

— vordenker-archive —

Rudolf Kaehr

(1942-2016)

Title

Sketch of Memristic Cellular Automata
A concept of time-oriented morphic cellular automata

Archive-Number / Categories

3_06 / K11, K12

Publication Date

2011

Keywords

Cf.: [Memristive Cellular Automata \(3_05\)](#)

Disciplines

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

Abstract

A concept of time-oriented morphic cellular automata
Cf.: [Memristive Cellular Automata \(3_05\)](#)

Citation Information / How to cite

Rudolf Kaehr: "Sketch of Memristic Cellular Automata", www.vordenker.de (Sommer Edition, 2017) J. Paul (Ed.),
URL: http://www.vordenker.de/rk/rk_Sketch-of-Memristic-Cellular-Automata_2011.pdf

Categories of the RK-Archive

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

Sketch of Memristic Cellular Automata

A concept of time-oriented morphic cellular automata

Rudolf Kaehr Dr.phil[®]

Copyright ThinkArt Lab ISSN 2041-4358

Abstract

The concept of morphic cellular automata with time-orientation is proposed as a further explication of memristive interactions.

===SHORT VERSION ===

1. Metaphor towards memristive CAs

1.1. Motivation

Conflict-free endomorphisms

A mapping of elements on a grid of cells is well established if the mapping is restricted to some obvious rules. There should be only one element per cell. And the cells are well defined as identical places in a topology of a grid. Also the mapping should be restricted to one cell only, and not involving neighbor cells of the target constellation too. Each neighbor cell of a cell becomes in a next application a cell and therefore gets its occupation by one element. Such a mapping of all cells by elements is called a configuration. Configurations appears in time-steps. For each time-step there is a single configuration of cells and their elements. Time-steps are evolving along a numeric time line. Configurations of time-steps are disjunct.

This concept is well realized by 1-dimensional cellular automata, i.e. 1 D CAs.

This is a perfect working concept and nothing has to be changed or added.

The desire of normal CAs is to produce complexity on the base of simplicity is realized in the framework of elementaristic order and a presumed homogeneous concept of a universe in which such complexity happens.

The mechanism of transition from time $t=0$ to $t=1$ is depicted classically with the following diagram:

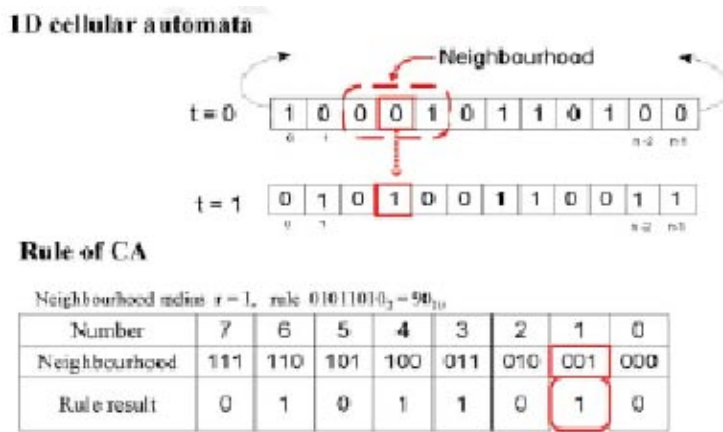


Fig. 1. 1D cellular automata with neighborhood = 1.

Conflict with polysemy

Nevertheless, it might turn out to be quite annoying. And it could be of more fun to introduce a different concept.

Say, why should the transition function based on the local rules deliver one and only one result to the configuration?

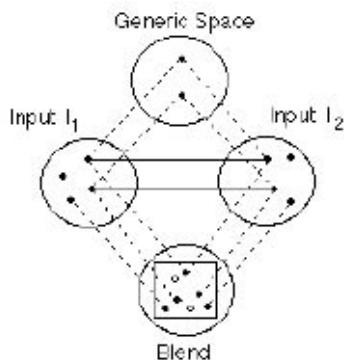
What happens if we allow, say 2 elements?

Obviously they will produce a conflictive situation. There is no place for 2 elements per local rule and single cell because all neighbors of the cell are treated equally the same, they get occupied by one and only one element.

A compromise could resolve the conflict: only the identical elements are allowed to be re-occupied by an element. Hence a kind of an invisible overlapping could do the job. But this would not only restrict the mappings to the very same elements and produce the very same configuration, it is also quite superfluous.

Hence, there is still the desire to map different elements onto the same cell.

Fusion, glue and blend



<http://www.thinkartlab.com/CCR/rudys-chinese-challenge.html>

The basic obstacle towards a construction of visibly overlapping mappings are not the cells, neither the mapping procedure but the concept of *identity* of the involved elements (states, symbols).

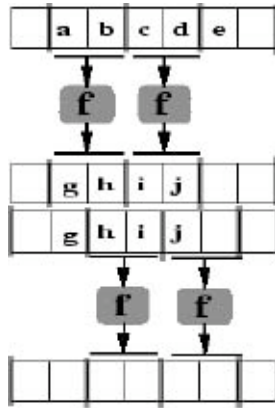
Hence, we have to get rid of this inherited obstacle. But to get rid of the identity of the elements, symbols or states, mappings and systems would necessarily destroy the whole approach of cellular automata as such, and everything else too.

Diagrams

A simple diagram may depict the difference between overlapping monomorphies and non-overlapping blocked elements.

The *block* diagram shows how *blocks* are treated by the transition function *f*. Blocking is reorganizing the blocks of elements without getting in trouble by any overlapping of elements at all.

Block rules without conflicts



Jörg R. Weimar

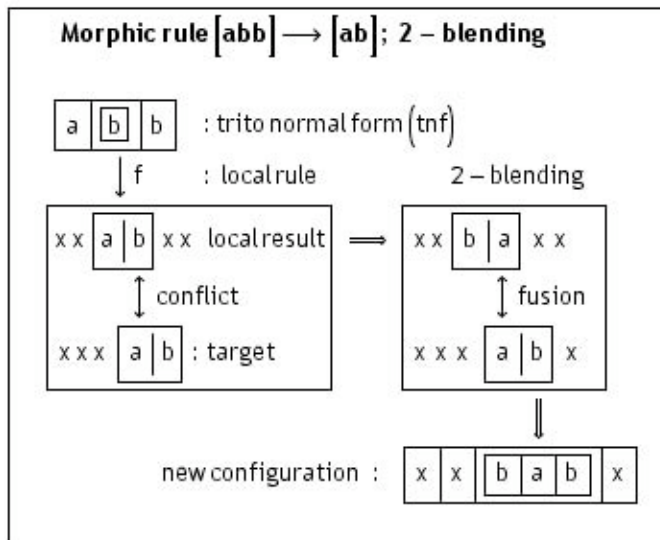
<http://www.jweimar.de/jcasim/main/node10.html>

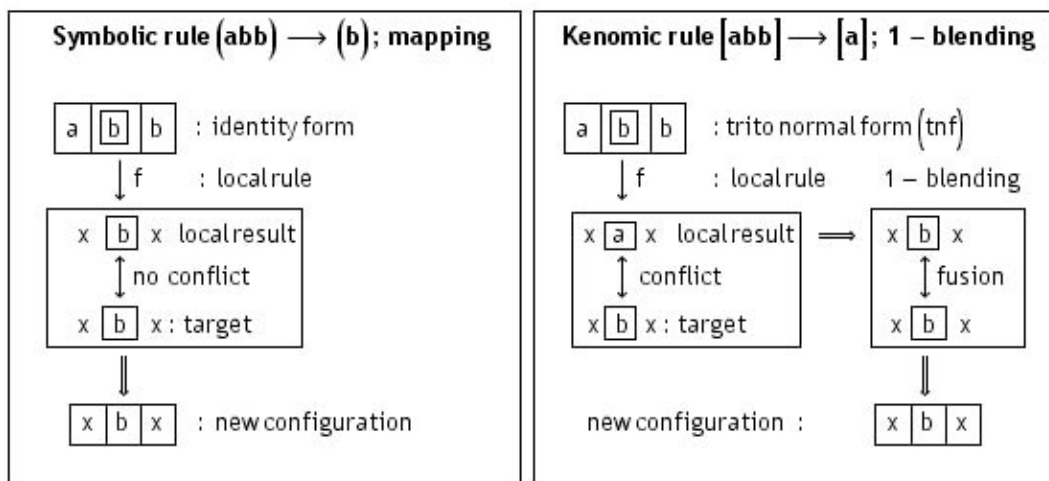
1.2. Explication of memristive rules

1.2.1. Mechanism of morphic elementary CA rules

The diagram for the morphic rules for dyads, i.e. 2-element monomorphies, shows how the transition mapping for monomorphies is producing an *overlapping* at the target configuration. Focused on the elements of the monomorphies there is a conflicting situation between the monomorphy of the mapping, [ab] and the monomorphy of the target constellation, [ba]. Because monomorphies are defined as *patterns* (morphograms) and not by their elements as such, a fusion between mapping and target monomorphy is applied which resolving the conflict without changing the structure of the pattern. Hence, the applied monomorphy of the rule, [aba] --> [ab], is properly mapped onto the target constellation, [ab], without changing its monomorphic definition as a dyad. Hence, [b] --> [ab] \equiv_{MORPH} [ba] --> [bab].

Morphic rules are offering a right- and a left-oriented mapping of monomorphies onto the target constellations.





Morphograms

Now, we want to keep the *mechanism* of CAs. But not the conflict producing elements (data structure, states, symbols).

A good compromise could be to accept not the signs of a CA alphabet but the *patterns* of sign-configurations only. Hence, the objects of such an alphabet-less machine would be patterns abstracted from sign-configurations and the rules of the CA mechanism would be defined by patterns only.

This approach seems to offer a good chance to overlap other patterns without getting into conflict with the elements of the representation of the pattern. Again, the elements don't count. What is in the game are the patterns, called *morphograms*. Nevertheless, to write and realize patterns we have to use signs for elements as forms of presentation of morphograms.

This concept of togetherness of elements demands to abandon the identity of signs in favor of morphograms.

Having introduced morphograms, morphic local rules are easily defined and overlapping configurations are becoming natural situations of morphic CA transitions.

As one of many consequences, our morphicCA gets some *temporal orientation* not only by the way an application of the morphic local rules happens but much more directly as a consequence of the *asymmetric* structure of the configurations of the local rules themselves.

Kenomic CAs

Morphic rules are build on the base of monomorphies of morphograms. Monomorphies are parts of morphograms as sings are elements of sign sequences. Monomorphies are build up and realized realized by kenograms. Kenograms are elementary parts of a morphic patterns. This motivates the definition of kenomic CAs. KenoCAs are realized by kenomic *endomorphisms*, and are therefore still keeping the properties of non-overlapping in CAs alive.

<http://memristors.memristics.com/CA-Compositions/Memristive%20Cellular%20Automata%20Compositions.html>

Normal CAs

Despite the *dynamical* character of the behavior of classical CAs they nevertheless might be considered as "frozen" kenomic CAs involving a reduction of the rule set and the mechanisms of building complexions of composed kenomic CAs.

Memristivity

What could be the purpose of the construction of time-oriented morphic CAs?

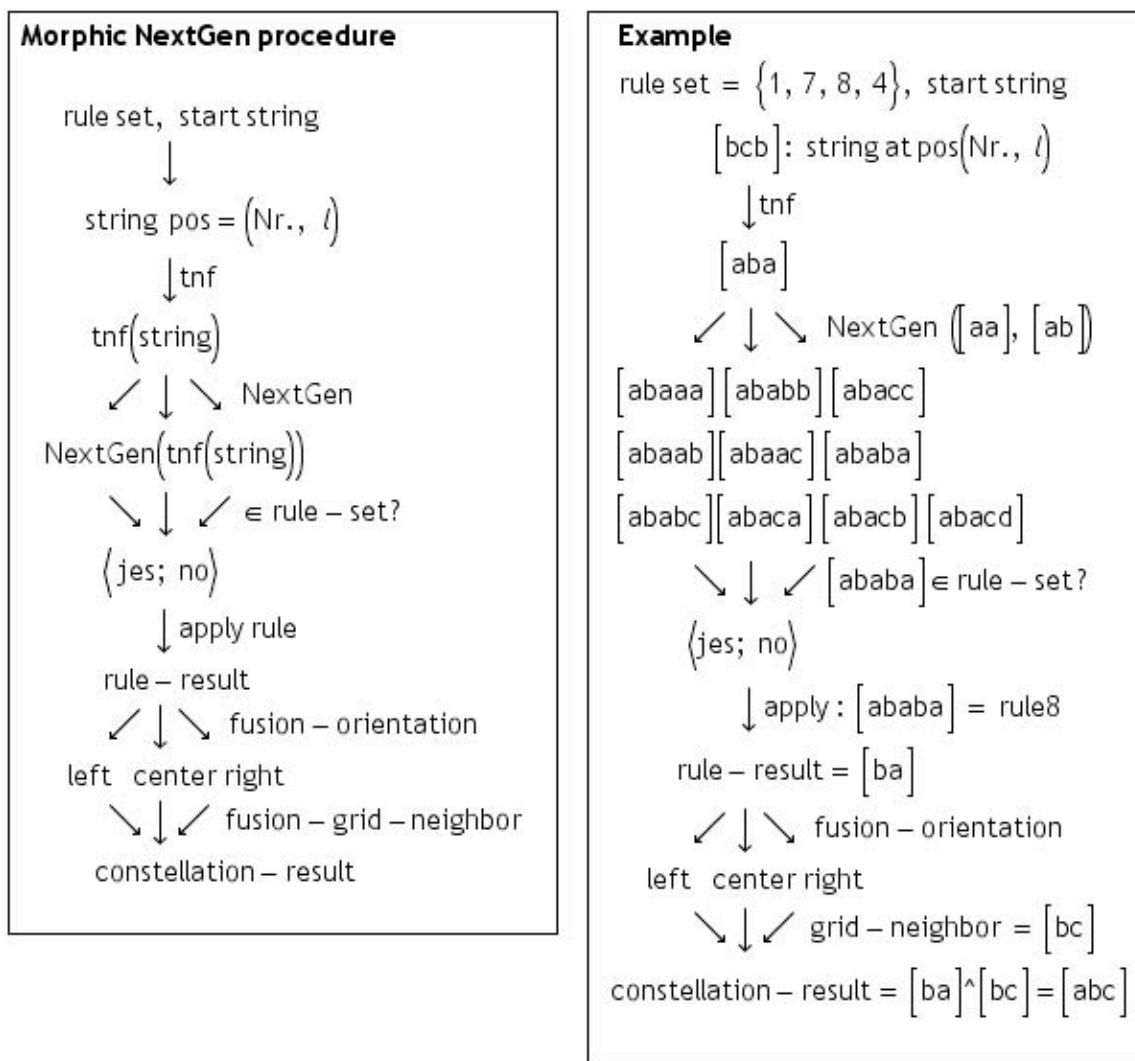
According to the studies into the memristivity of memristor-based crossbar systems, *time-* and *history-* dependence and the overlapping, fusion or blending, i.e. chiastic interplay of *computation* and *memory* functions are crucial for a new paradigm of nanotechnological computing.

Novelty

Morphograms, invented 1962 by the cybernetician and philosopher Gotthard Gunther, and the theory of the interactivity of morphograms, *morphogramatics*, are still a quite intriguing project to realize. Morphic CAs are implementing *time-dependence* by their elementary rules directly into the automatism even before any information processing has to be involved.

To opt for a 'machine'-interpretation of morphograms might be even more surprising. In fact, morphograms had always been reflected in the context of operational, machine- and program-related constructions. But a direct application of morphogramatics in the very context of cellular automata constructions is indeed a brand new adventure.

1.2.2. Procedure of morphic elementary CA rules



The *NextGen* rule, or procedure, is not simply realized with an application of a local rule, the result of such an application has to be embedded and aligned with the states of the cells of the target configuration. Because of the non-atomic but morphic structure of the result there is a space conflict for the integration of the morphic result. Hence, a *fusion* of the rule result and the neighbor state of the configuration has to be established. This is naturally realized with the fusion-rules of *monomorphies* of morphogramatics.

<http://memristors.memristics.com/MorphoProgramming/Morphogrammatic%20Programming.html>

1.3. Neighborhood rules for morphic CAs

1.3.1. Mechanism of morphic CA rules

As a first step to the realization of monomorphic transitions a new neighborhood relationship has to be introduced.

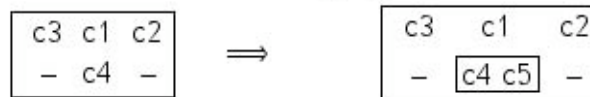
Not only the transition rules, discrete or monomorphic, are defined with neighbors but the placement of their results has to be kenomically adjusted to the results in the grid of constellations of the application of previous rules.

Hence, at first there are two kinds of neighborhood relations to distinguish: the neighborhood cells of the primary generic *rules* and the neighborhood relations of the results of the rules for a new result of a *configuration*.

This is achieved with the rule scheme for monomorphic successors (NextGen, results) with at least dyadic successors (monomorphies) in contrast to the singular atomic successors of classical m-neighborhood rules and the monadic successors for kenomic CA rules.

The main rule is still composed by the set of local rules fulfilling the combinatorics with length 4 for 2 elements and 5 for 3 elements. Hence, the combination of rules for 1D kenoCA shall be applied for 1D morphCA.

Discrete: $c312 - c4 \implies$ Overlapping: $c312 - c45$



Nr. \ l	1	2	3	4	5	6	7	8	rule = [[morph]; {1, 7, 8, 4}]
1	□	□	□	□	■	□	□		rules : 7, 8, 4
2	□	□	□	■	□	■ ≡ ■	■		7, 8, 4 : coinc (6 _{r8} , 6 _{r4}) : ■ ≡ ■ =
3	□	□	□	■	□	■	■ ≠ □	■	4, 7, 8, 4 : confl (7 _{r4} , 7 _{r7}) : ■ ≠ □
3'	□	□	□	■	□	■	■ □		resolution (7 _{r4} , 7 _{r7}) : □ ■ = MG

Coincidence and kenomic adjustment

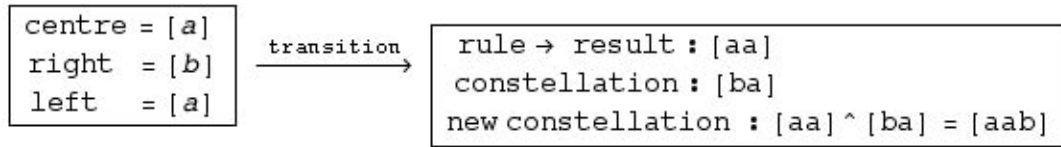
- [□ ■ □] = [aba]
- 1; 4, 5, 6; r8 : (aba) → (ab) : 5, 6
- 1; 5, 6, 7; r4 : (baa) → (ab) : 6, 7 : coinc (6_{r8}, 6_{r4})
- 2; 4, 5, 6; r8 : (bab) → (ab) : 5, 6
- 2; 5, 6, 7; r4 : (baa) → (bb) : 6, 7 : coinc (6_{r8}, 6_{r4})
- 2; 6, 7, 8; r7 : (bba) → (ab) : 7, 8 : confl (7_{r4}, 7_{r7}) : a ≠ b

r8 (5, 6, 7) conflict r4 (6, 7, 8) results into monomorphic addition :

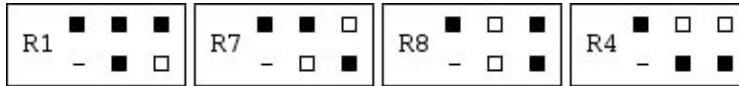
$$2; 6, 7, 8 : r7 : (bb) ; (bba) \rightarrow (ab) : (bb) \wedge (ab) = (bbba) \Rightarrow (bba) : 6, 7, 8.$$

$$\boxed{\square \blacksquare} =_{MG} \boxed{\blacksquare \square}$$

$$\blacksquare \Rightarrow \boxed{\blacksquare \blacksquare} \wedge \boxed{\square \blacksquare} = \boxed{\blacksquare \blacksquare \square}$$



Correct notation

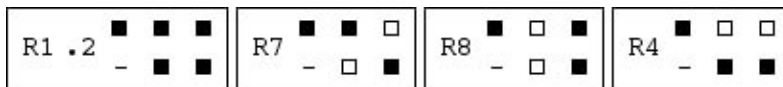


Nr./1	1	2	3	4	5	6	7	8	9	rule = {1, 7, 8, 4}
1	□	□	□	□	■	□	□	□	□	1, 1, 1, 7, 8, 4, 1, 1, 1
2	□	■	□	■	□	■	■	□	■	1, 1, 1, 7, 8, 4, 1, 1, 1
3	□	■	□	■	□	■	□	■	□	8, 8, 8, 8, 8, 8, 8, 8, 8
4	□	■	□	■	□	■	□	■	□	8, 4, 7, 8, 8, 8
5	■	■	□	■	□	■	□	■	□	stop



Nr./1	1	2	3	4	5	6	7	8	9	rule = {1, 7, 8, 4}
1	□	□	□	□	■	□	□	□	□	12, 13, 1, 1 : [ca]^[cb] = [cab]
2	□	□	□	□	■	□	■	□	■	[cba]^[cb] = [cbab]; [cbab]^[cb] = [cbabc]
3	□	□	□	□	■	□	■	□	■	□
4	□	□	□	□	□	□	□	□	□	□
5	□	□	□	□	□	□	□	□	□	stop

- 2; 4, 5, 6: R12: [aba] - [cb]
- 2; 5,6,7: R13: [abb] - [ca] : [cb]^[ca] = [cba]; [ab]^[bc] = [abc]
- 2; 6,7,8: R1: [aaa] - [ab]: [cba]^[ab] = [cbab]
- 2;7,8,9: R1: [aaa] -[ab]: [cbab]^[ab] = [cbaba]
- 2;8,9,10: R12: [aba]-[cb]: [cbab]^[cb] = [cbabc]



Nr./1	1	2	3	4	5	6	7	8	9	rule = {1, 7, 8, 4} : right
1	□	□	□	□	■	□	□	□	□	1 \rightarrow 7 \rightarrow 8 \rightarrow 4 \rightarrow 1
2	-	-	■	■	□	■	■	-	-	7 \rightarrow 4 \rightarrow 7 \rightarrow 8 \rightarrow 4 \rightarrow 7 \rightarrow
3	-	x	■	■	□	■	■	x	-	1, 1, 7, 8, 4, 7, 4 \rightarrow
4	-	■	■	■	□	■	■	x	x	4, 1, 7, 8, 4, 7, 1, 4 \rightarrow
5	■	■	■	■	□	■	■	x	x	stop

Nr /.	1	2	3	4	5	6	7	8	9	rule = {1, 7, 8, 4}, centre: left ???
1	□	□	□	□	■	□	□	□	□	7, 8, 4, →
2	-	-	-	■	□	■	■	-	-	1, 7, 8, 8, 4, 1
3	-	-	x	■	□	■	■	-	-	1, 1, 7, 8, 8, 1, 7
4	-	x	x	■	□	■	■	■	-	1, 1, 7, 8, 8, 4, 1, 1, 7
5	x	x	x	■	□	■	■	■	x	stop

The new value is determined by the applied local rule and the existing value of the constellation in the grid: if the left-value y and the new value x at y is different then value x changes to y, and z is building with y the new monomorphy.

The center-application is left- and/or right-oriented.

1.3.2. Left-right-asymmetry of the applications of local rules

The classical simultaneity of the application of rules, which is a 1-1-mapping has to be modified because of the asymmetry of the beginnings, center-, right- and left-applications.

For classical CAs we obviously have a non-oriented structure: *“Notice that the order in which we calculated the rows of squares did not matter, we could have done them right-to-left, or in an arbitrary order, instead of left-to-right.”* (F.L. Thaulaw, 2010)

Hence, the beginning of the application and its direction is a determining property of the automaton. This formal property might be instrumentalized to introduce a kind of an observer-dependence of the construction. The understanding of the developed configurations are depending on the direction of its construction and re-construction, i.e. interpretation. Time- and history-dependence might be a reason too. Kenomic and symbolic CA rules are producing disjunct results. Morphic CAs are producing overlapping results.

Therefore symbolic and kenomic CAs are non-directed, while morphic CAs are time-oriented measured by the overlapping function. There is a difference in approaches which are introducing, say time-dependence secondary with the help of arbitrary techniques and an approach which is implementing time-dependence at the very beginning of the construction.

Nr /.	1	2	3	4	5	6	7	8	9	rule = {1, 7, 8, 4}]
1	□	■	□	■	■	■	□	■	□	7, 8, 8, 4, 1, 7, 8, 8 : left to right; 1; 1 – 9
2	x	■	x	■	■	■	x	■	x	-
3	x	■	x	x	x	■	x	■	■	8, 8, 4, 1, 7, 8, 8, 4 : right to left; 1; 9 – 1

The results for left-, right- and centre-directions of the application of the sub-rules of the general rule are different.

Therefore, the axiom of simultaneity for classical CA has to be deconstructed towards an asymmetric double strategy.

Because there is a bijection between the chain of rule names and the new configuration, the chain of rule names gives the answer of the internal structure of the new configuration, i.e. the combination of rule application and fusion. The chain of local rules defines the global rule of the application. Hence the intricate combinations and fusions of the new configuration are reflected by the rule chain and can therefore be reconstructed from the global constellation with the application of the chain of the local rules.

The right configuration (2; 1-9) is characterized by the rule chain (7, 8, 8, 4, 1, 7, 8, 8 : left to right of 1; 1 – 9).

The left configuration (3; 9-1) is characterized by the rule chain (8, 8, 4, 1, 7, 8, 8, 4: right to left of 1; 1-9).

The classical arrangement for CAs *“There’s one cell at each node of the lattice.”* (Thaulaw) holds in some sense for morphCAs too. But because of the overlapping of the morphograms the states might also be considered as double defined and derived from different rules albeit resulting in coincidence.

Thus, the states in morphCAs are functionally doubled, polysemical and ambiguous but as results they might be considered by an *external* interpretation as mono-semical.

$$\text{RuleChain}_{(8, 8, 4, 1, 7, 8, 8, 4 : \text{right})}([\square \blacksquare \square \blacksquare \blacksquare \blacksquare \square \blacksquare \square]) = [x \blacksquare x \ x \ x \ \blacksquare \ x \ \blacksquare \ \blacksquare]$$

$$\text{RuleChain}_{(7, 8, 8, 4, 1, 7, 8, 8 : \text{left})}([\square \blacksquare \square \blacksquare \blacksquare \blacksquare \square \blacksquare \square]) = [x \ \blacksquare \ x \ \blacksquare \ \blacksquare \ \blacksquare \ x \ \blacksquare \ x]$$

1 – D morphCA

$$\text{RuleChain}_{\text{right}}([c]) \neq \text{RuleChain}_{\text{left}}([c]) \implies e_{\text{left}}([c]) \neq e_{\text{right}}([c])$$

Classical CA

For classical 1 D CAs(0,1) the rule application is *neutral* to the distinction of right- and left-orientation. This corresponds with the principle of *simultaneity* of the rule application for CAs.

1 – D CA

$$\text{RuleChain}_{\text{left}}(c) \equiv \text{RuleChain}_{\text{right}}(c) = e(c)$$

1.3.3. Left-right-asymmetry of local rules

Orientation implemented by the asymmetry of local rules

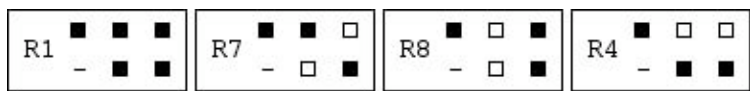
A further concretization of the implementation of *orientedness* might be achieved with the introduction of the difference of right- and left-oriented local cellular automata rules. Instead of the restricted orientation achieved by different applicative orientations of the local rules, the new approach is based on the structural *asymmetry* of the local rules which are defining directly a left- or right-orientation for their application.

The two schemes:

right: (c3,c1,c2) --> (c4,c5) and left local rules: (c3,c1,c2) --> (c4,c6)
 are inviting to study a new class of elementary cellular automata.

Those automata are time-oriented by the definition of their asymmetric rules, and not just by the external application of symmetric elementary local rules.

right local rules: (c3,c1,c2) --> [c4c5]:



left local rules: (c3,c1,c2) --> [c4c6]:



morpho-neutral local rules (c3,c1,c2) --> [c6c4c5]:



Examples

left local

Nr./l	1	2	3	4	5	6	7	8	9	rule = {1, 7, 8, 4}, left local
1	□	□	□	□	■	□	□	□	□	7, 8 , 4,
2	□	-	-	■	x	x	□	□	□	8, 4, 1, 1
3	□	-	x	■	■	■	■	□	□	7, 4, 1, 1, 4
4	x	■	■	■	■	■	■	■	□	□

right local

Nr./l	1	2	3	4	5	6	7	8	9	rule = {1, 7, 8, 4}, right local
1	□	□	□	□	■	□	□	□	□	1; 7, 8, 4
2	□	□	■	■	x	■	■	□	□	4, 4, 7, 8, 4, 7
3	□	■	■	■	x	■	■	x	□	7, 4, 1, 7, 8, 4, 4, 4
4	x	■	■	■	x	■	■	■	■	□

keno-neutral local rules (c3,c1,c2) --> [c4]:



Nr./l	1	2	3	4	5	6	7	8	9	rule = {1, 7, 8, 4}, keno – rules
1	□	□	□	□	■	□	□	□	□	1; 7, 8, 4
2	□	□	□	x	x	■	□	□	□	4, 4, 7, 8, 4, 7
3	□	□	■	■	x	x	■	□	□	7, 4, 1, 7, 8, 4, 4, 4
4	□	x	■	x	■	x	x	■	□	□
5	■	x	x	x	x	■	x	x	■	stop

1 – DmorphCA

$$[c_1] =_{MG} [c_2] \implies e_{left}([c_1]) \neq e_{right}([c_2])$$

2. Formalism of CA

2.1. Formal definition of CA after Kari

Cellular Automaton CA

CA = finite items $\{d = \text{dimension}, S = \text{state set}, N = \text{neighborhood}, f = \text{local rule}\}$

$$\text{CA} = (d, S, N, f)$$

configuration: $c : \mathcal{Z}^d \rightarrow S$, $d = \text{dimension}$

local rule: $f : S^n \rightarrow S$, where n is the size of the neighborhood.

State: $f(a_1, a_2, \dots, a_n)$ is the new state of a cell whose n neighbors were at states a_1, a_2, \dots ,

Neighborhood

Let $N = (\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n)$ be a vector of n distinct elements of \mathcal{Z}^d .

The neighbors of a cell at a location $\vec{x} \in \mathcal{Z}^d$ are the n cells at locations $\vec{x} + \vec{x}_i$, for $i = 1$

The cells change their states *synchronously* at *discrete* time steps.

The next state of each cell depends on the current states of the neighboring cells according to f .
All cells use the same rule, and the rule is applied to all cells at the same time.

global transition function

Configuration c becomes in one time step the configuration e where for all $\vec{x} \in \mathcal{Z}^d$,

$$e(\vec{x}) = f(c(\vec{x} + \vec{x}_1), c(\vec{x} + \vec{x}_2), \dots, c(\vec{x} + \vec{x}_n)).$$

We say that $e = G(c)$, and call $G : S^{\mathcal{Z}^d} \rightarrow S^{\mathcal{Z}^d}$ global transition function of the CA.

Examples for neighborhoods

$\{-1, 0, 1\}$: 1-D CA neighborhood,

$\{(0, 0), (\pm 1, 0), (0, \pm 1)\}$: Von Neumann neighborhood,

$\{-1, 0, 1\} \times \{-1, 0, 1\}$: Moore neighborhood.

(Kari, IntroCA, p.42)

Kari gives a very useful intro to CAs. The full mathematical theory is to collect at the same address.

<http://users.utu.fi/jkari/ca/CAintro.pdf>

2.2. Preliminary formal definitions for morphic CAs**2.2.1. Sketch of basic definitions for morphCA**

Some preliminary formalizations are sketched to precise the approach to morphic CAs with tin

$$\text{morphCA} = \left[(d, S, N, f)_{\text{morph}}, \omega, \varphi \right]$$

d = dimension,

S = collection of monomorphisms

N = neighborhood,

f = local rule

ω = orientation of the local application f

φ = fusion of local and global constellations

N : neighborhood $\vec{x} \in N$: right neighborhood $\overleftarrow{x} \in N$: left neighborhood**1 – D neighborhood** $N = \{-1, 0, 1\}$:classic : $N \rightarrow \{0\}$ morph_{left} : $N \rightarrow (-1, 0)$ morph_{right} : $N \rightarrow (0, 1)$ **c : configuration** $c_{\text{right}}(\vec{x} + \vec{x}_1)$ $c_{\text{left}}(\overleftarrow{x}_1 + \overleftarrow{x})$ **e : transition** $e(\vec{x}) = f(\varphi(c(\vec{x} + \vec{x}_1), c(\vec{x} + \vec{x}_2), \dots, c(\vec{x} + \vec{x}_n)))$. $e(\overleftarrow{x}) = f(\varphi(c(\overleftarrow{x}_1 + \overleftarrow{x}), c(\overleftarrow{x}_2 + \overleftarrow{x}), \dots, c(\overleftarrow{x}_n + \overleftarrow{x})))$. **S : finite set of states** (monomorphies) $\vec{x} \in \text{Stirling2}(\mathcal{Z}, d)$, class of trito – normal forms $c(\vec{x})$: current tnf – state **f : local rule** (local transition map δ) $f_{\text{morph}} : (a_1, a_2, \dots, a_n) \rightarrow (a_1, a_2, \dots, a_n)$, with $a_i \in \text{Monomorphies}$ $f_{\text{tnf}} : \text{Sn2}(S, n) \rightarrow \text{Sn2}(S, l)$, $n = \text{neighborhood}$, $l = \text{length of result}$.

φ : fusion

$$\varphi \left(c \left(\overrightarrow{x} + \left(\overrightarrow{x}_{i_1}, \overrightarrow{x}_{i_2} \right)_i \right), c \left(\overrightarrow{x} + \left(\overrightarrow{x}_{i_1}, \overrightarrow{x}_{i_2} \right)_j \right) \right), \text{ for } 1 \leq i, j \leq n$$

$$\varphi \left(\left(\overrightarrow{x}_{i_1}, \overrightarrow{x}_{i_2} \right)_i, \left(\overrightarrow{x}_{i_1}, \overrightarrow{x}_{i_2} \right)_j \right) \rightarrow \left(\overrightarrow{x}_{i_1}, \varphi \left(\overrightarrow{x}_{i_2}, \overrightarrow{x}_{j_2} \right) \right)_i$$

$$\forall i, j: e_{\text{morph}} \left(\overrightarrow{x} \right) =$$

$$f: \varphi \left(c \left(\left(\overrightarrow{x_{i_1} x_{i_2}} \right), \left(\overrightarrow{x_{j_1} x_{j_2}} \right) \right) \rightarrow c \left(\overrightarrow{x_{i_1}, \varphi \left(x_{i_2}, x_{j_2} \right)} \right) \right),$$

$$\forall i, j: e_{\text{morph}} \left(\overleftarrow{x} \right) =$$

$$f: \varphi \left(c \left(\left(\overleftarrow{x_{i_1} x_{i_2}} \right), \left(\overleftarrow{x_{j_1} x_{j_2}} \right) \right) \rightarrow c \left(\overleftarrow{x_{i_1}, \varphi \left(x_{i_2}, x_{j_2} \right)} \right) \right),$$

$$f: \varphi \left(c \left(\left(\overleftarrow{x_{i_1} x_{i_2}} \right), \left(\overleftarrow{x_{j_1} x_{j_2}} \right) \right) \right) \rightarrow c \left(\varphi \left(\overleftarrow{x_{i_1}, \varphi \left(x_{i_2}, x_{j_2} \right)} \right) \right) \bullet$$

 ω : orientation

The application of the fusion operation of the local rules are oriented by a left or a right orientat

$$\text{right: } \left(\overrightarrow{x} + \left(\overrightarrow{x}_{i_1}, \overrightarrow{x}_{i_2} \right) \right), \text{ for } i = 1, 2, \dots, n :$$

$$\text{right-neighbors of } \overrightarrow{x}, \text{ and monomorphy } \left(\overrightarrow{x}_{i_1}, \overrightarrow{x}_{i_2} \right) = \left(\overrightarrow{x_{i_1} x_{i_2}} \right)$$

$$\text{left: } \left(\overleftarrow{x} + \left(\overleftarrow{x}_{i_1}, \overleftarrow{x}_{i_2} \right) \right), \text{ for } i = 1, 2, \dots, n$$

$$\text{left-neighbors of } \overleftarrow{x}, \text{ and monomorphy } \left(\overleftarrow{x}_{i_1}, \overleftarrow{x}_{i_2} \right) = \left(\overleftarrow{x_{i_2} x_{i_1}} \right)$$

$$c \left(\varphi \left(\left(\overrightarrow{a_{i_1} b_{ij_2}}, \overrightarrow{b_{j_2} a_{j_1}} \right) \right) \right) = \begin{cases} \text{right: } c \left(\overrightarrow{a_{i_1} b_{ij_2}, a_{j_2}} \right) \\ \text{left: } c \left(\overleftarrow{a_{i_1} b_{ij_2}, a_{j_2}} \right) \\ \text{centre: left / right} \end{cases}$$

morphCA :

$$G_{\text{morph}} : \left(\text{Sn2} \left(S, \mathcal{Z}, d \right) \rightarrow \text{Sn2} \left(S, \mathcal{Z}, d \right) \right) \rightarrow \text{Sn2} \left(S, \mathcal{Z}, d \right) /_{\text{fusion}}$$

Example for fusion

$$f: \varphi : \left(c \left(\left(\overrightarrow{(a_{i_1} b_{i_2})}, \overrightarrow{(a_{j_1} b_{j_2})} \right) \right) \rightarrow c \left(\overrightarrow{\left(\overrightarrow{(a_{i_1} b_{i_2})}, \varphi \left(\overrightarrow{(a_{j_1} b_{j_2})} \right) \right)} \right) \right) = c \left(\overrightarrow{\left(\overrightarrow{(a_{i_1} b_{i_2})}, \overrightarrow{(b_{j_2} a_{j_1})} \right)} \right) = c \left(\overrightarrow{(a_{i_1} b_{i_2}, a_{j_1})} \right) \bullet$$

For $[ab] = c$, current state, $[ba] = f$, rule result

$$c \left(\varphi \left([ab], \varphi [ab] \right) \right) = c \left(\varphi \left([ab], [ba] \right) \right) = c \left([aba] \right)$$

new current state: $[aba] \bullet$

A global mapping on configurations is obtained by uniform and synchronous application of δ :

$$\forall c \in S^{Z^d}, \forall \vec{z} \in Z^d : A(c) \xrightarrow{\vec{z}} = \delta \left(c \xrightarrow{\vec{z} + \vec{n}_1}, \dots, c \xrightarrow{\vec{z} + \vec{n}_n} \right)$$

2.2.2. Classification of 1D morphic CA rule types

"The local dynamics, represented by the transition function of a cell, denoted by δ , is a function which receives as inputs the state of a cell and its 'neighbors', and computes by the local rules the next state of the cell." (Varun Bhatia)

Hence, the local dynamics of a 1-D CA, semiotic, indicational, kenomic and morphic can be defined by the following equations.

Depending on the kind of CA, the types of local rules are differentiated into 3 classes: the *classical* type of 1 D CAs, the *kenomic* rule type and the *morphic* rule types with left - and right-oriented rule types with tupels and one left-right-oriented rule type with a triple generation.

Classification of local dynamics

$x_i \in \text{Elements} :$
 $\delta_{\text{class}} (x_{i-1}, x_i, x_{i+1}) = x'_i,$

$[x_i] \in \text{Kenograms} :$
 $\delta_{\text{kenom}} (x_{i-1}, x_i, x_{i+1}) = [x'_i],$

$[xyz] \in \text{Monomorphies} :$

$$\delta_{\text{morph}} (x_{i-1}, x_i, x_{i+1}) = \begin{cases} [x_{i1} x_{i2}]_{c45} : \text{right} \\ [x_{i2} x_{i1}]_{c46} : \text{left} \\ [x_{i1} x_{i2} x_{i3}]_{c645} \end{cases}$$

2.2.3. Mediative compositions of CAs

Special distinctions of 1 – D CA for mediation of different 1 – D CAs :

Binary function: $(l, q, r) \mapsto f(l, q, r)$, $l = \text{left}$, $q = \text{center}$, $r = \text{right}$:

$CA_i, i = 1, 2, 3$:

$\text{dom}(l_i, q_i, r_i) = l_i$

$\text{cod}(l_i, q_i, r_i) = r_i$

Mediation μ

$\mu: (CA^1, CA^2) \rightarrow (CA^1 \amalg CA^2) \amalg CA^3 :$

$\text{cod}(CA^1) \cong \text{dom}(CA^2)$

$\text{cod}(CA_2) \cong \text{cod}(CA^3)$

$\text{dom}(CA^1) \cong \text{dom}(CA_3)$.

$CA^{(3)} = (CA^1 \amalg CA^2) \amalg CA^3 :$

$f^{(3)}(CA^{(3)}) = (f^1(CA^1) \amalg f^2(CA^2)) \amalg f^3(CA^3) \bullet$

Compostion of CA_1 and CA_2 : $\circ: CA_1, CA_2 \rightarrow CA_1 \circ CA_2$.

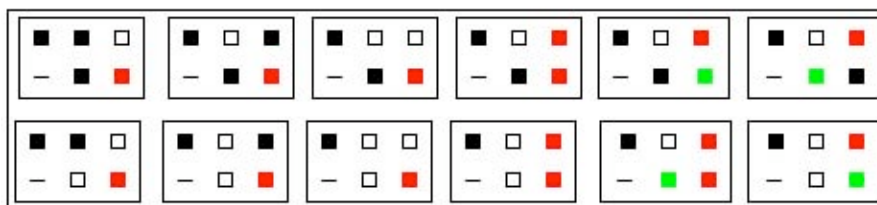
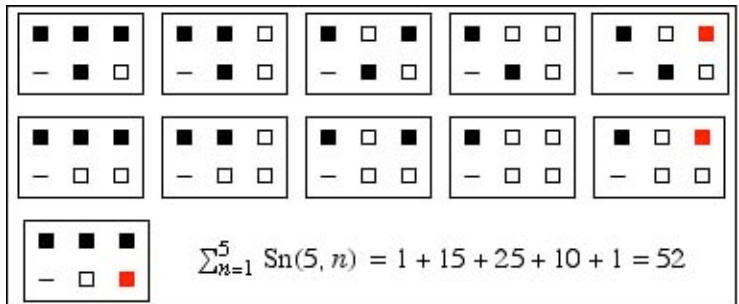
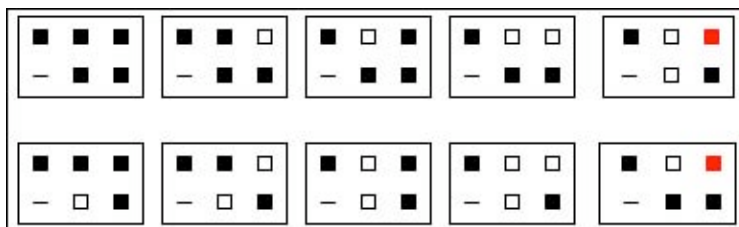
Bifunctionality of $\mu(CA_i^{(3)}, \circ, \amalg), i = 1, 2$:

Bifunctionality of compositions of mediated CAs

$$\begin{bmatrix} CA_2^1 & - & CA_2^3 \\ CA_1^1 & CA_2^2 & - \\ - & CA_1^2 & CA_1^3 \end{bmatrix} :$$

$$\left(\begin{array}{c} \left((CA_1^1 \circ_{1.0.0} CA_2^1) \right) \\ \amalg_{1.2.0} \\ \left((CA_1^2 \circ_{0.2.0} CA_2^2) \right) \\ \amalg_{1.2.3} \\ \left((CA_1^3 \circ_{0.0.3} CA_2^3) \right) \end{array} \right) = \left(\begin{array}{c} CA_1^1 \\ \amalg_{1.2.0} \\ CA_1^2 \\ \amalg_{1.2.3} \\ CA_1^3 \end{array} \right) \circ_{1.2.3} \left(\begin{array}{c} CA_2^1 \\ \amalg_{1.2.0} \\ CA_2^2 \\ \amalg_{1.2.3} \\ CA_2^3 \end{array} \right)$$

2.3. System of 2-morphic right CA rules



Null

