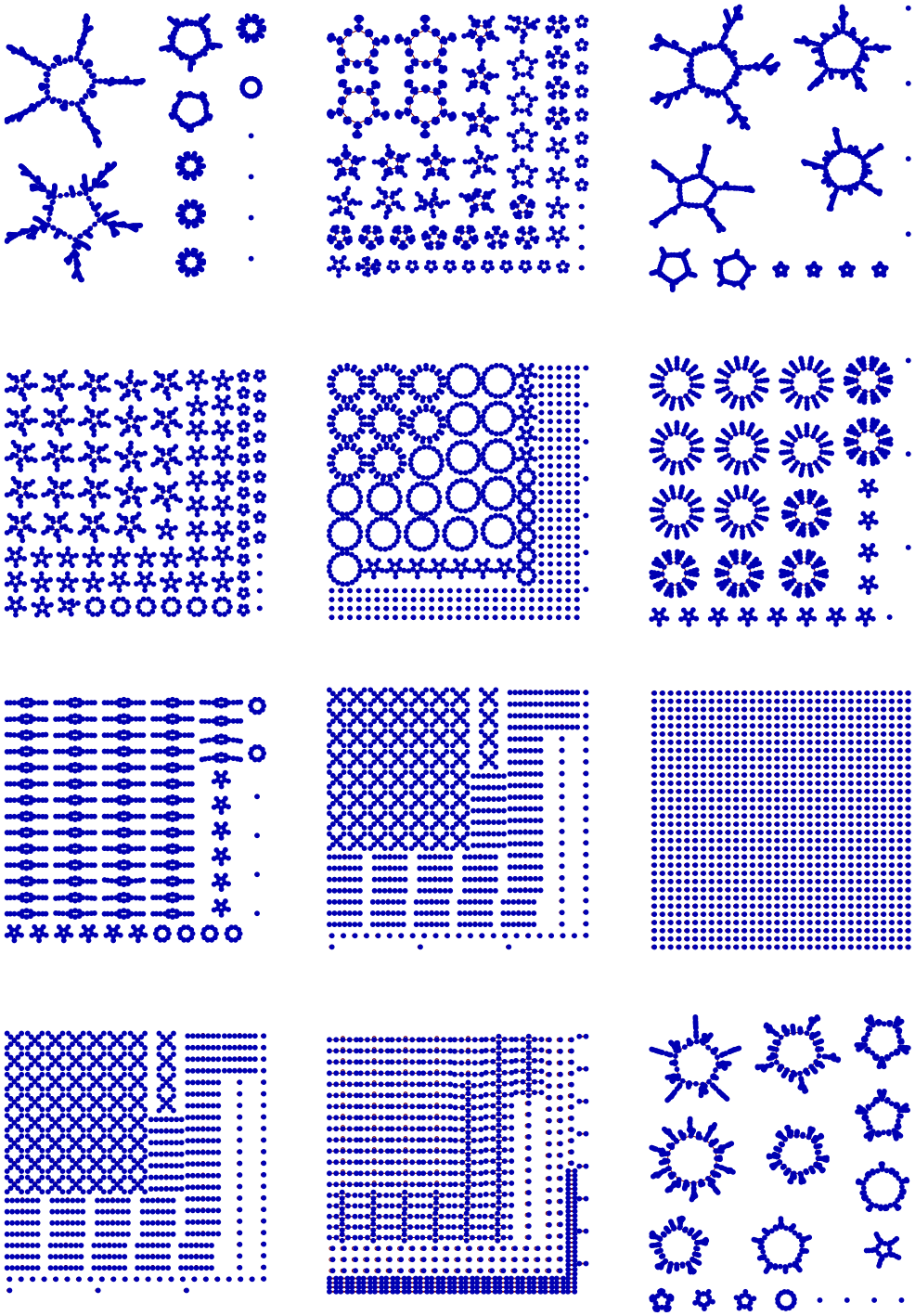


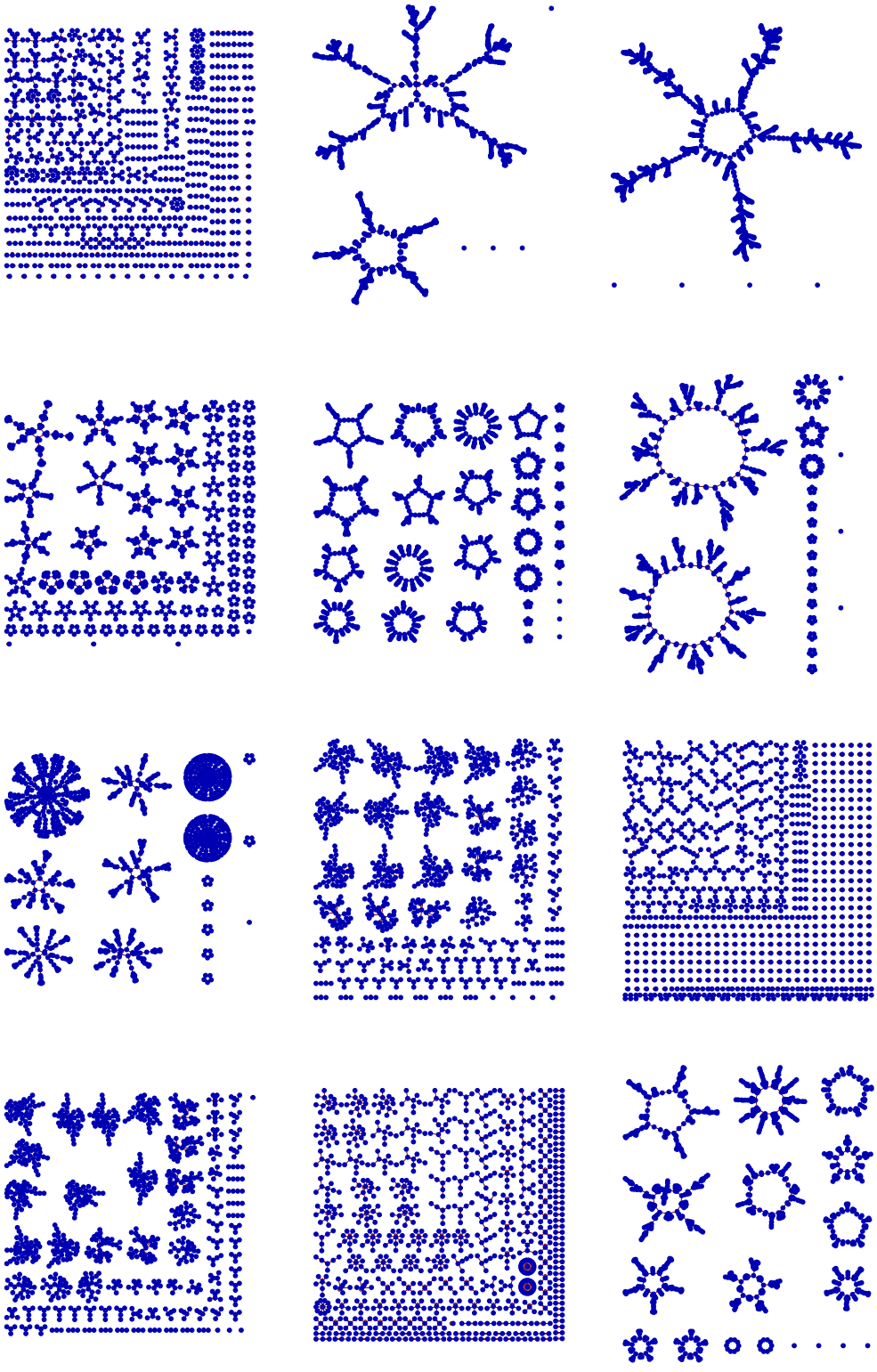


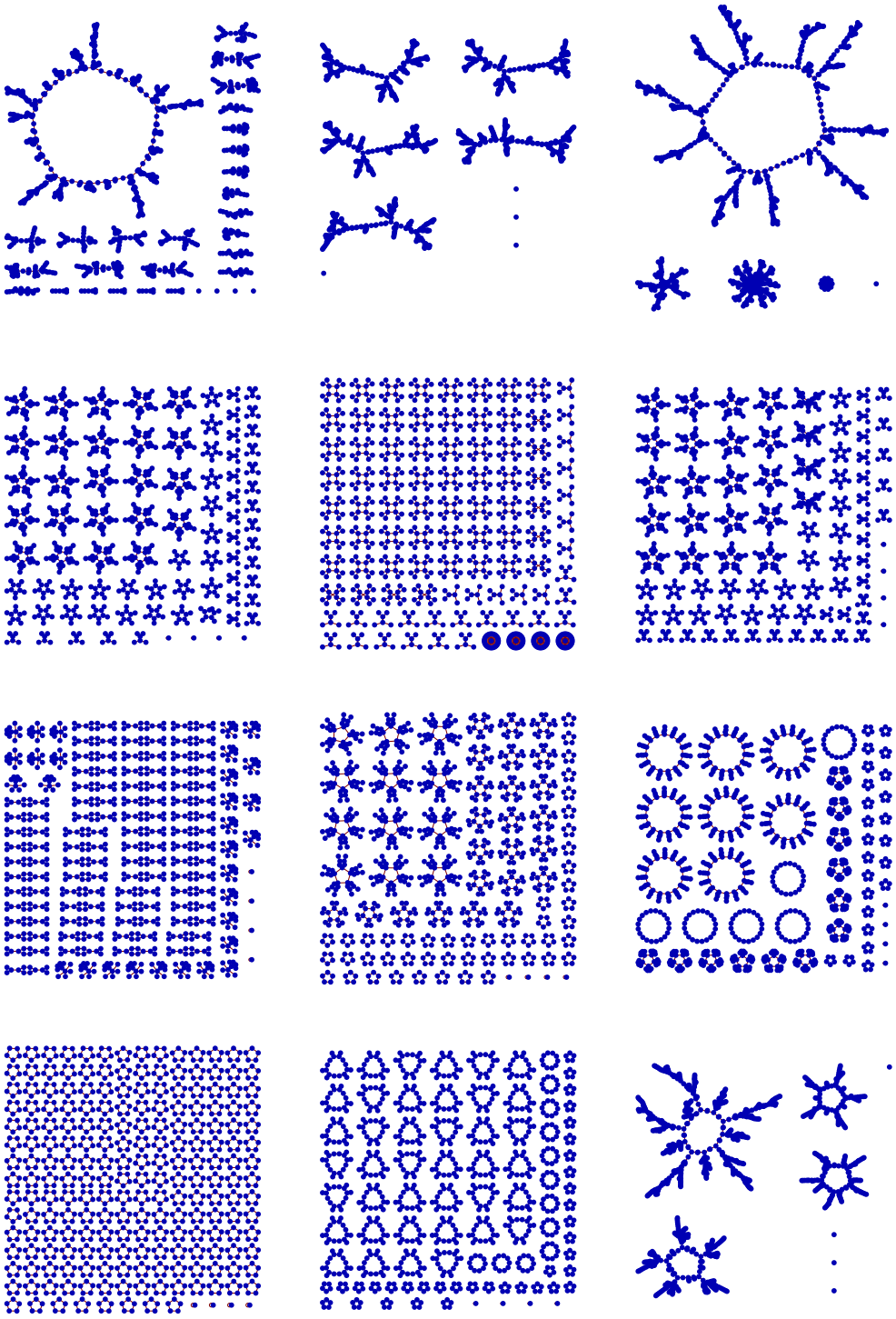


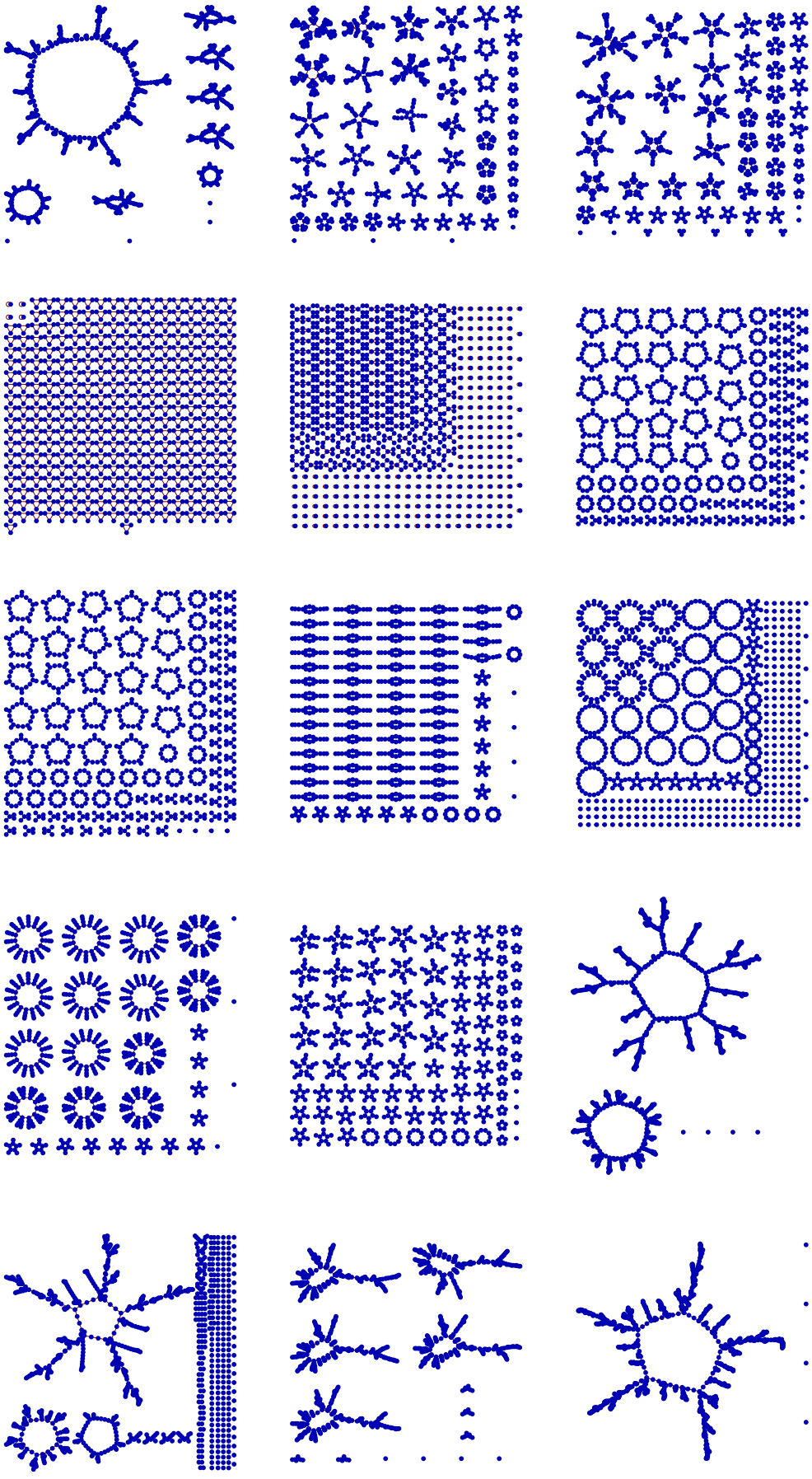
Transition table for ruleDM, w=5

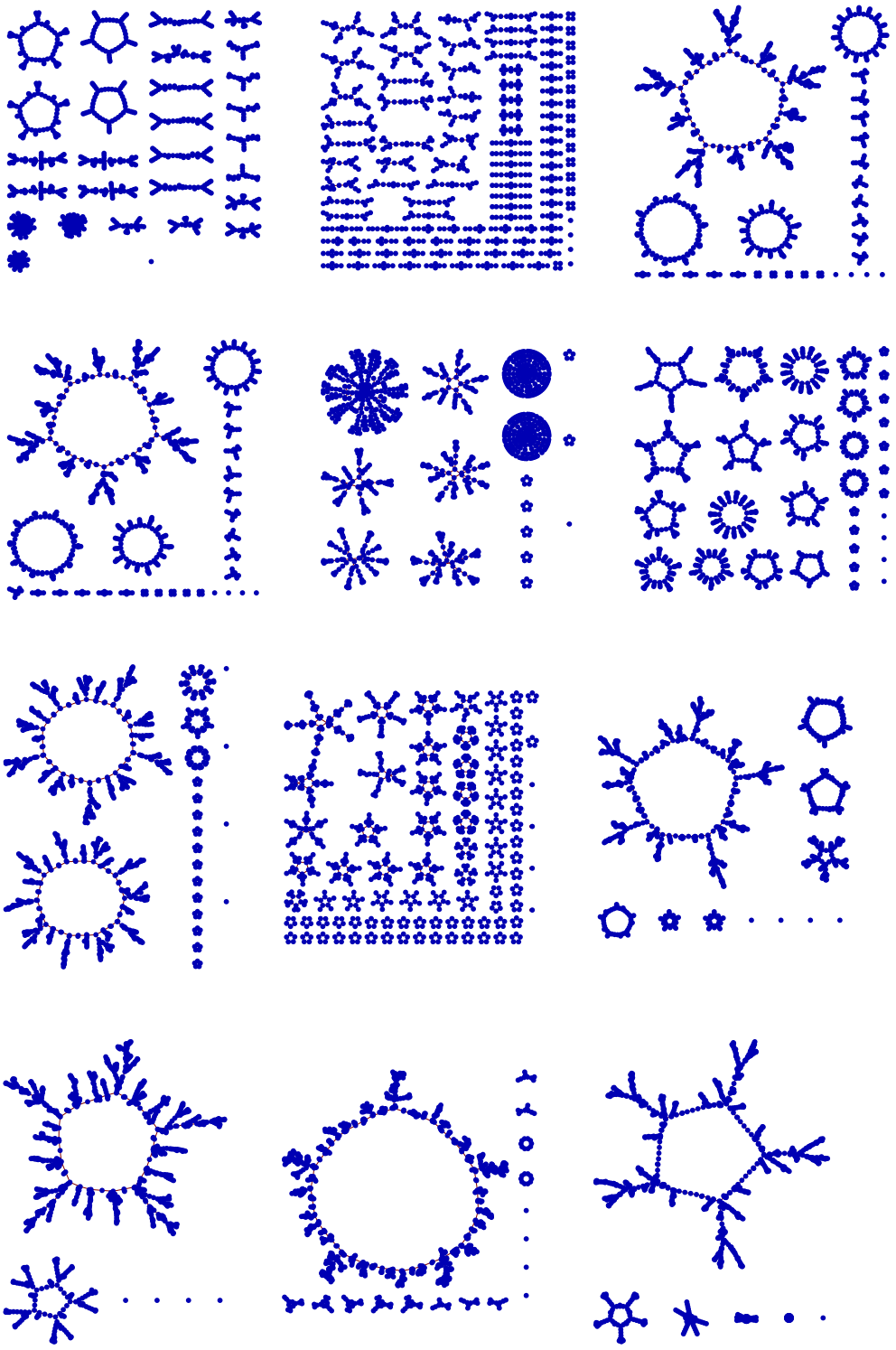


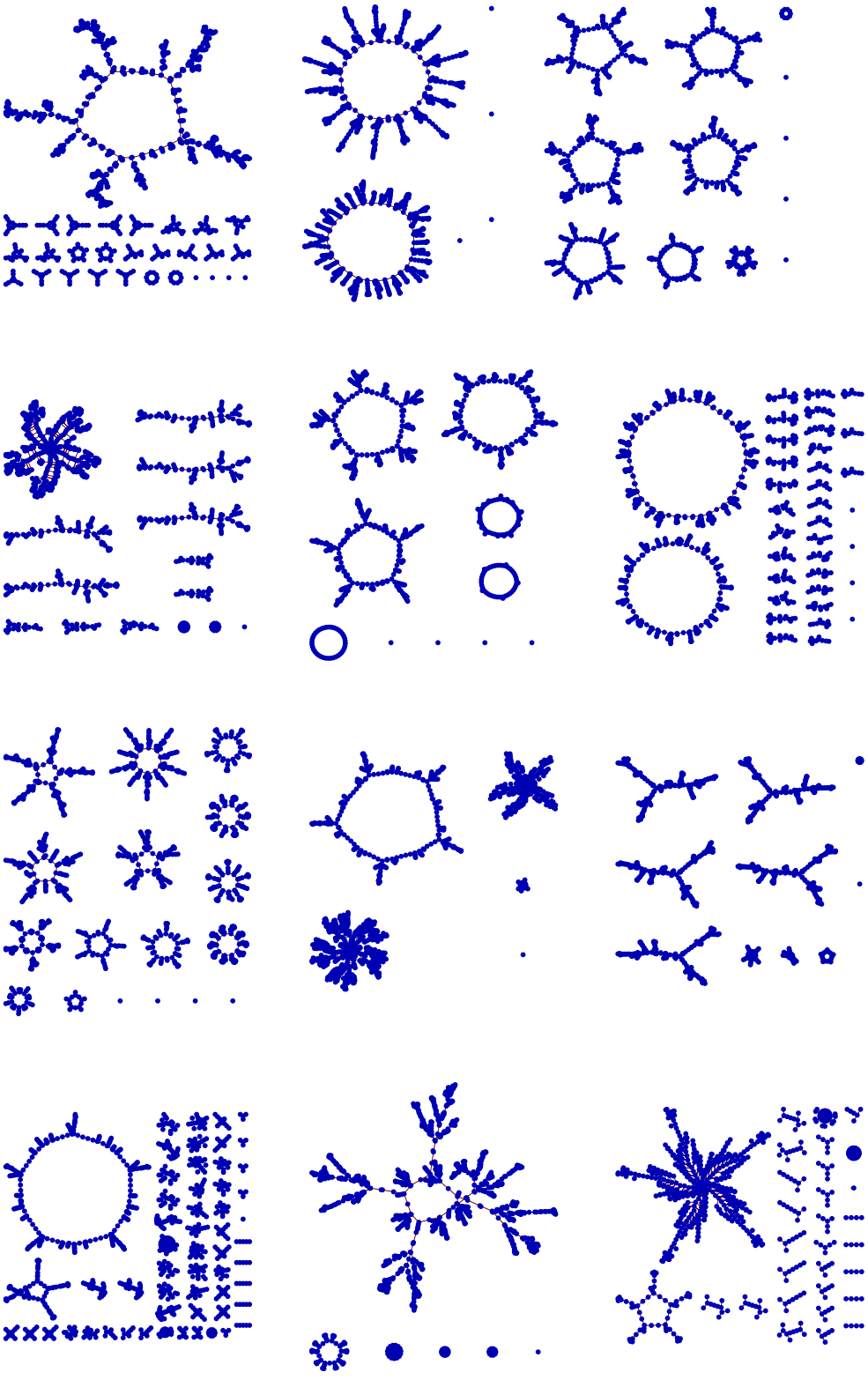


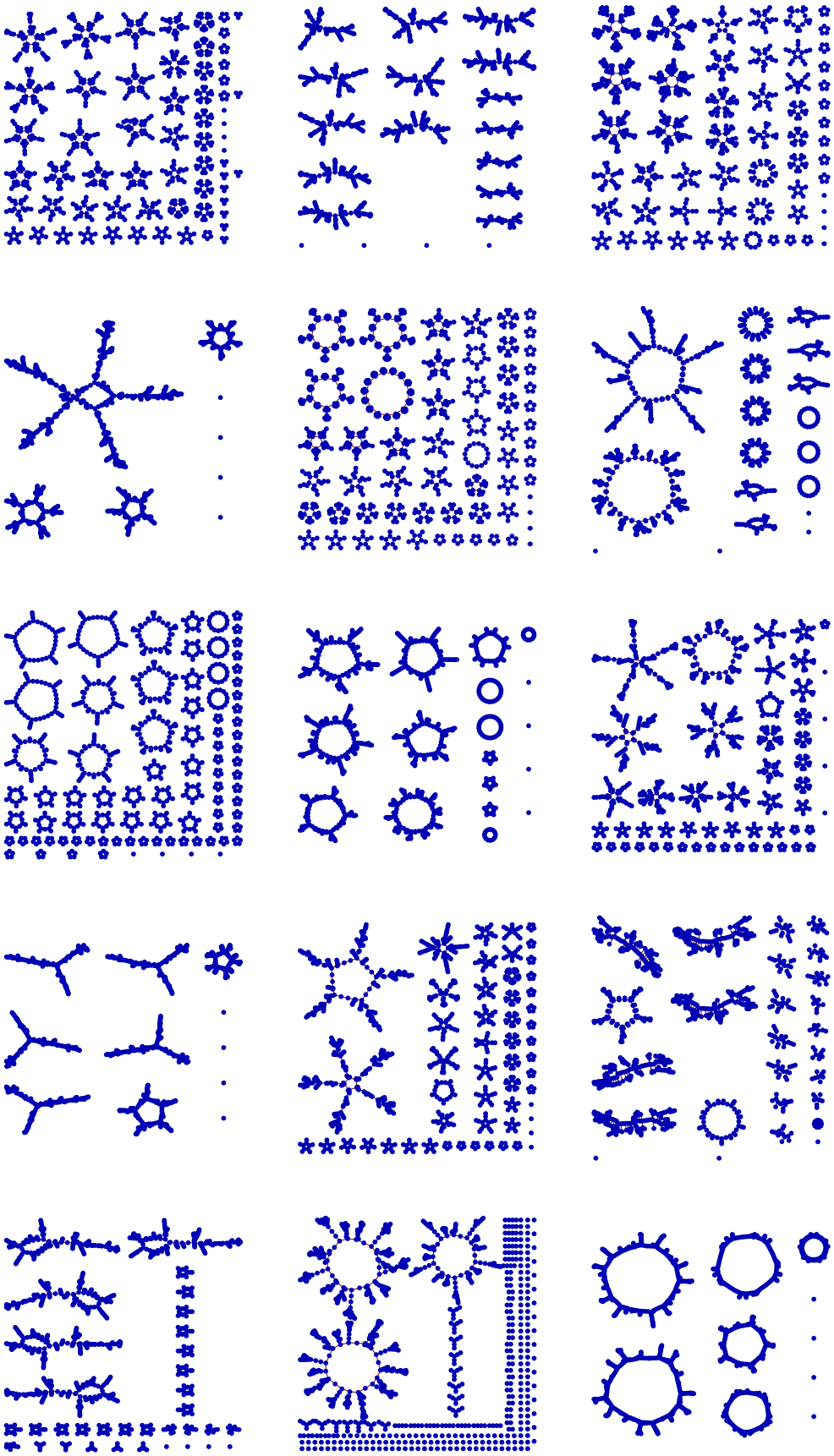


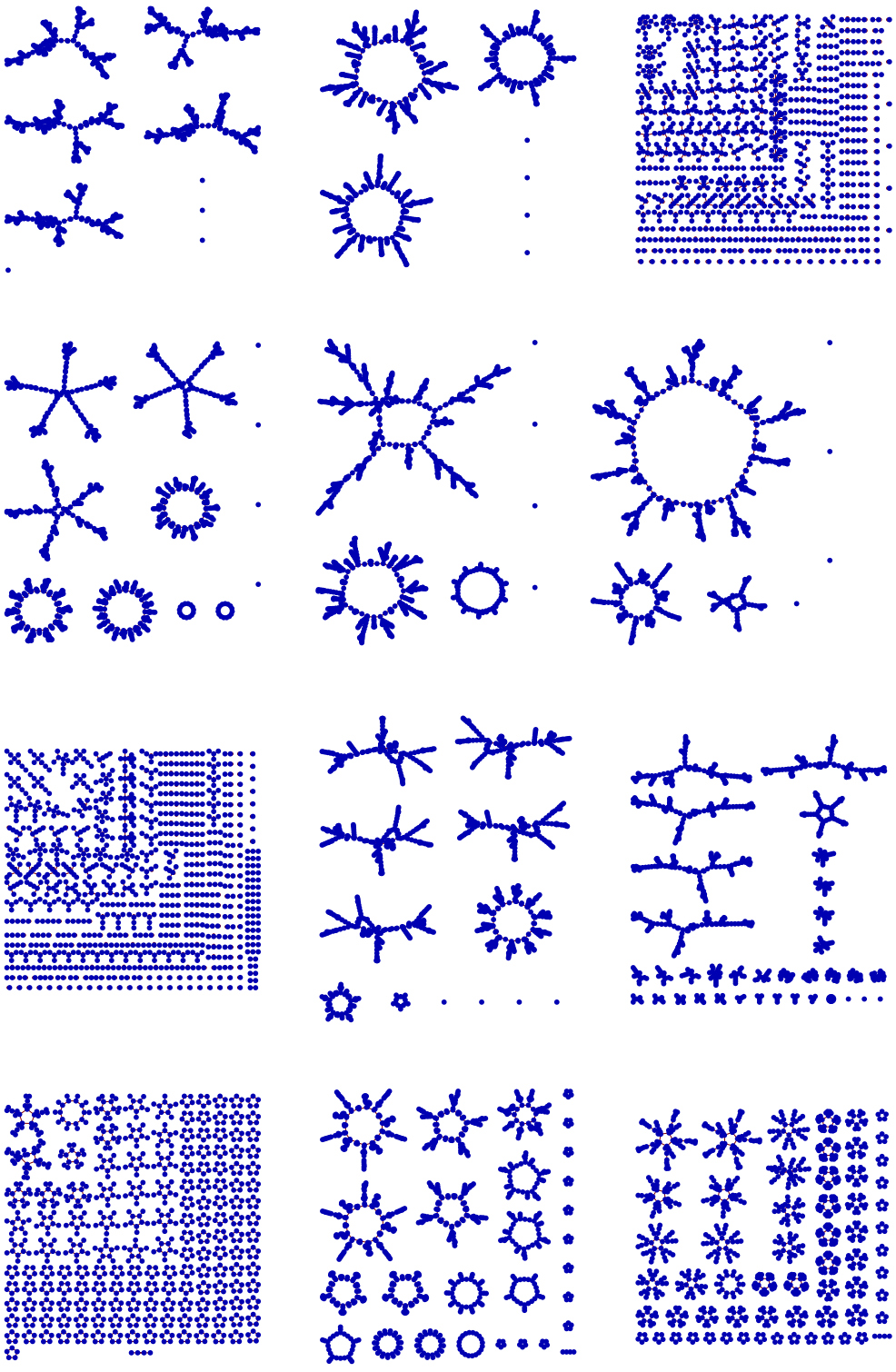


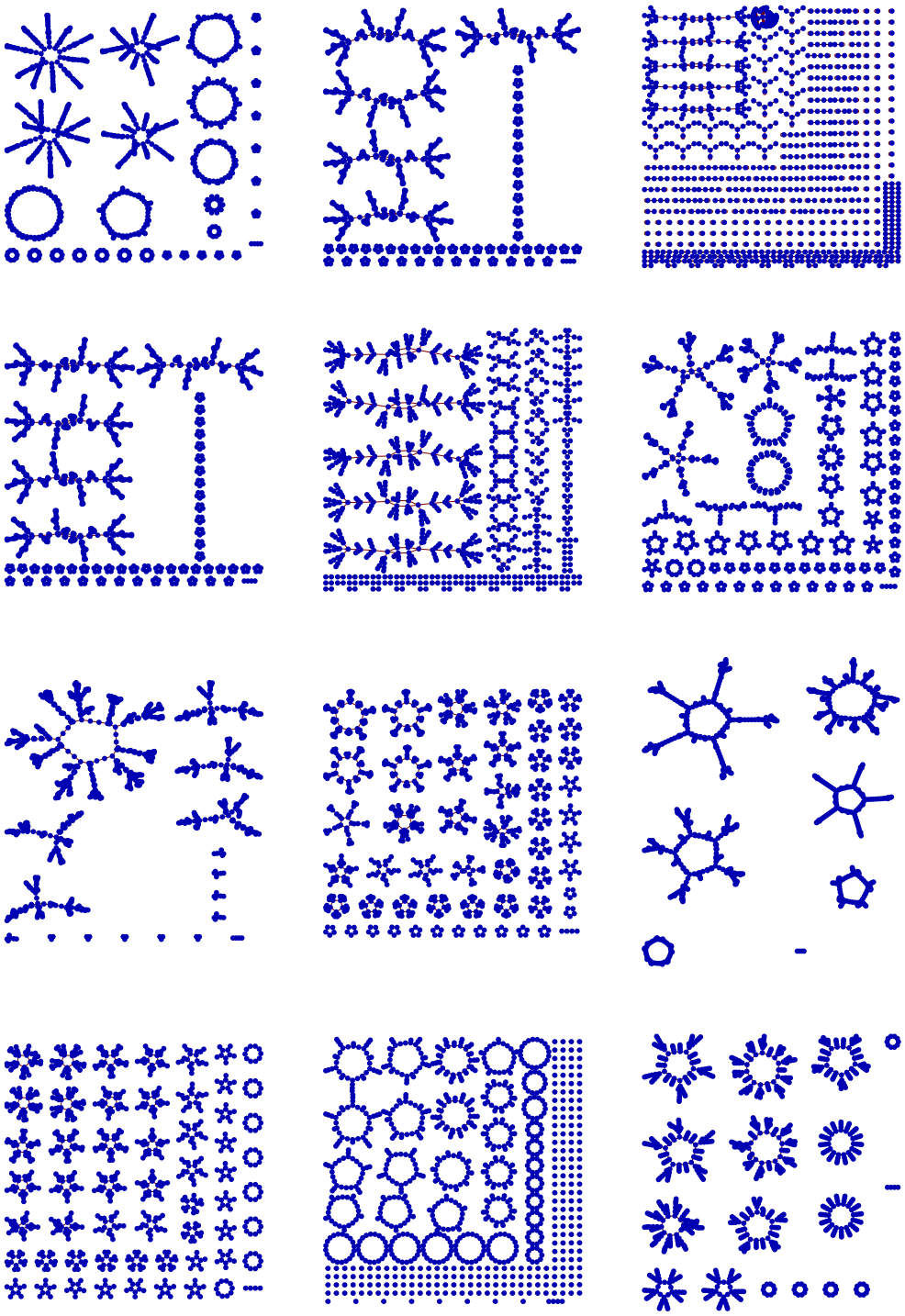


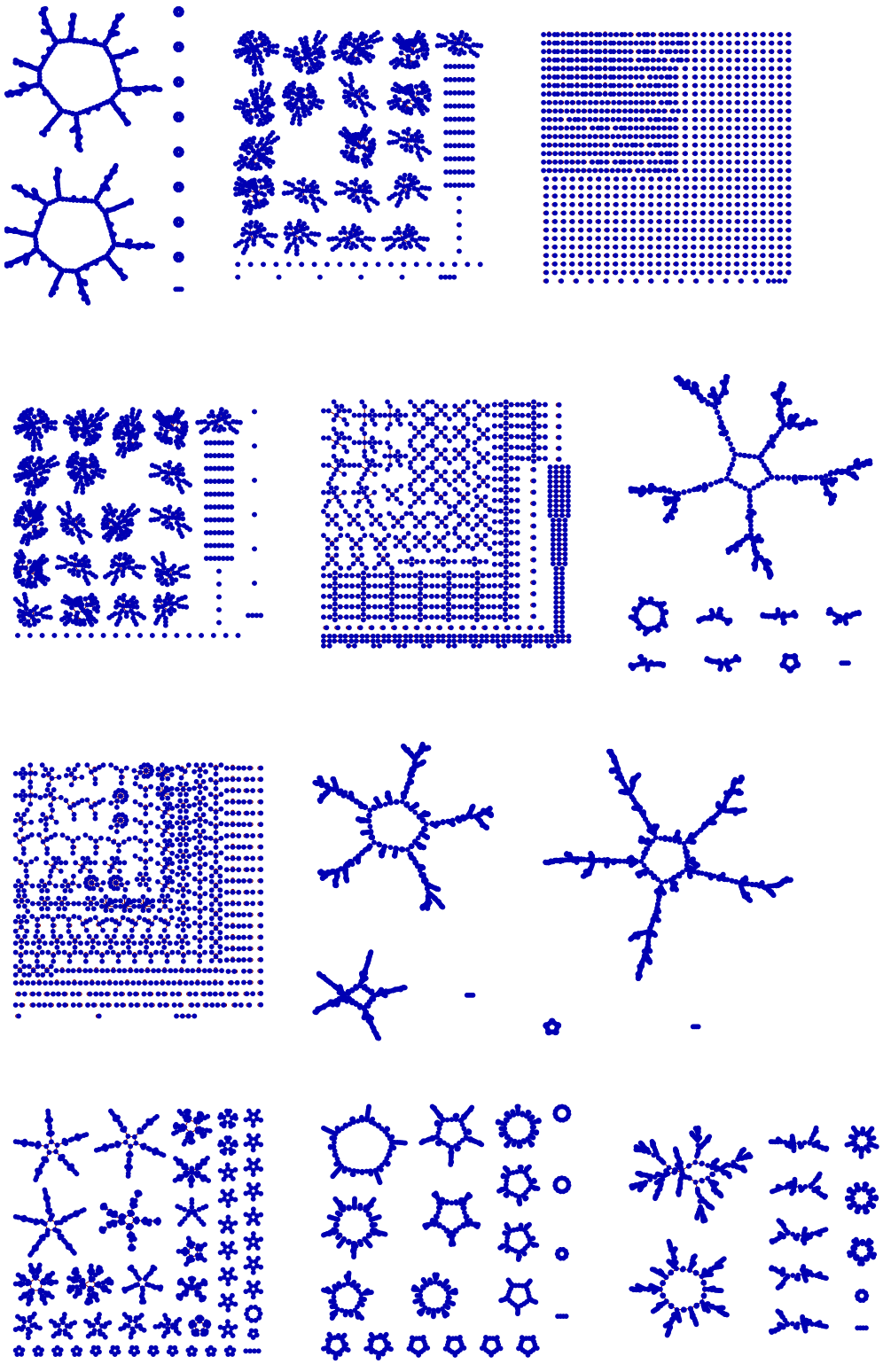


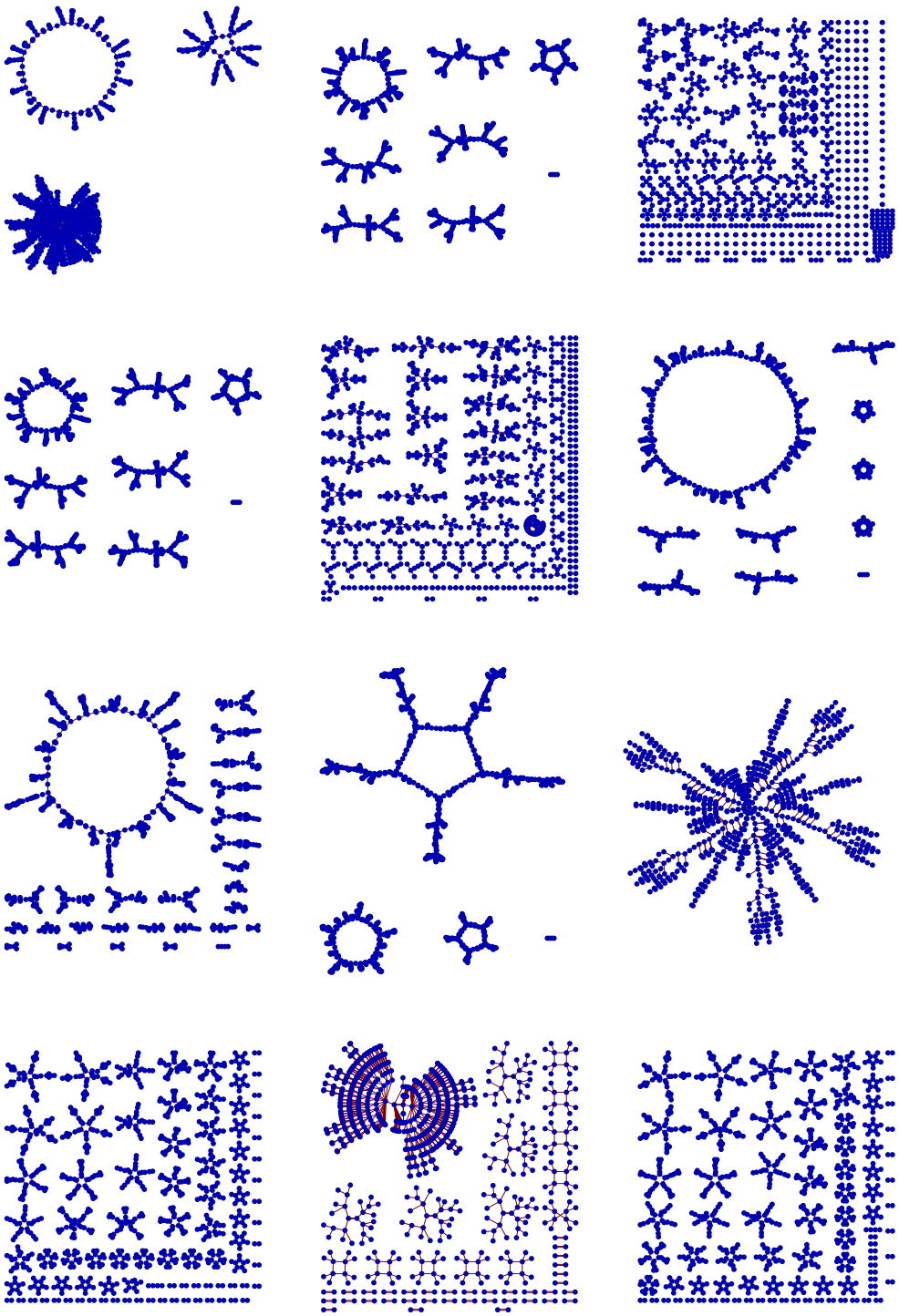


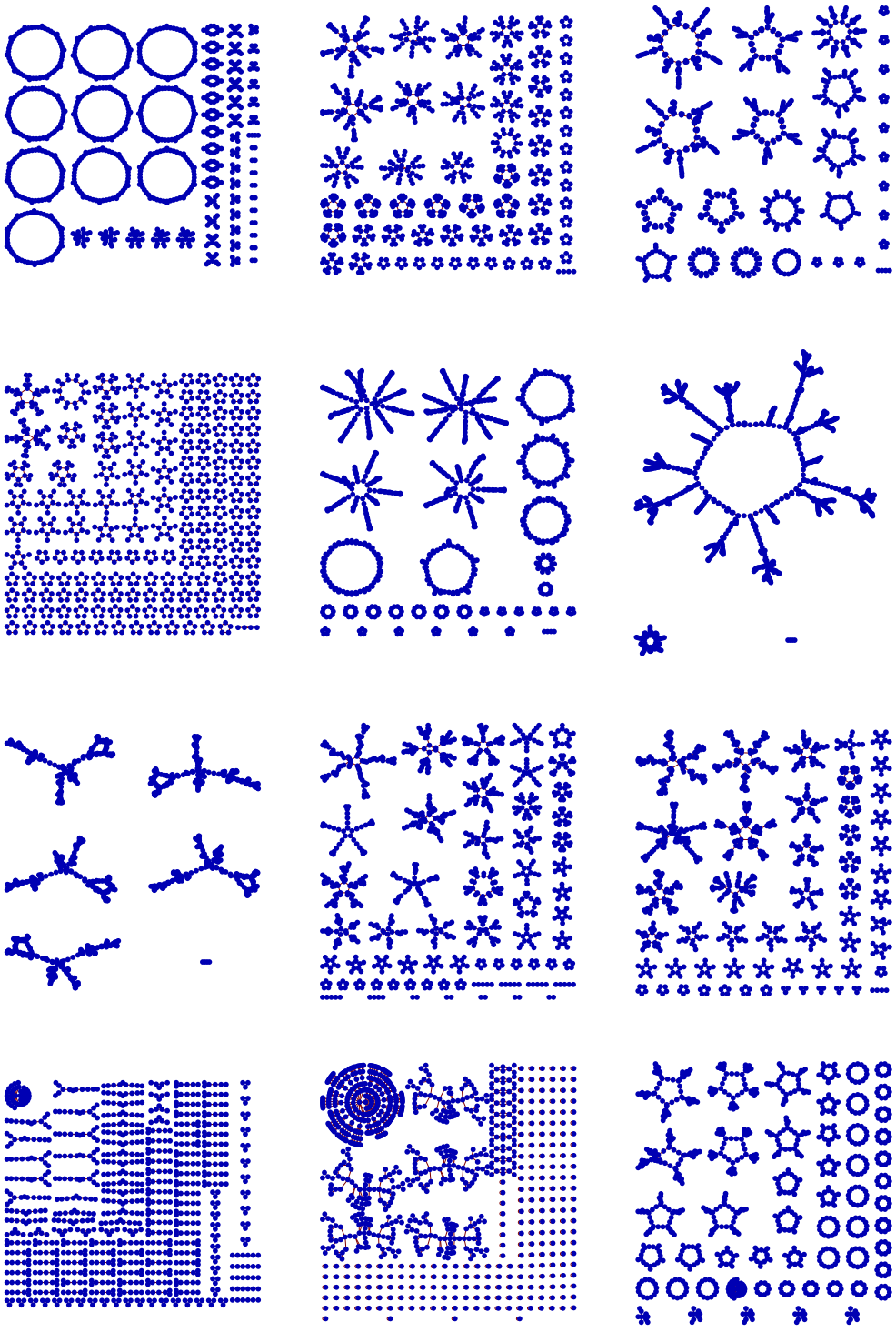


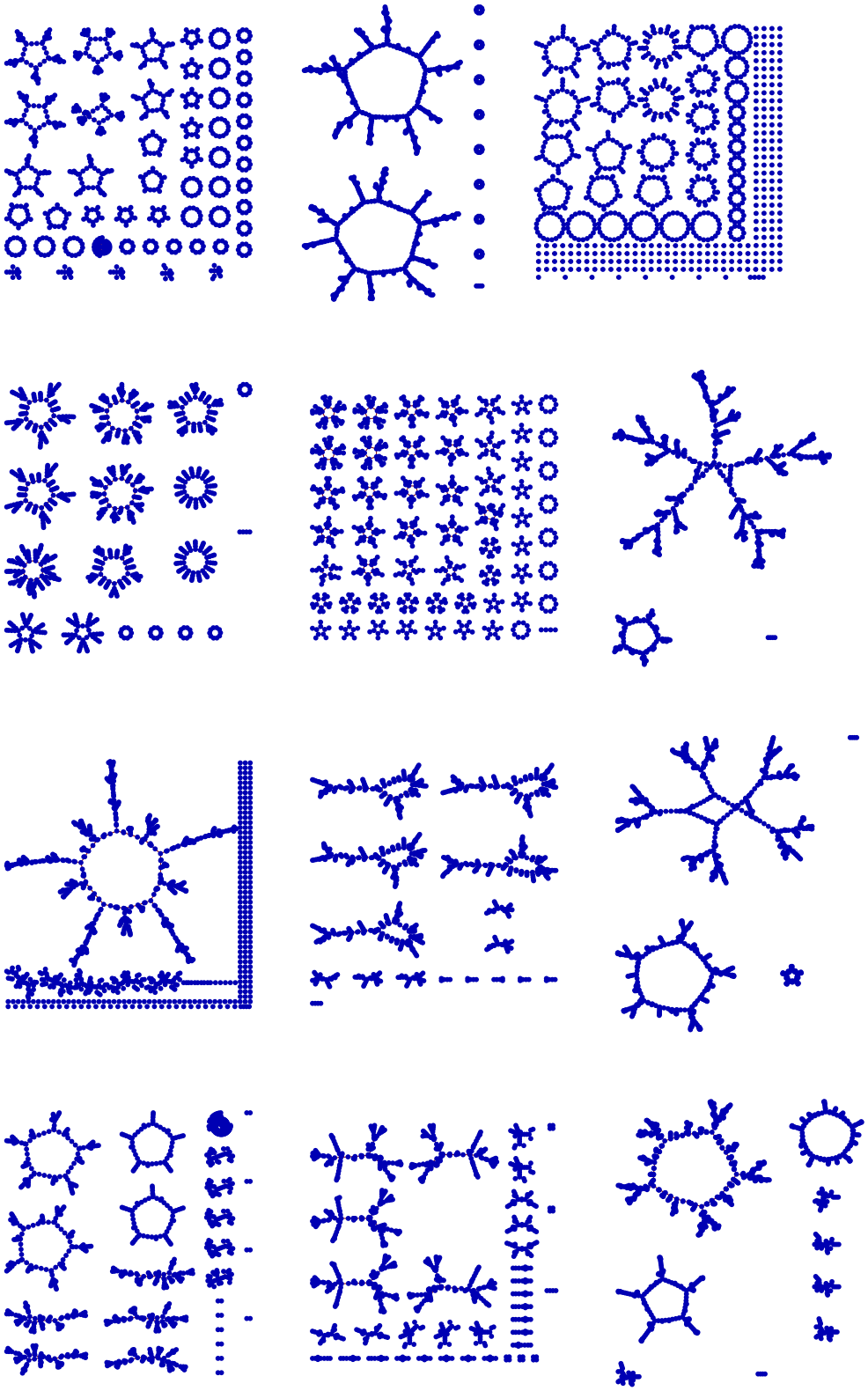


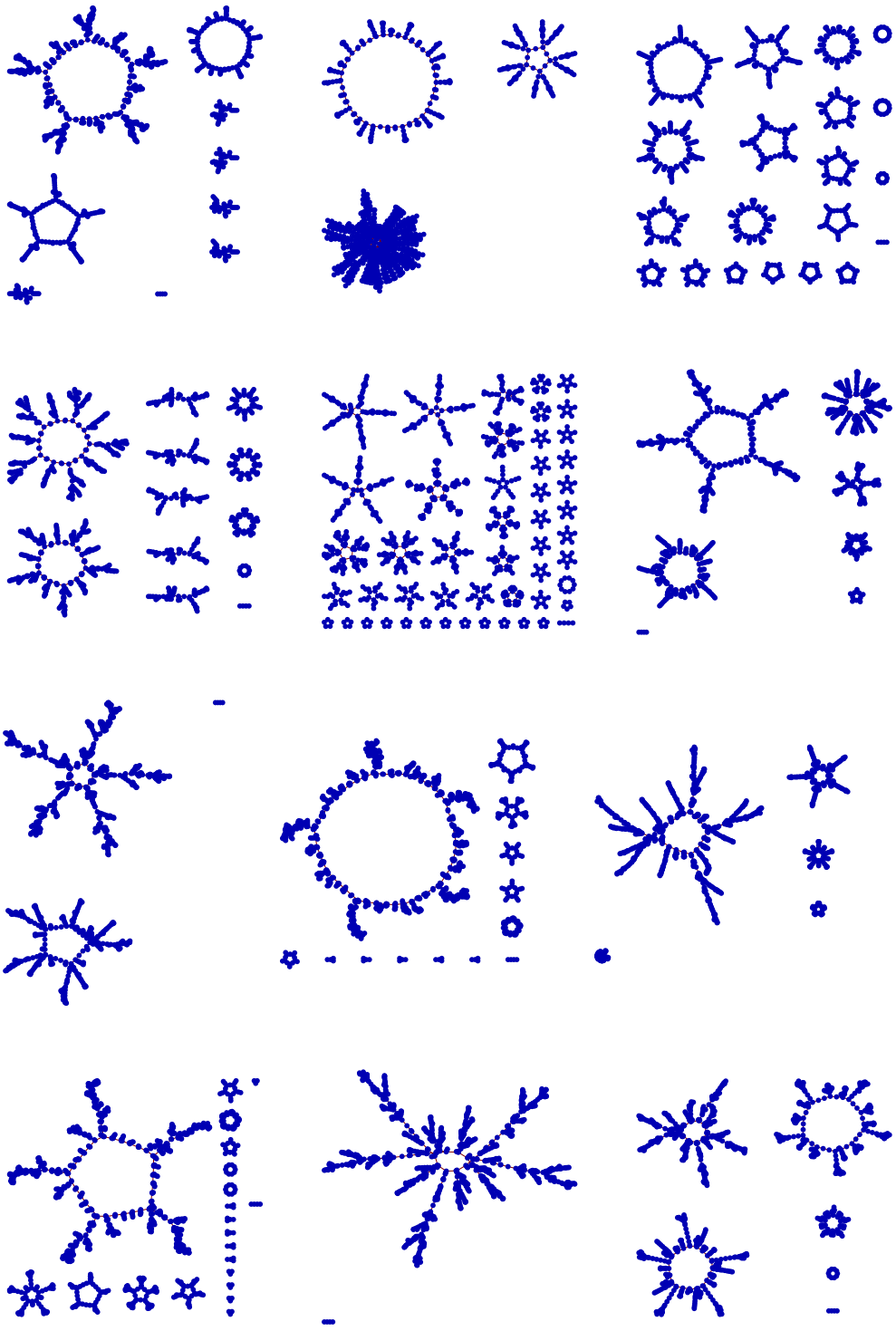


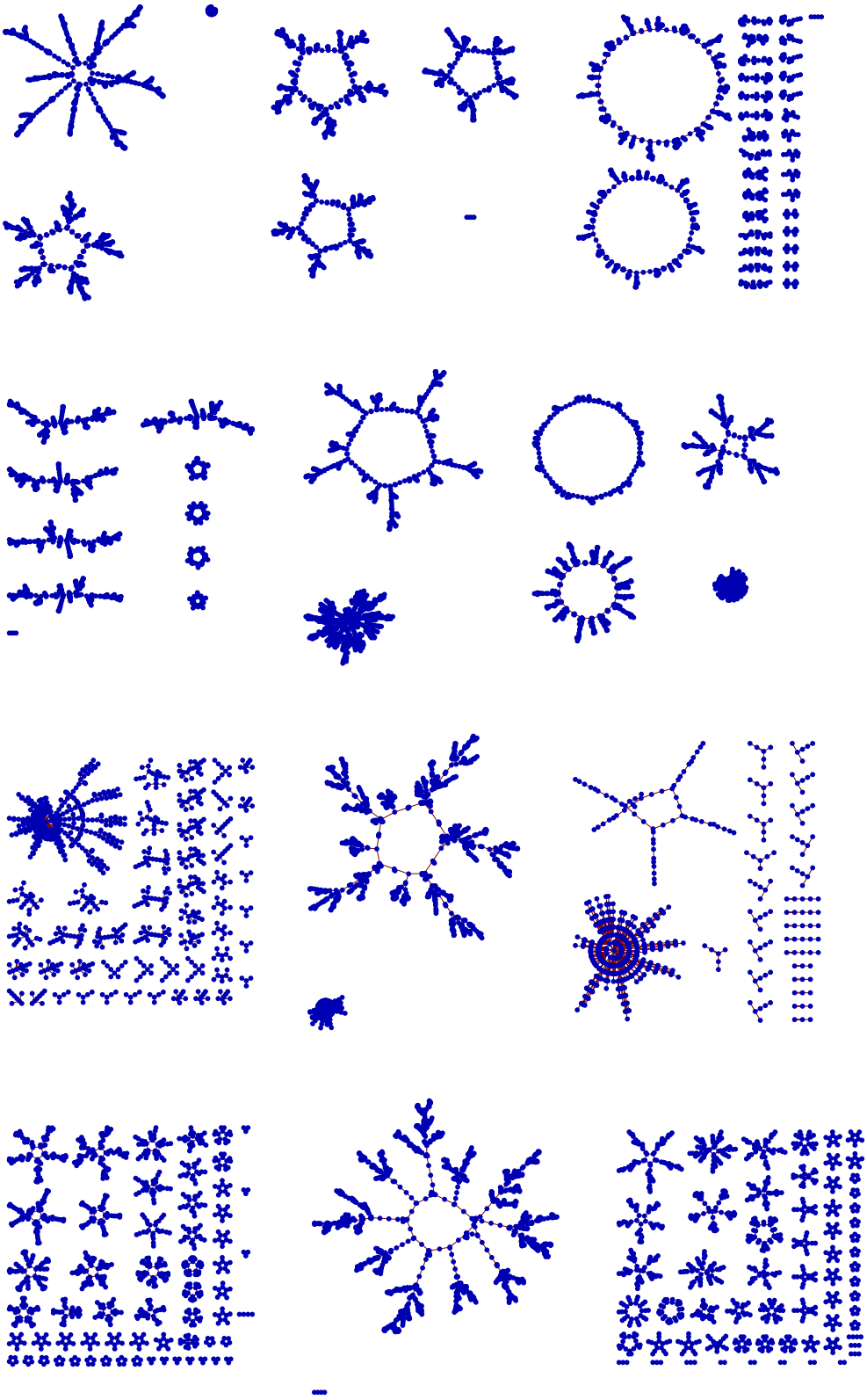


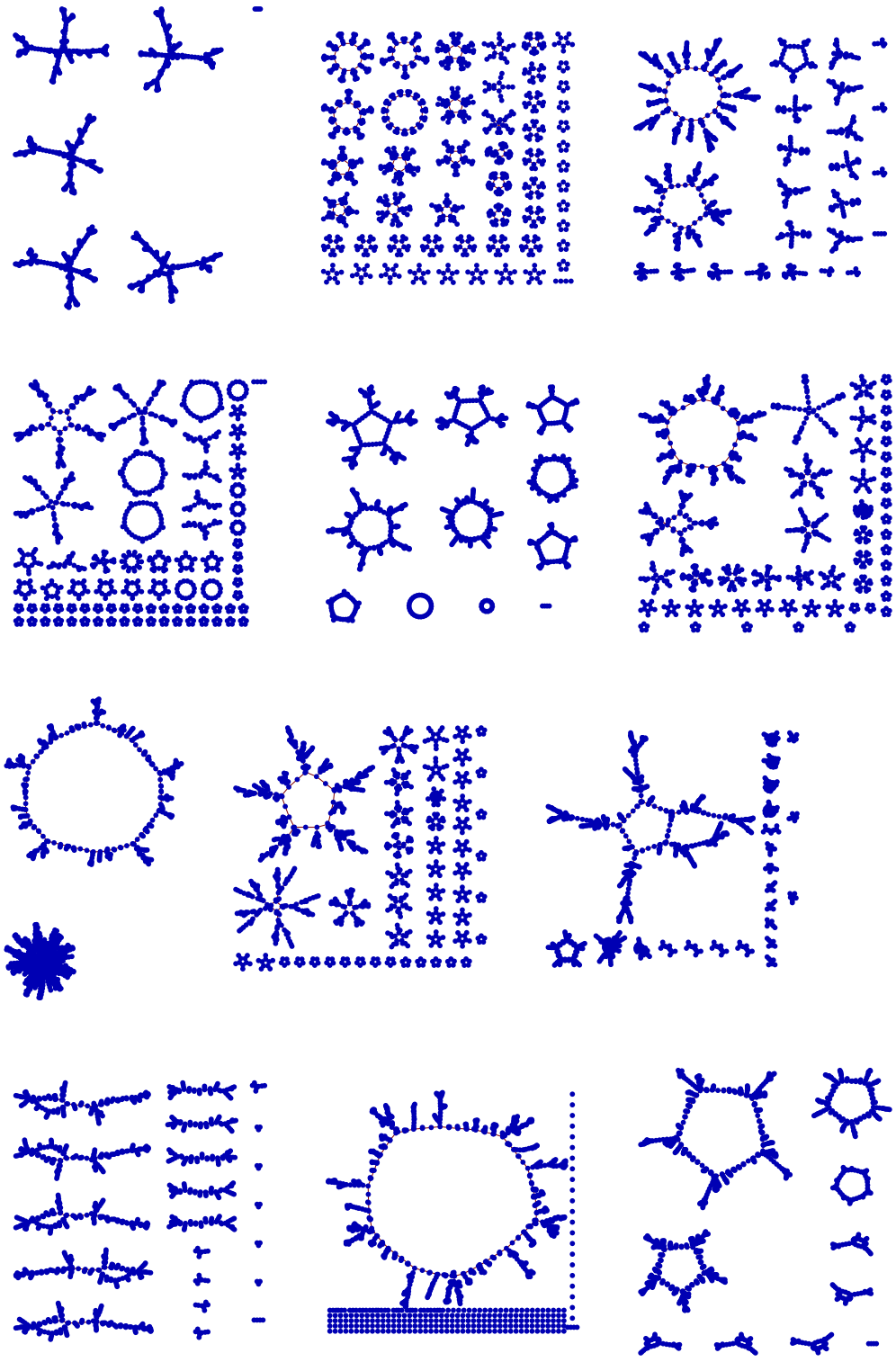


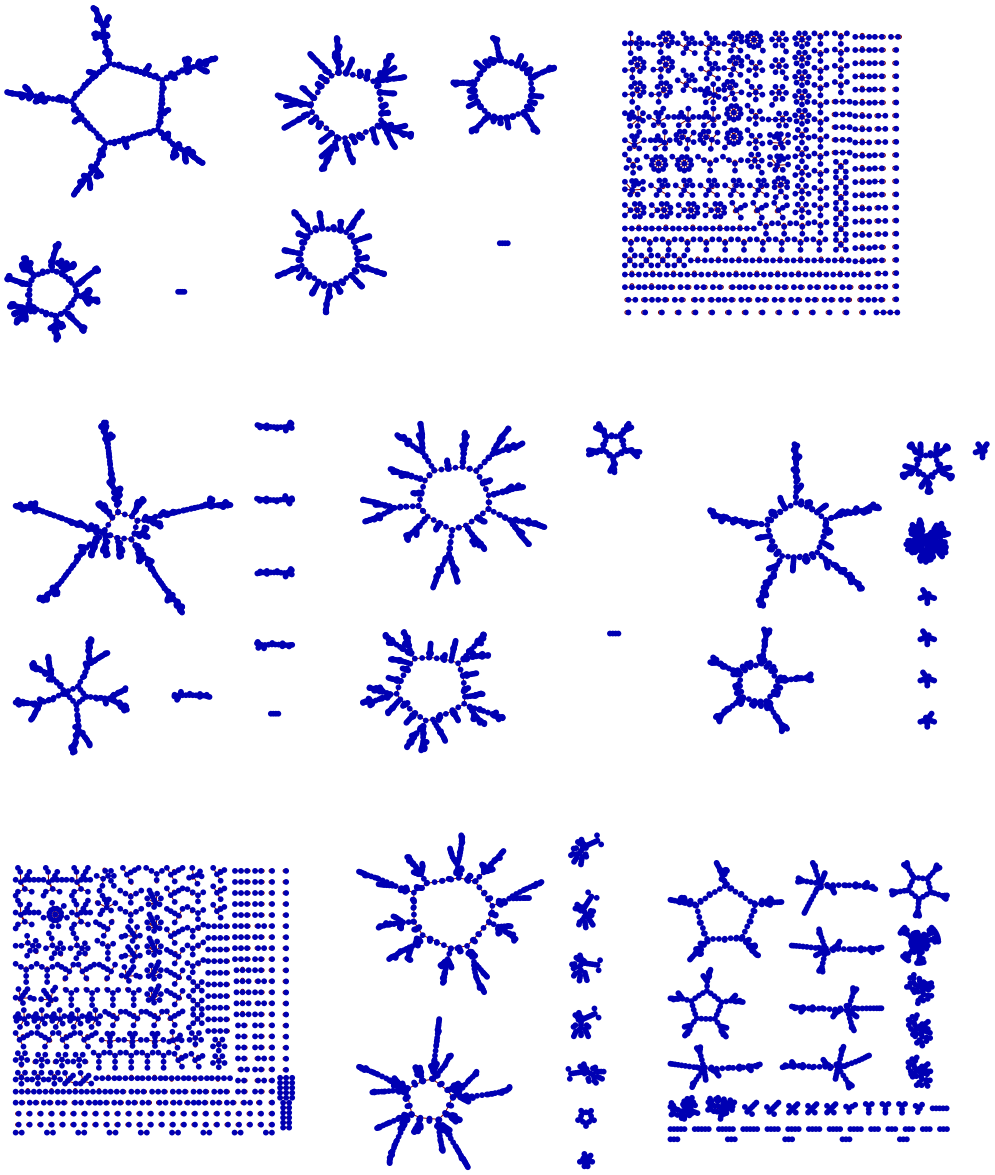












Butterflyoids?

Transition graph for ruleMNP[{1, 7, 12, 13, 10, 14, 15}], Tuples[{0,1,2},9]