# — vordenker-archive —

## Rudolf Kaehr

(1942-2016)

### Title

## Memristive Cellular Automata

First Steps towards a design for kenomic cellular automata

### Archive-Number / Categories

3_05 / K11, K12

### Publication Date

2011

### Keywords

TOPICS: Memristive Cellular Automata – Kenomic matrix and cellular patterns – Elementary 4-kenomic cellular automata rules – Elementary 5-kenomic cellular automata rules – Morphogrammatics as a theory of kenomic CA – Cellular automata are morphogrammatically incomplete – Composition of kenomic CAs

### Disciplines

Cybernetics, Computer Sciences, Artificial Intelligence and Robotics, Systems Architecture and Theory and Algorithms, Memristive Systems/Memristics

### Abstract

Memristive cellular automata are introduced as a new interpretation and model of general morphogrammatics. On the other hand, memristive cellular automata based on morphogrammatics are sheding some new light on the still little explored paradigm of morphogrammatic thinking as it was invented by the cybernetician Gotthard Gunther (1962) and elaborated by Kaehr/Mahler (1993). Understanding morphogrammatics as a system of kenomic cellular automata. Applying retrograde recursivity to kenomic cellular automata by the definition of the rules and by their applications. Hence, additional to the properties of CAs, i.e. locality, uniformity, syncronicity, the property of memristivity and polysemy shall be implemented. This is possible only by a transformation from a symbolic to a kenomic concept of CA.

### Citation Information / How to cite

**Rudolf Kaehr**: "Memristive Cellular Automata", www.vordenker.de (Sommer Edition, 2017) J. Paul (Ed.),
URL: http://www.vordenker.de/rk/rk_Memristive-Cellular-Automata_2011.pdf

### Categories of the RK-Archive

# Memristive Cellular Automata

*First steps towards a design for kenomic cellular automata*

**Rudolf Kaehr Dr.phil**

 ISSN 2041-4358

## Abstract

Memristive cellular automata are introduced as a new interpretation and model of general morphogrammatics. On the other hand, memristive cellular automata based on morphogrammatics are sheding some new light on the still little explored paradigm of morphogrammatic thinking as it was invented by the cybernetician Gotthard Gunther (1962) and elaborated by Kaehr/Mahler (1993).

Understanding morphogrammatics as a system of kenomic cellular automata. Applying retrograde recursivity to kenomic cellular automata by the definition of the rules and by their applications. Hence, additional to the properties of CAs, i.e. locality, uniformity, syncronicity, the property of *memristivity* and *polysemy* shall be implemented. This is possible only by a transformation from a symbolic to a kenomic concept of CA.

# 1. Calculating Spaces

"Rechnender Raum in denkender Leere" (SKIZZE-0.9.5)

„D´une certaine manière, ´la pensée‘ ne veut rien dire." Derrida

„Seine These, es gäbe weder die ´eine Wahrheit´ noch die ´eine Wirklichkeit‘, sondern das Universum sei vielmehr als ein ´bewegliches Gewebe´ aufeinander nicht zurückführbarer Einzelwelten zu denken, formulierte die entscheidende Aufgabe der Philosophie der Zukunft: eine Theorie bereitzustellen, die es gestattet, die Strukturgesetze des organischen Zusammenwirkens der je für sich organisierten Teilwelten aufzudecken.“ Gotthard Günther, 15. Juni 1980

Cellular Structured Space (Rechnender Raum)
Tom wrote:
> 1) Plankalkul
>
> :)-
Rechnender Raum. (Okay, that was cheap).
From: Eugene.Leitl@lrz.uni-muenchen.de
Date: Wed May 02 2001 - 15:24:32 PDT

http://www.thinkartlab.com/pkl/media/SKIZZE-0.9.5-Prop-book.pdf

# 2. Memristive Cellular Automata

## 2.1. Kenomic matrix and cellular patterns

"A cellular automaton is a collection of "colored" cells on a grid of specified shape that evolves through a number of discrete time steps according to a set of rules based on the states of neighboring cells. The rules are then applied iteratively for as many time steps as desired." http://mathworld.wolfram.com/CellularAutomaton.html

"The simplest class of one-dimensional cellular automata. Elementary cellular automata have two possible values for each cell (0 or 1), and rules that depend only on nearest neighbor values."

Weisstein, Eric W. "Elementary Cellular Automaton." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/ElementaryCellularAutomaton.html

Instead of the pre-defined {0, 1} values of each cell in a elementary cellular automaton, the value of the neighbor cells gets defined by the kenomic successor rules.

Therefore, the cellular automata rules for kenomic and morphogrammatic cellular automata are defined by the memristivity of retrograde recursivity. In a further step it will become obvious that the applications of the rules will change memristively according to their retrofgradeness. Hence, the ‘set’ of ‘beginning’ rules is defined memristively as well as the applica-

tions (iterability) of the rules. What is thus produced by kenomic automata are memristive domains, worlds, universes of kenomic computation.

Rules of functorial *interchangeability* are guiding the interactions of different cellular worlds enabled by kenomic cellular automata.
(Recall: SKISZZE-0.9.5)
http://www.thinkartlab.com/pkl/media/SKIZZE-0.9.5-medium.pdf

**Configurations: Transition in time**
*"The only reason for time is so that everything doesn't happen at once."*
— Albert Einstein
The exact reason for morphogrammatics and its memristivity is the fact that things happens at once, all the time.

"One further ingredient that is needed for our cellular lattice to evolve with discrete time steps is a local rule or local *transition function* governing how each cell alters its state from the present instant of time to the next based on a set of rules that take into account the cell's current state and the current state of its neighbors."

**Homogene and heterogene transitions**

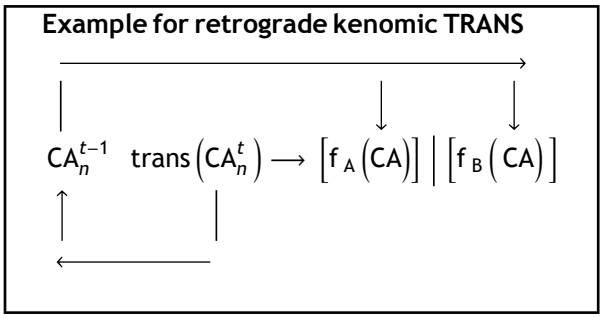
**A. Homogene**
One set of unambiguos rules are applied in time. Hence, a unambigous transition is defined.

$c_i(t+1) = \phi\,[c_{i-1}(t), c_i(t), c_{i+1}(t)]$ .

$c_i(t+1) =$
$\phi$: $[c_{i-1}(t), c_i(t), c_{i+1}(t)] \longrightarrow [c_{i-1}(t), c_i(t), c_{i+1}(t)]$
$\phi : \{\text{rules}\}$

$x_n^{t+1} = f\left(x_{n+1}^t, x_n^t, x_{n-1}^t\right)$

B. **Retrogradness of memristic transitions**




The transition 'trans' for a CA at the time t to a new configuration at t+1 is depending retrograde recursively on the constellation of the CA of t-1. Hence, $CA^{t+1}$= trans($CA^{t-1}$, $CA^t$).

C. **Heterogene**
Several rules are applied and defining multiple transitions.

This allows a choice of rules. Otherwise all possible transitions are holding.

$c_{i\,.j}(t + 1) = \phi_{i.j}\,([c_{i\,-1}(t),\, c_i(t),\, c_{i+1}\,(t)] \mid [c_j - 1(t),\, c_j(t),\, c_{j+1}\,(t)])$.

$$x_{nA}^{t+1} \,\Big|\, x_{nB}^{t+1} =$$

$$\text{trans}\left(x_{n+1}^{t},\, x_n^t,\, x_{n-1}^t\right) = \left(x_{n+1}^{t},\, x_n^t,\, x_{n-1}^t\right)_A \,\Big|\, \left(x_{n+1}^{t},\, x_n^t,\, x_{n-1}^t\right)_B.$$

### 2.1.1. Properties of CA and kenoCA

"Therefore the three fundamental features of a cellular automaton are:
**uniformity**: all cell states are updated by the same set of rules;
**synchronicity**: all cell states are updated simultaneously;
**locality**: the rules are local in nature."

The new property is **memristivity** of kenomic cellular automata.

**Locality versus retrogradness**
Locality of the rules is not meaning retrogradness of the applicability of the rules applied locally.

"A fundamental precept of cellular automata is that the local transition function determining the state of each individual cell at a particular time step should be based upon the state of those cells in its immediate neighborhood at the previous time step or even previous time steps.
Thus the rules are strictly *local* in nature and each cell becomes an information processing unit integrating the state of the cells around it and altering its own state in unison with all the others at the next time step in accordance with the stipulated rule."
http://www.texnology.com/joel.pdf

"That is, complex global features can **emerge** from the strictly local interaction of individual cells each of which is only aware of its immediate environment."

**Emergent** features are not related to retrogradness.

**First and second order automata**
"These elementary cellular automata are examples of first order automata in the sense that the state of a cell at time step t + 1 only depends on the state of its neighbors at the previous time step t. Whereas in a second order automaton a cell's state at time step t + 1 is rather more demanding and depends on the state of its neighbors at time steps t - 1 as well as t, analogous to the way the Fibonacci sequence was formed."

Retrogradness is functionally of **second order** but it is not defined by second order rules.

Hence, for the cell (i, j) = (a), the neighbor cells have the values (a) and (b). That is producing 6 kenomic patterns for {<a>, <b>} and not 8 distinct digital patterns for {0, 1}. The next steps of the rules are depending on their history which is not identical with an abstract continuation of the application of rules. Hence, the "resulting value" of the rule, define by the 2

"neighboring cells" is not abstractly defined by combinatorics of possible valuations but by the possibilities opened up by the predecessor states, i.e. by the history of the previous development.

Therefore, the combinatorics between digital and kenomic automata are not identical.
For a 3 cell grid with 2 states there are $2^3$= 8 digital constellations.
For a 3 cell grid (kenomic matrix)and 2 kenomic 'states' there are 4 = $\sum_{n=1}^{2} Sn(3, n)$. Hence, for 3 states of a 3 cell grid there are $\sum_{n=1}^{3} Sn(3, n)$ kenomic constellations.

For 8 binary states there are a total of $2^8$= 256 elementary cellular automata.

A visualization of elementary cellular automata is quite straightforward. There are no ambiguities and perplexities involved. Surprises are appearing on an application level but not on the level of the basic definitions of the rules and their graphic representations.

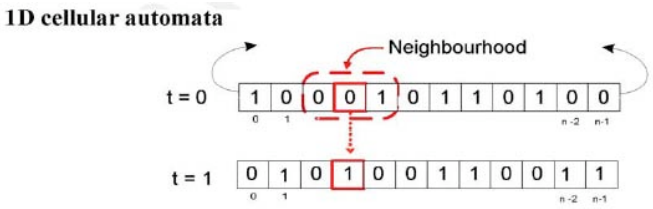
It is a principle of morphogrammatic thinking that ambiguity and perplexity comes first.

Morphograms are interpretatively ambiguous. A solution of a graphic representation of ambiguous cellular automata is not easily accessible.

**Context rules of applicability**
In contrast to the classical definitions of CAs, memristic CAs have to realize their antidromic recursivity on all levels of the construction and application, i.e. the context rules for antidromic repetitions have to be explicitly defined to rule or guide the applications of the elementary kenomic rules of kenomic CAs.

The combinatorics for cellular automata is exponential with the stable base 2 for two elements, i.e. $2^m$.
Kenomic developments are combinatorially defined by the Stirling numbers of the second kind, $\sum_{n=1}^{m} Sn(m, n)$.
Therefore, a kenomic definition of an "elementary cellular automaton" has to taken all 4 cells into account to determine the "resulting value" of the next generation. Otherwise, a value constellation for the 3 cells of (000) and (111) would have to be considered as kenomically identical. But then the constellations $\begin{bmatrix} 0 & 0 & 0 \\ - & 1 & - \end{bmatrix}$ and $\begin{bmatrix} 1 & 1 & 1 \\ - & 1 & - \end{bmatrix}$ which are different couldn't be produced.

**Cyclic applications**
This is nicely depicted in the paper:
Pascal Bouvry et al, Cellular automata computations and secret key cryptography
http://pascal.bouvry.org/ftp/parco04.pdf




**Rule of CA**

Neighbourhood radius r = 1, rule $01011010_2 = 90_{10}$

| Number | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Neighbourhood | 111 | 110 | 101 | 100 | 011 | 010 | 001 | 000 |
| Rule result | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |

Fig. 1. 1D cellular automata with neighborhood = 1.

**Ambiguity versus computational reduction**
Mirrored rules, which are the same as their mirrored rule are called **amphichiral** (64). Complementary rules are changing the roles of 0 and 1 in the rule definition.

Number sequences defined by elementay cellular automata: Jacobsthal, Pascal, etc.

Stirling numbers of the second kind are crucial for the architectonics of kenomic cllular automata.

**Wolfram Rule 30**

$$\begin{bmatrix} 0 & 0 & 0 \\ \square & 0 & \square \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \\ \square & 0 & \square \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ \square & 0 & \square \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 \\ \square & 1 & \square \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ \square & 1 & \square \end{bmatrix}, \begin{bmatrix} 1 & 0 & 1 \\ \square & 1 & \square \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ \square & 1 & \square \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ \square & 0 & \square \end{bmatrix}.$$

"For example, the table giving the evolution of rule 30 (30 = $00011110_2$) is illustrated above. In this diagram, the possible values of the three neighboring cells are shown in the top row of each panel, and the resulting value the central cell takes in the next generation is shown below in the center." (Weisstein, ibd.)

In contrast to the abstract combinatorial definition of the elementary cellular automata rules as a product of the the $2^3$=8 states of neighboring cells and the binary next generation states $2^8$=256 the kenomic cells and ther kenomic states are building a 'holistic' pattern. Therefore, the whole structure of the base of the elementary rules is changed.

Because the next generation state of the fourth cell is depending on the previous kenomic states, the number of the whole pattern is maximally $\sum_{n=1}^{4} Sn\,(4,\ n) = 1 + 7 + 6 + 1 = 15$.

## 2.2. Elementary 4-kenomic cellular automata rules

### 2.2.1. Dyadic and kenomic CA scheme

Following the standard definitions for dyadic CA presented in a systematic way by Jaime Rangel-Mondragón:

rule1 = {0, 1, 0, 1, 1, 1, 1, 1};

We interpret this rule as describing a
transformation from the set $\{0, 1\}^3$ to the set $\{0, 1\}$.

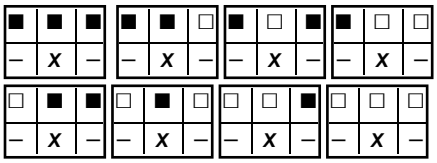
```
Thread[Map[StringJoin[ToString /@ IntegerDigits[#, 2, 3]] &, Range[0, 7]] -> rule1]
```

```
{000 -> 0, 001 -> 1, 010 -> 0, 011 -> 1, 100 -> 1, 101 -> 1, 110 -> 1, 111 -> 1}
```

That is, a rule of the form axb -> y describes the new value y of a given cell, given its present value x and those of its two neighbors a and b. In the case of dyadic CA, there are 2^2^3=256 possible rules. It is possible to prove that linear CA do not need to have large neighborhoods; [...]."
http://library.wolfram.com/infocenter/MathSource/505/CAcatalog.nb

**D yadic CA scheme**

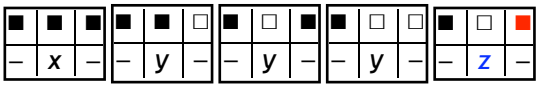
CA = $2^3$ = 8 positions and $2^8$ = 256 constellations with x = {■, □}.



Null

**Kenomic CA scheme**

$$\sum_{n=1}^{3} \mathrm{Sn}(3, n) = 1 + 3 + 1 = 5$$



This representation of the kenomic scheme is conventional. Every other representation which fulfils the epsilon/nu-structure of the patterns is accepted.

**Epsilon/Nu-structure of morphograms**

How to define the elements of the rule schemata?
Obviously, the elements are not considered as semiotic or syntactic entities and are therefore not primarily determined by their atomic identity.

The two sign sequences (aba) and (bab) are seen as structurally, i.e. kenomically equivalent. Instead of using abstractions to form equivalence classes of signs the simple method of relational equality and non-equality of pairs of signs shall be used. This defines the epsilon/nu-structure of morphograms,(epsilon=equal, nu=non-equal).

```
(aba)   (bab)
≠ ≠     ≠ ≠
 =       =
```

Hence both tuples are structurally equivalent because their relations are equal. This allows to give a precis definition of the transition rules for Cas.

For reasons of convenience and aesthetics the relational symbols shall be replaced by the usual set of marks {■, □, ■}.

$(aaa) \Longrightarrow (===) \Longrightarrow [■■■].$

$(aab) \Longrightarrow (= \neq \neq) \Longrightarrow [■■□].$

$(aba) \Longrightarrow (\neq \neq =) \Longrightarrow [■□■].$

$(abb) \Longrightarrow (\neq = \neq) \Longrightarrow [■□□].$

$(abc) \Longrightarrow (\neq \neq \neq) \Longrightarrow [■□■].$

**Transitions**

$(aba) \longrightarrow (a) : (aaa)$

$(bab) \longrightarrow (bbb)$

$(aba) \longrightarrow (aaa) \Longrightarrow [(\neq \neq =) \longrightarrow (===)] \ : \ [■■■].$

**Number of realizations**

$x = \{a, b\}, \ y = \{a, b, c\}, \ z = \{a, b, c, d\},$

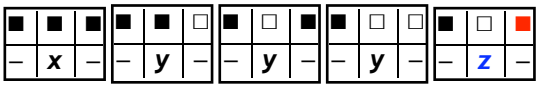
$m = 2, \ x*y*y*y*y = 2^5 = 32$

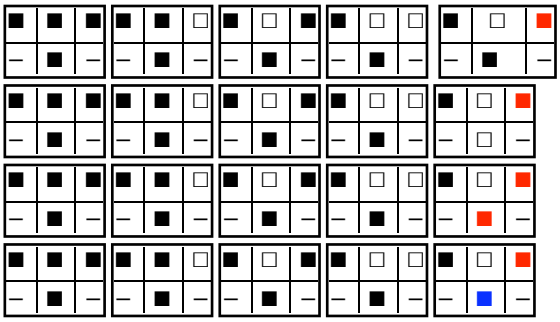
$m = 3, \ x*y*y*y*y = 2*3^4 = 162$

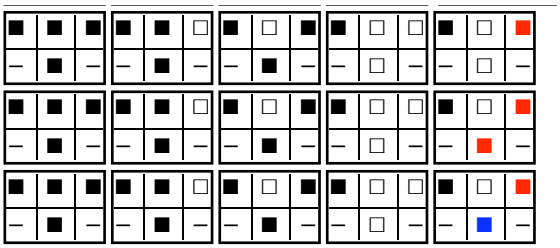
$m = 4, \ x*y*y*y*z = 2*3*3*3*4 = 216$

$[xyyyz] =$

aaaaa, aaaab, aaaac, aaaad,
aaaba, aaabb, aaabc, aaabd,
aabba, aabbb, aabbc, aabbd,
abbaa, abbab, abbac, abbad,
abbba, abbbb, abbbc, abbbd,
and so on!





**and so on!!!**

**Reduction**

Kenomic reduction of the d : $2^n,\ n = 3$ :

$$\{0,\ 1\}^3 = 8 \Longrightarrow \sum_{n=1}^{3} \mathrm{Sn}(3,\ n) = 1 + 3 + 1 = 5$$

$$\sum_{n=1}^{4} \mathrm{Sn}(4,\ n) = 1 + 7 + 6 + 1 = 15$$

## 2.2.2. Rules

| c2 | c1 | c3 |
|---|---|---|
| – | c4 | – |

$$\Longrightarrow (c1, (c2,\ c3), c4),$$

$$(c1, (c2, c3)) \Longrightarrow (c1, (c2, c3), c4) :$$
$$(a,\ (a,\ a)) \Longrightarrow (a, (a,\ a), a),\ (a, (a, a), b)$$
$$(a,\ (a, b)) \Longrightarrow (a, (a, b), a),\ (a, (a, b), b),\ (a, (a, b), c)$$
$$(a,\ (b,\ a)) \Longrightarrow (a, (b, a), a),\ (a, (b, a), b),\ (a, (b, a), c)$$
$$(a,\ (b,\ b)) \Longrightarrow (a, (b, b), a),\ (a, (b, b), b),\ (a, (b, b), c)$$
$$(a,\ (b,\ c)) \Longrightarrow (a, (b, c), a),\ (a, (b, c), b),\ (a, (b, c), c),\ (a, (b, c), d).$$

This construction of the rule-schemata is using the *retrograd recursivity* of the continuation operation for morphograms. Therefore, retrogradness as a memristive property is implemented at the very beginning of kenomic cellular automata. In other words, th rules of kenomic cellular automata are memristive. This holds for all further extensions of the memristive construction to higher order and more complex kenomic cellular automata.
http://memristors.memristics.com/MorphoReflection/Morphogrammatics%20of%20Reflection.html

**Rules**

1. $(a,\ a,\ a) \longrightarrow \begin{pmatrix} a \\ b \end{pmatrix}$, that is : $(a,\ a,\ a)$ or $(a,\ b,\ a)$.

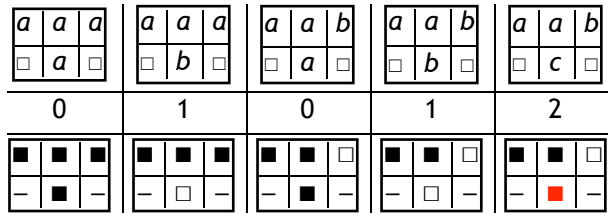
Kenomic rules are build in analogy to the CA rules. Thus the *"resulting value the central cell takes in the next generation is shown in the center"*.
1. (a, a, a) ⟶ (a, a, a), (a, b, a).
In fact, the rule might also be interpreted not as disjunctive but as a simultaneity of both: (a, a, a) | (a, b, a).

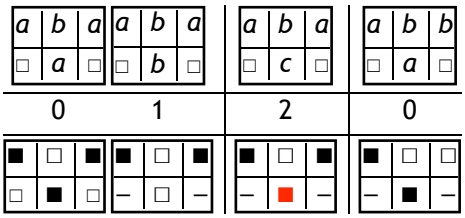
Thus $(a,\ a,\ a) \longrightarrow \begin{pmatrix} a \\ b \end{pmatrix}$, hence $\begin{pmatrix} (a,\ a,\ a) \\ \amalg \\ (a,\ b,\ a) \end{pmatrix}$

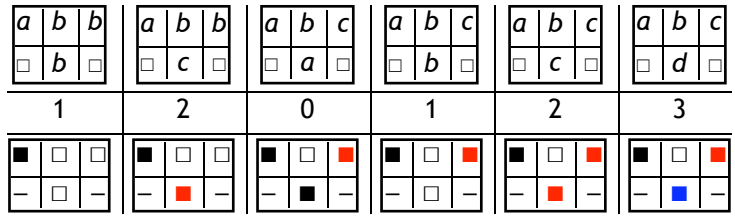
2. $(a,\ a,\ b) \longrightarrow \begin{pmatrix} a \\ b \\ c \end{pmatrix}$ 3. $(a,\ b,\ a) \longrightarrow \begin{pmatrix} a \\ b \\ c \end{pmatrix}$

4. $(a,\ b,\ b) \longrightarrow \begin{pmatrix} a \\ b \\ c \end{pmatrix}$ 5. $(a,\ b,\ c) \longrightarrow \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}$.

For a≡ ■, b≡□, c≡ ■ and d ≡ ■ the following symbolic and graphical representation is depicted.

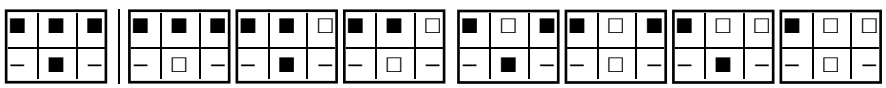


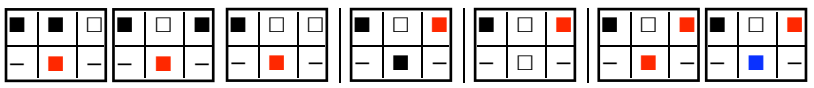




.

**2 elements**

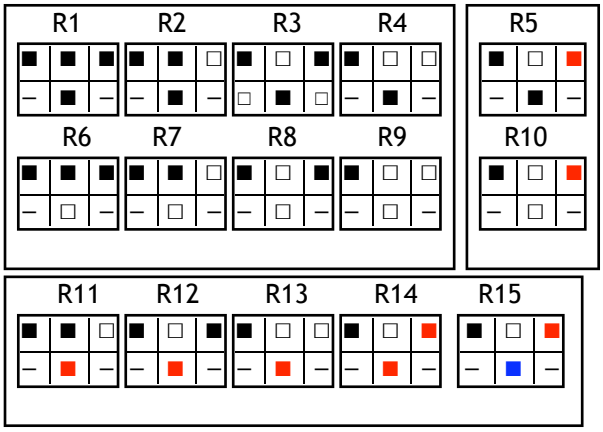


**3 elements** **4 elements**



.

**System of elementary kenomic cellular rules**




## 2.2.3. Examples

**Wolfram**
CellularAutomaton[150, {1, 0, 1, 1}, 3]
{{1, 0, 1, 1},
{0, 0, 0, 1},
{1, 0, 1, 1},
{0, 0, 0, 1}}

**Kenomic model for rule 150**
kenom([150], {1,0,1,1}, 3]):
[[150] = r1, r7, r8, r9], [dual[150] = r6, r2, r3, r4]

| CA | wolfram | keno | | | dual | keno | | |
|---|---|---|---|---|---|---|---|---|
| 1011 | r5, 3, 5, 1 | r9 .8 | .9 | .1 | 0100 | r4 .3 | .4 | .6 |
| 0001 | r7, 8, 7, 6 | r7 .1 | .7 | .8. | 1110 | r2, 6, 2, 4 | | |
| 1011 | r5, 3, 5, 1 | r1 .8 | .9 | .1 | 1011 | r4, 3, 4!, 6 | | |
| 0001 | n = 3, stop | | | | 1110 | n = 3, stop | | |

**Dual kenomic rules**
r1, r6,
r2, r7; r11
r3,r8, r12
r4, r9; r13
r5, r10; r14, r15

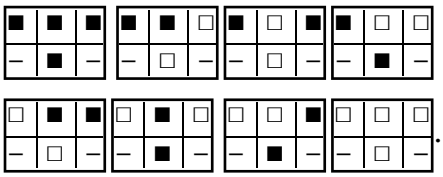
**Strict Combinations**

| | |
|---|---|
| type1 = r1.2.3.4.5/10 | dual-type1 = r6.7.8.9.10/5 |
| type2 = r1.7.8.9.5/10 | dual-type2 = r6.2.3.4.10/5 |
| type3 = r1.2.8.9. 5./10 | dual-type3 = r6.7.3.4.10/5 |
| type4 = r1.2.3.9.5/10 | dual-type4 = r6.7.8.4.10/5 |

**Combinations with ambiguity or decision space of application**
type3 = r1.2.**3.8**.9.**10.14.15**

### 2.2.4. Keno analogy of rule 150

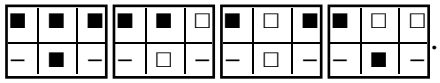
Symbolic CA rule 150 = $\{10010110\}$



| Nr.\ l | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | rule = $[150; \{10010110\}]$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | □ | □ | □ | □ | ■ | □ | □ | □ | □ | 5, 7, 4 |
| 2 | □ | □ | □ | ■ | ■ | ■ | □ | □ | □ | 7, 5, 1, 2, 4 |
| 3 | □ | □ | ■ | □ | ■ | □ | ■ | □ | □ | 7, 6, 3, 6, 3, 6, 4 |
| 4 | □ | ■ | ■ | □ | ■ | □ | ■ | ■ | □ | □ |
| 5 | □ | □ | □ | □ | □ | □ | □ | □ | □ | stop |

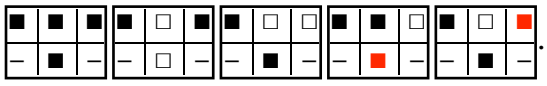
**Kenomic CA rules**

keno $[150]$ = r1 .7 .8 .4 = $[1001]$



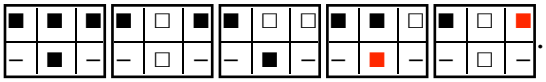
| Nr.\l | 1 | 2 | 3 | 4 | **5** | 6 | 7 | 8 | 9 | rule = $[[150]; 1, 7, 8, 4]$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | □ | □ | □ | □ | ■ | □ | □ | □ | □ | 8, 7, 8, 4 |
| 2 | □ | □ | □ | □ | □ | ■ | □ | □ | □ | 1, 7, 8, 4 |
| 3 | □ | □ | □ | ■ | □ | □ | ■ | □ | □ | 7, 8, 4, 7, 8, 4 |
| 4 | □ | □ | ■ | □ | ■ | □ | □ | ■ | □ | 7, 8, 8, 8, 4, 7, 8, 4 |
| 5 | □ | □ | □ | □ | □ | ■ | □ | □ | ■ | 1, 1, 1, 7, 8, 4, 7, 8 |
| 6 | □ | ■ | ■ | ■ | □ | □ | ■ | □ | □ | 7, 4, 1, 7, 4, 7, 8, 4, 1 |
| 7 | □ | ■ | ■ | □ | ■ | □ | □ | ■ | ■ | □ |
| 8 | □ | □ | □ | □ | □ | □ | □ | □ | □ | stop |

keno $[150+]$ = r1 .8 .4 .11 .5 = $[10121]$



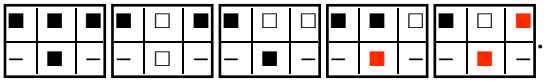
| Nr.\l | 1 | 2 | 3 | 4 | **5** | 6 | 7 | 8 | 9 | rule = $[[150^{+}]; 1, x, 8, 4, 11, 5]$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | □ | □ | □ | □ | ■ | □ | □ | □ | □ | 11, 8, 4 |
| 2 | □ | □ | □ | ■ | ■ | ■ | □ | □ | □ | 11, 5, 4, 11, 4 |
| 3 | □ | □ | ■ | ■ | ■ | ■ | ■ | □ | □ | 5, 8, 11, 8, 5, 8 |
| 4 | □ | | ■ | ■ | ■ | □ | ■ | ■ | □ | 4, 4, 11, 5, 5, 4, 4 |
| 5 | □ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | 11, 4, 11, 8, 4, 1, 11, 5 |
| 6 | ■ | ■ | ■ | □ | ■ | ■ | ■ | □ | ■ | 11, 8, 8, 11, 4, 11, 8, 5, 8 |
| 7 | ■ | □ | □ | ■ | ■ | ■ | □ | ■ | □ | 8, 4, 11, 4, 11, 5, 5, 8, |
| 8 | □ | ■ | ■ | ■ | ■ | ■ | ■ | □ | ■ | □ |
| 9 | □ | □ | □ | □ | □ | □ | □ | □ | □ | stop |

**keno [150 =]** = r1 .8 .4 .11 .10 = [10120]



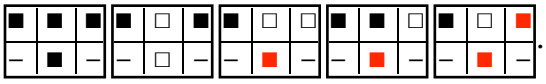
| Nr.\l | 1 | 2 | 3 | 4 | **5** | 6 | 7 | 8 | 9 | rule = $[[150^{=}]$; r1 .8 .4 .11 .10] |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | □ | □ | □ | □ | ■ | □ | □ | □ | □ | 11, 8, 4 |
| 2 | □ | □ | □ | ■ | □ | ■ | □ | □ | □ | 10, 8, 10, 10 |
| 3 | □ | □ | □ | □ | □ | □ | □ | □ | □ | 1, 1, 1, 1, 1, 1, 1 |
| 4 | □ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | □ | 8, 4, 1, 1, 1, 1, 1, 1, 11 |
| 5 | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | □ |
| 6 | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | □ | □ |
| 7 | ■ | ■ | ■ | ■ | ■ | ■ | ■ | □ | ■ | □ |
| 8 | ■ | ■ | ■ | ■ | ■ | ■ | □ | □ | □ | □ |
| 9 | □ | □ | □ | □ | □ | □ | □ | □ | □ | stop |

**keno [150 =]** = r1 .8 .4 .11 .14 = [10122]



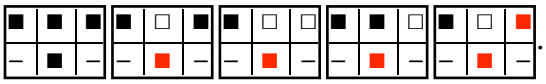
| Nr.\ $l$ | 1 | 2 | 3 | 4 | **5** | 6 | 7 | 8 | 9 | rule = [[150 =]; r1 .8 .4 .11 .14] |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | □ | □ | □ | □ | ■ | □ | □ | □ | □ | 11, 8, 4 |
| 2 | □ | □ | □ | ■ | □ | ■ | □ | □ | □ | 11, 8, 14, 8, 4 |
| 3 | □ | □ | ■ | □ | ■ | □ | ■ | □ | □ | 11, 8, 8, 8, 8, 4 |
| 4 | □ | ■ | □ | □ | □ | ■ | □ | ■ | □ | 11, 8, 4, 1, 11, 8, 14, 8, |
| 5 | ■ | □ | ■ | ■ | ■ | □ | ■ | □ | ■ | □ |
| 6 | □ | ■ | ■ | ■ | ■ | □ | □ | ■ | □ | □ |
| 7 | ■ | ■ | □ | ■ | ■ | ■ | ■ | □ | ■ | □ |
| 8 | □ | □ | □ | □ | □ | □ | □ | □ | □ | stop |

**keno [150 =]** = r1 .8 .13 .11 .14 = [10222]



| Nr.\ $l$ | 1 | 2 | 3 | 4 | **5** | 6 | 7 | 8 | 9 | rule = [[150 =]; r1 .8 .13 .11 .14] |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | □ | □ | □ | □ | ■ | □ | □ | □ | □ | 11, 8, 13 |
| 2 | □ | □ | □ | ■ | □ | ■ | □ | □ | □ | 11, 8, 8, 8, 13 |
| 3 | □ | □ | ■ | □ | □ | □ | ■ | □ | □ | 8, 8, 1, 14, 8, 1 |
| 4 | □ | □ | □ | ■ | ■ | ■ | □ | ■ | ■ | 1, 1, 14, 8, 14, 8, 14, 14 |
| 5 | ■ | ■ | □ | ■ | □ | ■ | □ | ■ | ■ | 13, 11, 14, 8, 8, 8, 13, 11 |
| 6 | ■ | ■ | ■ | □ | □ | □ | □ | ■ | ■ | 13, 1, 11, 13, 1, 1, 11, 13, 11 |
| 7 | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | □ |
| 8 | □ | □ | □ | □ | □ | □ | □ | □ | □ | stop |

**keno [150 =]** = r1 .12 .13 .11 .14 = [12222]



| Nr.\ $l$ | 1 | 2 | 3 | 4 | **5** | 6 | 7 | 8 | 9 | rule = [[150 =]; r1 .12 .13 .1 .14] |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | □ | □ | □ | □ | ■ | □ | □ | □ | □ | 11, 12, 13 |
| 2 | □ | □ | □ | ■ | ■ | ■ | □ | □ | □ | 11, 13, 1, 11, 11, |
| 3 | □ | □ | ■ | ■ | ■ | ■ | ■ | □ | □ | 11, 13, 1, 1, 1, 11, 13, 13 |
| 4 | □ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | □ | □ |
| 5 | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | □ |
| 6 | □ | □ | □ | □ | □ | □ | □ | □ | □ | stop |

Inside the application :

**(8, 1)** $=_{KG}$ **(8, 2) : [** □ ■ □] ⟷ [■ □ ■] : rule8

## 2.2.5. Representation of rules

**Logical representation of symbolic rules**
CellularAutomaton[30,init,t] : 00011110
Mod[p + q + r + q r, 2]
$(p \veebar q \veebar r)$
○●●
128: and(p q r)
252: or(pq)  -- non(or(pq)): 3
60: non(and(pq))

**Morphogrammatic representation of kenomic rules**
Elementary kenomic cellular rules are not represented by logical functions but by morphograms.
But each logical function can be represented by morphograms.

LOG([252, 3]) = MG[8] .

http://atlas.wolfram.com/01/01/30/

## 2.3. Elementary 5-kenomic cellular automata rules

| | | |
|---|---|---|
| – | c5 | – |
| c2 | c1 | c3 |
| – | c4 | – |

$\Longrightarrow \big(\big(c1, \big(c2,\ c3\big), c4\big),\ c5\big),$

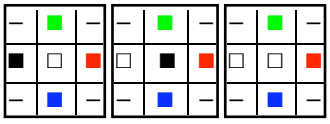
$$\sum_{n=1}^{5} Sn\big(n,\ 5\big) = 1 + 15 + 25 + 10 + 1 = 52$$

$$\big(c1, \big(c2, c3\big)\big) \Longrightarrow \big(c1, \big(c2, c3\big), c4\big) \Longrightarrow \big(\big(c1, \big(c2,\ c3\big), c4\big),\ c5\big):$$

$$\begin{aligned}
&\big(c1, \big(c2, c3\big)\big) \Longrightarrow \big(c1, \big(c2, c3\big), c4\big):\\
&\big(a,\ \big(a,\ a\big)\big) \Longrightarrow \big(a, \big(a,\ a\big), a\big),\ \ \big(a, \big(a, a\big), b\big)\\
&\big(a,\ \big(a, b\big)\big) \Longrightarrow \big(a, \big(a, b\big), a\big),\ \ \big(a, \big(a, b\big), b\big),\ \ \big(a, \big(a, b\big), c\big)\\
&\big(a,\ \big(b,\ a\big)\big) \Longrightarrow \big(a, \big(b, a\big), a\big),\ \ \big(a, \big(b, a\big), b\big),\ \ \big(a, \big(b, a\big), c\big)\\
&\big(a,\ \big(b,\ b\big)\big) \Longrightarrow \big(a, \big(b, b\big), a\big),\ \ \big(a, \big(b, b\big), b\big),\ \ \big(a, \big(b, b\big), c\big)\\
&\big(a,\ \big(b,\ c\big)\big) \Longrightarrow \big(a, \big(b, c\big), a\big),\ \ \big(a, \big(b, c\big), b\big),\ \ \big(a, \big(b, c\big), c\big),\ \big(a, \big(b, c\big), d\big).
\end{aligned}$$

$$\begin{aligned}
&\big(c1, \big(c2, c3\big), c4\big) \Longrightarrow \big(\big(c1, \big(c2,\ c3\big), c4\big),\ c5\big):\\
&\big(a, \big(a,\ a\big), a\big) \Longrightarrow\ \big(\big(a, \big(a,\ a\big), a\big)a\big), \big(\big(a, \big(a,\ a\big), a\big)b\big)\\
&\big(a, \big(a, a\big), b\big) \Longrightarrow\ \big(\big(a, \big(a,\ a\big), b\big)a\big), \big(\big(a, \big(a,\ a\big), b\big)b\big), \big(\big(a, \big(a,\ a\big), b\big)c\big)
\end{aligned}$$

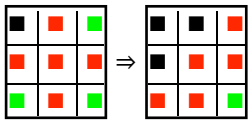
and so on!



# 3. Morphogrammatics as a theory of kenomic CA

Morphogrammatics was well formalized and implemented as a theory of form and its transformations. One specific transformation is produced by the so called “reflector”. A reflector is reversing the order of a basic morphogram. The results of such reflections are obviously very simple but elementary. But morphogrammatics is studying the behavior of complex compounds of basic morphograms and their transitions.

It might be of interest to transform the results of reflectional morphogrammatics into the framework of memristive cellular automata.
Hence, there shall be a transition from a patter [10, 2,10] to a pattern [2, 2, 11] of morphogramatics on the base of kenomic cellular automata transformations.



The morphogrammatic abstraction helps to hold the numbers low in the realm of the Stirling numbers of the seconsd kind.

There are just 3281 = $\sum_{n=1}^{9} Sn(9, n)$ different patterns which are defining 3-compound kenomic cellular atomata.

"The simplest neighborhood is an elementary system consisting of a one-dimensional row of cells, each of which can contain the value 0 or 1 (depicted as two colors), with a local neighborhood of size 3 (range or radius of 1).
More complex CA can be defined on two- or higher-dimensional arrays with multicolored cells and larger ranges. Each rule
is represented as an array of cells.
For the case of a local neighborhood of size 3, each triplet determines a single output cell in an array. A triplet with binary values can have eight possible patterns from 111 to 000. A local neighborhood of size 3 thus can generate 256 possible rules.

The formula for calculating the rule size space in a one-dimensional system is $k^{k^{(2r+1)}}$, where k represents the color possibilities for each state and r is the range or radius of the neighborhood. It is interesting to note that merely increasing r from 1 to 2 and maintaining the colors at two increases the rule space from 256 to 4.3 billion."
http://www.wolframscience.com/conference/2006/presentations/materials/speller-complex_systems-17-1-2.pdf

# 4. Cellular automata are morphogrammatically incomplete

## 4.1. Reduction and representation

Symbolic CA are representable by keno CA.

## 4.2. Symbolic CAs are morphogrammatically incomplete

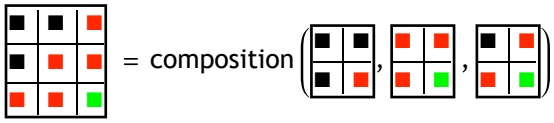
The morphogrammatic base of symbolic CAs are the 8 basic morphograms with 2 kenograms.
It is shown that a pattern of 4 places gives space for morphograms with up to 4 kenograms.
Hence, the morphogrammatic base of kenoCA are 15 morphic patterns and not just 8 like for symbolic CAs.
In this sense, the elementary rules of symbolic CA are incomplete in respect of their kenomic deep-structure.

## 4.3. Compounds of kenoCAs

### 4.3.1. Composition of kenomic CAs

Interaction between basic kenoCAs is established with a mediation of the 15 basic kenoCAs to compound kenoCAs.
The crucial question where are the new 'values' coming from that appeared for kenomic cellular automata with complexity m=3 and n=2 has an answer in the theory of *compound* kenomic automata.

**Composition**




| M | 1 | 2 | 3 |
|---|---|---|---|
| 1 | ■ | ■ | ■ |
| 2 | x | □ | □ |
| 3 | x | x | ■ |

$\Longrightarrow$

| M | 1 | 2 | 3 |
|---|---|---|---|
| 1 | c2 = d2 | c1 | d1 |
| 2 | c4 | c3 = e2 | e1 |
| 3 | d4 | e4 | e3 = d3 |

| c2 | c1 | c3 |
|---|---|---|
| – | c4 | – |

,

| d2 | d1 | c3 |
|---|---|---|
| – | d4 | – |

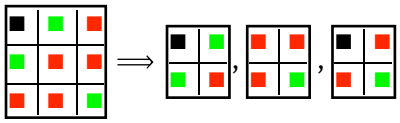
,

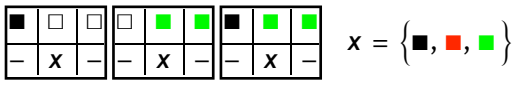
| e2 | e1 | e3 |
|---|---|---|
| – | e4 | – |

$\Longrightarrow$

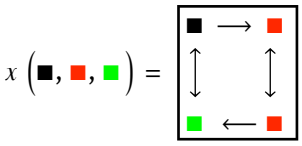
| $CA_{1.3}$ | $CA_1$ | $CA_3$ |
|---|---|---|
| $CA_1$ | $CA_{1.2}$ | $CA_2$ |
| $CA_3$ | $CA_2$ | $CA_{2.3}$ |

**Interactional exmple**






Chiastic structure of composed kenomic 'states' $\{■, ■, ■\}$:



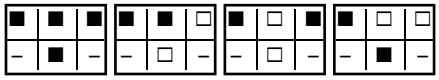

As it was designed before,
this scheme holds generally for all *m* an *n* of distrubuted and mediated contextures.
Each CA system is contained in *a* contexture of *a* polycontextural compound system.

### 4.3.2. Example

**Homogene composition**

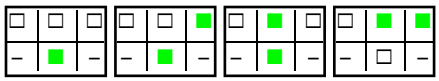
**keno[150]$^{(3)}$ =[**[1001], [0110], [1221]**]**; ((■, □), (□, ■ ), (■, ■))

**keno [150 – 1]** = r1 .7 .8 .4 = [1001]



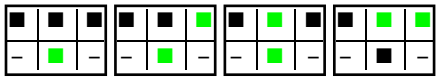
| Nr.\l | 1 | 2 | 3 | 4 | **5** | 6 | 7 | 8 | 9 | rule = [[150 – 1]; 1, 7, 8, 4] |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | □ | □ | □ | □ | ■ | □ | □ | □ | □ | 8, 7, 8, 4 |
| 2 | □ | □ | □ | □ | □ | ■ | □ | □ | □ | 1, 7, 8, 4 |
| 3 | □ | □ | □ | ■ | □ | □ | ■ | □ | □ | 7, 8, 4, 7, 8, 4 |

**keno [150 – 2]** = r1 .7 .8 .4 = [0220]; (□, ■ )



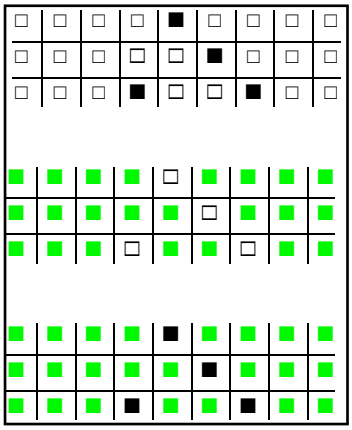
| Nr.\l | 1 | 2 | 3 | 4 | **5** | 6 | 7 | 8 | 9 | rule = [[150 – 2]; 1, 7, 8, 4] |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ■ | ■ | ■ | ■ | □ | ■ | ■ | ■ | ■ | 8, 7, 8, 4 |
| 2 | ■ | ■ | ■ | ■ | ■ | □ | ■ | ■ | ■ | 1, 7, 8, 4 |
| 3 | ■ | ■ | ■ | □ | ■ | ■ | □ | ■ | ■ | 7, 8, 4, 7, 8, 4 |

**keno [150 – 3]** = r1 .7 .8 .4 = [1221]; (■, ■)



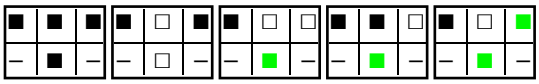
| Nr.\l | 1 | 2 | 3 | 4 | **5** | 6 | 7 | 8 | 9 | rule = [[150 – 3]; 1, 7, 8, 4] |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | 8, 7, 8, 4 |
| 2 | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | 1, 7, 8, 4 |
| 3 | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | 7, 8, 4, 7, 8, 4 |

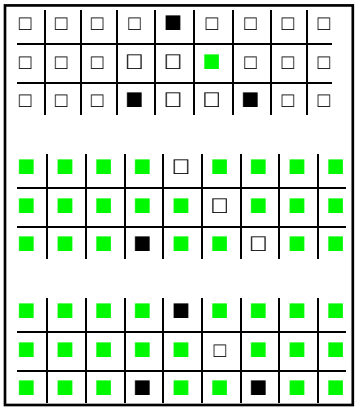
**keno [150] (3) :**



**Interactional constellations**

**keno [150 =]** = r1 .8 .13 .11 .14 = [01 222]





## 4.4. Kenomic CAM?

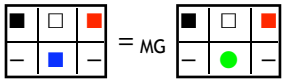
The question is what kind of physical realization of kenomic cellular automata could be imagined to transform kenomic CAs into kenomic automata machines?
Interacting grids of memristive crossbar systems is the answer.

Is it time for a new *"Cellular Automata Machine"* (Toffoli/Margolus)?

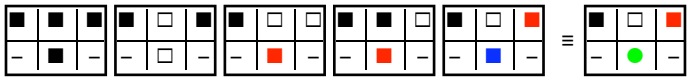
**4.5 APPENDIX**
**Polysemy and morphogrammatic saturation**
Polysemy holds inside and between Cas. The conventional start of a CA run has not to be restricted to a single beginning it is possible to have polysemy right at the start of a run.

(Gunther, Cybernetic Ontology, 1962, p. 95)

http://www.thinkartlab.com/pkl/archive/GUNTHER-BOOK/CYBERN1.html

 = MG  = rule[15]

**keno [polysem]** = r1 .8 .13 .14 .15 = [10 223] | [10 224]

 .

| Nr. | 1 | 2 | 3 | 4 | **5** | 6 | 7 | 8 | 9 | rule = [[polysem]; r1 .8 .13 .14 .15] |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | □ | □ | □ | □ | ■ | □ | □ | □ | □ | 11, 8, 13 |
| 2 | □ | □ | □ | ■ | □ | ■ | □ | □ | □ | 11, 8, 8, 8, 13 |
| 3 | □ | □ | ■ | □ | □ | □ | ■ | □ | □ | 8, 8, 1, 14, 8, 1 |
| 4 | □ | □ | □ | ■ | ■ | ■ | □ | ■ | ■ | 1, 1, 14, 8, 14, 8, 14, 14 |
| 5 | ■ | ■ | □ | ■; ● | □ | ■ | □ | ■; ● | ■ | 13, 11, 15, 8, 8, 8, 15, 11 |
| 6 | ■ | ■ | ■; ● | □ | □ | □ | □ | ■; ● | ■ | 13, 1, 11, 13, 1, 1, 11, 13, 11 |
| 7 | ■ | ■; ● | ■ | ■ | ■ | ■ | ■ | ■; ● | ■ | 15 |
| 8 | ■; ● | □ | ■ | ■ | □ | □ | ■; ● | □ | ■; ● | 15 |
| 9 | □ | □ | □ | □ | □ | □ | □ | □ | □ | stop |