Rudolf Kaehr
(1942-2016)

Title

Reduction and Mediation of morphoCAs

Functional Analysis of morphoCAs

Archive-Number / Categories

3_48 / K12, K09, K11

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

Despite the decision for a morphogrammatic foundation of morphic cellular automata, a more function-oriented analysis of morphoCAs is applied to give some new insights into the techniques of reduction and mediation of the complication/complexity of morphic CAs as they have been proposed in previous papers.

Well known features of polycontextural and morphogrammatic systems (structurations), like super-addivity of mediation and super-subtractivity of decomposition of morphoCAs will be sketched.

Citation Information / How to cite

Rudolf Kaehr: "Reduction and Mediation of morphoCAs", www.vordenker.de (Sommer Edition, 2017) J. Paul (Ed.), http://www.vordenker.de/rk/rk_Reduction-and-Mediation-of-morphoCAs_2015.pdf

Reduction and Mediation of morphoCAs

Functional Analysis of morphoCAs

Dr. phil Rudolf Kaehr

ISSN 2041-4358
(work in progress, v. 0.3, October 2015)

Motivation

Despite the decision for a morphogrammatic foundation of morphic cellular automata, a more *function*-oriented analysis of morphoCAs is applied to give some new insights into the techniques of *reduction* and *mediation* of the complication/complexity of morphic CAs as they have been proposed in previous papers.

Well known features of polycontextural and morphogrammatic systems (structurations), like *super-addivity* of mediation and *super-subtractivity* of decomposition of morphoCAs will be sketched.

Stirling reductions

A very first reduction of the quantitative complexity of functions is given by the fact that morphoCAs are not based on exponential functions but on Stirling partitions. Hence, instead of having 3^{3^3} ternary functions of a three-valued CA, morphoCAs are based on StirlingS2, $\mathcal{S}_n^{(m)}$, hence *Sum[StirlingS2[27,m],{m,3}]*

The reduction is impressive:

From 3^{3^3} = 7 625 597 484 987 three-valued ternary functions to *Sum[StirlingS2[27,m],{m,3}] = 1 270 932 914 165* three-valued ternary Stirling functions.

There are 3^{3^2} = 19 683 three-valued binary functions and just *3281 = Sum[StirlingS2[9,m],{m,3}]* three-valued binary Stirling patterns. For the unary three-valued case there are 3^{3^1} *= 27* functions. And *Sum[StirlingS2[4,m],{m,4}] = 15* for the number of morphograms.

Redundancy-Reductions

Reduction of redundancy in the symbolic (numeric) interpretation of morphogrammatic CA rules. This kind of reduction is not changing the morphogrammatic structure of the morphoCAs but is minimizing the redundant elements of its interpretation. Only the relevant elements (symbols, numbers) of the interpretation of the morphogrammatic compounds necessary for the realization of morphoCAs are considered. Until now, this reduction didn't play a significant role in the definition of the newly introduced morphoCAs and their claviatures.

To make morphograms visible, they have to interpreted. But the complexity of the interpretations is contextually depending on the whole configuration of the morphogrammatic compounds. There is not just one reduced interpretation available for all situations. For non-reducible morphoCAs the full range of the possible interpretations is necessary to fulfill the definition and realization of the morphoCA.

Single morphograms get different complex interpretations depending on their morphoCA environment.

The morphogram [1] in ruleDM[{1,11,8,4,15}] or ruleDM[{1,11,8,9,15}] demands for an interpretation with the set of all possible mappings in the system of complexity 4: *{0,0,0}→0, {1,1,1}→1, {2,2,2}→2, {3,3,3}→3.*

In contrast, a similar constellation like *ruleDM[{1, 11, 3, 9, 15}]* demands in the same system for the interpretion of morphogram [1] just one mapping with *{0,0,0}→0*.

Depending on the context, there are other interesting cases too.
Just 3 interpretetions needed: *ruleDM[{1,2,12,13,14}]* with the mapping *{0,0,0}→0, {1,1,1}→1, {2,2,2}→2* for morphogram [1]. With a litle change by replacing the morphogram [14] by the morphogram [15], all interpretations accessible in the system are needed.

Because morphoCAs are dynamical systems, it might take a non-trivial amount of steps to observe the necessity of specific interpretations. The morphoCA may work without problems for some steps without a complementation of the interpretations. But it will show lacks of realizations some steps later.

It also turns out that the reductions are not necessarily fully working for arbitrary seeds, like Random seeds.

Nevertheless, not all morphoCAs need a full set of interpretations.

That shows nicely the context-dependence of the Interpretation of morphograms in morphoCAs.

Karnaugh reductions

A further functional reduction might be achieved with the help of Karnaugh maps. Pattern reductions are not the same as Karnough reductions. Pattern reductions are reducing the redundancy of the interpretations of the formulas without changing the formal structure of their definitions (literals, terms). In contrast, Karnaugh mappings are changing the definitions of the functions by reducing the numbers of variables of the original formula albeit without disturbing their functional purpose.

Equivalent to the Karnaugh maps, simplification of Boolean expressions are delivering the same result. For morphoCAs, Boolean expressions have to be adjusted to polycontextural logical formulas and their reduction techniques.

The sketched specifications of the structure of morphoCAs could be understood as a ‘blue print’ for a physical realization as a kind of new multi-layered chips for morphoCA machines (postToffoli).

UML process modelling

A first step to a UML modelling of morphoCAs could encourage to get a glimpse into the complex mechanism of morphoCAs.

Mediation Question:
“*Given two elementary CA (Boolean one - dimensional bi - infinite lattices of cells, the evolution of each cell being influenced by its direct neighbors) with known behaviors, what can be said on the behavior of the CA obtained by composing them (e.g., using a logic disjunction)?”*

https://scs.carleton.ca/sites/default/files/tr/TR-96-31.pdf

Again, Reduction and Mediation

Functional interpretation of ruleDM[{1, 2, 12, 4, 5}] in morphoCA$^{(3,3)}$.

```
{
 {0, 0, 0} -> 0,
 {1, 1, 1} -> 1,
 {2, 2, 2} -> 2,

 {0, 0, 1} -> 0,
 {0, 1, 0} -> 2,
 {1, 1, 0} -> 1,
 {1, 0, 1} -> 2,
 {0, 1, 1} -> 0,
 {1, 0, 0} -> 1,

 {1, 1, 2} -> 1,
 {1, 2, 1} -> 0,
 {1, 2, 2} -> 1,
 {2, 2, 1} -> 2,
 {2, 1, 2} -> 0,
 {2, 1, 1} -> 2,

 {0, 0, 2} -> 0,
 {0, 2, 0} -> 1,
```

```
{0, 2, 2} -> 0,
{2, 0, 0} -> 2,
 {2, 2, 0} -> 2,
{2, 0, 2} -> 1,

{0, 2, 1} -> 0,
{0, 1, 2} -> 0,
{1, 2, 0} -> 1,
{1, 0, 2} -> 1,
{2, 1, 0} -> 2,
{2, 0, 1} -> 2
}
```

transjunctional part of funct(ruleDM[{1, 2, 12, 4, 5}])

```
{0, 1, 0} -> 2,
{1, 0, 1} -> 2,
{1, 2, 1} -> 0,
{2, 1, 2} -> 0,
{2, 0, 2} -> 1,
{0, 2, 0} -> 1
```

mediating part of funct(ruleDM[{1, 2, 12, 4, 5}])

```
{0, 2, 1} -> 0,
{0, 1, 2} -> 0,
{1, 2, 0} -> 1,
{1, 0, 2} -> 1,
{2, 1, 0} -> 2,
{2, 0, 1} -> 2
```

Decomposition of ruleDM[{1, 2, 12, 4, 5}] into polylogical sub-systems

Junctional part with transjunctions

A[1]	A[2]	A[3]	op_1[A[1], A[2], A[3]]
T_1	T_1	T_1	T_1
T_1	T_1	F_1	T_1
T_1	F_1	T_1	F_2 →
T_1	F_1	F_1	T_1
F_1	T_1	T_1	F_1
F_1	T_1	F_1	F_2 ->
F_1	F_1	T_1	F_1
F_1	F_1	F_1	F_1

A[1]	A[2]	A[3]	op_3[A[1], A[2], A[3]]
T_3	T_3	T_3	T_3
T_3	T_3	F_3	T_3
T_3	F_3	T_3	F_1 →
T_3	F_3	F_3	T_3
F_3	T_3	T_3	F_3
F_3	T_3	F_3	T_1 ->
F_3	F_3	T_3	F_3
F_3	F_3	F_3	F_3

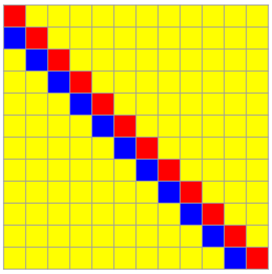
A[1]	A[2]	A[3]	op_2[A[1], A[2], A[3]]
T_2	T_2	T_2	T_2
T_2	T_2	F_2	T_2
T_2	F_2	T_2	T_1 ->
T_2	F_2	F_2	T_2
F_2	T_2	T_2	F_2
F_2	T_2	F_2	F_1 ->
F_2	F_2	T_2	F_2
F_2	F_2	F_2	F_2

mediating part

A[1]	A[2]	A[3]	op_2[A[1], A[2], A[3]]
$T_{1,3}$	$F_{2,3}$	F_1 T_2	$T_{1,3}$
$T_{1,3}$	F_1 T_2	$F_{2,3}$	$T_{1,3}$
F_1 T_2	$F_{2,3}$	$T_{1,3}$	F_1 T_2
F_1 T_2	$T_{1,3}$	$F_{2,3}$	F_1 T_2
$F_{2,3}$	F_1 T_2	$T_{1,3}$	$F_{2,3}$
$F_{2,3}$	$T_{1,3}$	F_1 T_2	$F_{2,3}$

Cellular automaton of ruleDM[{1, 2, 12, 4, 5}] in morphoCA$^{(3,3)}$

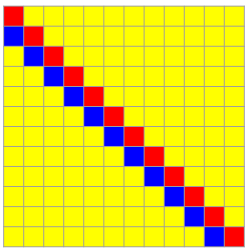
```
ArrayPlot[CellularAutomaton[
  {
   {0, 0, 0} → 0, {1, 1, 1} → 1, {2, 2, 2} → 2,  (* [1] *)

   {0, 0, 1} → 0, {0, 0, 2} → 0, {1, 1, 0} → 1,
   {1, 1, 2} → 1, {2, 2, 0} → 2, {2, 2, 1} → 2,  (* [2] *)

   {0, 2, 0} → 1, {1, 0, 1} → 2, {1, 2, 1} → 0,
   {0, 1, 0} → 2, {2, 1, 2} → 0, {2, 0, 2} → 1,  (* [12] *)

   {0, 1, 1} → 0, {0, 2, 2} → 0, {1, 0, 0} → 1,
   {1, 2, 2} → 1, {2, 0, 0} → 2, {2, 1, 1} → 2,  (* [4] *)

   {0, 2, 1} → 0, {0, 1, 2} → 0, {1, 2, 0} → 1,
   {1, 0, 2} → 1, {2, 1, 0} → 2, {2, 0, 1} → 2}, (* [5] *)

  {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green},
 Mesh → True, ImageSize → 300]
```



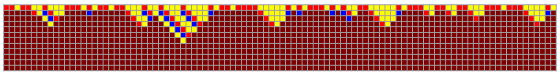
Functional reductions

Example : ruleDM[{1, 2, 12, 4, 5}]

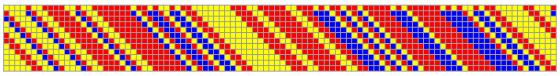
```
ArrayPlot[CellularAutomaton[
  {
    {0, 0, 0} -> 0,  (* junctional *)
    {0, 0, 1} -> 0,
    {1, 0, 0} -> 1,
    {0, 0, 2} -> 0,
    {0, 1, 0} -> 2, (* transjunctional *)
    {0, 2, 1} -> 0, (* mediating part *)
    {2, 1, 0} -> 2
  },
  {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue},
 Mesh → True, ImageSize → 300]
```



Random seeds

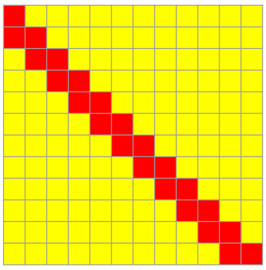


Without reduction, the random seeds are completely defining the morphoCA.



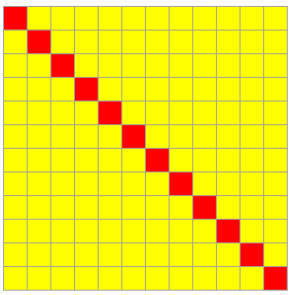
ruleDM[{1, 2, 8, 4, 5}] : parallel

```
ArrayPlot[CellularAutomaton[
   {
     {0, 0, 0} -> 0,
     {1, 0, 0} -> 1,
     {1, 1, 0} -> 1,
     {0, 1, 1} -> 0,
     {0, 1, 0} -> 1,
     {0, 0, 1} -> 0
   },
   {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green},
 Mesh → True, ImageSize → 300]
```

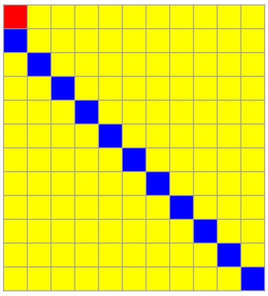


Functional reduction of the morphic rules

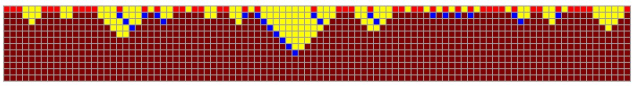
```
ArrayPlot[CellularAutomaton[
   {
     {0, 0, 0} -> 0,
     {1, 0, 0} -> 1,
     {0, 1, 0} -> 0,
     {0, 0, 1} -> 0
   },
   {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green},
 Mesh → True, ImageSize → 300]
```



```
ArrayPlot[CellularAutomaton[
  {
   {0, 0, 0} → 0,
   {0, 0, 1} → 0,
   {1, 0, 0} → 0,

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

   {0, 1, 0} → 2
  },
  {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue},
 Mesh → True, ImageSize → 300]
```



```
ArrayPlot[CellularAutomaton[
  {
   {0, 0, 0} → 0,
   {0, 0, 1} → 0,
   {1, 0, 0} → 0,

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

   {0, 1, 0} → 2
  },
  RandomInteger[1, 100], 11],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue},
 Mesh → True, ImageSize → 300]
```



Mediation of CA_1= (junctional) and CA_2 = (junctional + transjunctional) with functional reduction: final result is CA_3= (CA_1, CA_2, mediating part). Additionally, the elimination of {0,1,0} of $CA_1$and {0,2,0}, {2,0,0} of CA_2 happens.

Is CA_2 a genuine CA like CA_1?

(NKS page : 886 R)

Super-additivity of the mediation of $CA_1$and CA_2

Interchangeability of a 3 – contextural category with composition and mediation ($\coprod$)

$\mathcal{U}^{(3)} = (\mathcal{U}_1 \coprod_{1.2} \mathcal{U}_2) \coprod_{1.2,3} \mathcal{U}_3$
$(\mathcal{U}_1 \cap_{1.2} \mathcal{U}_2) \cap_{1.2,3} \mathcal{U}_3 = \emptyset :$
$\mathcal{U}_i = \{f_i, g_i\}, \; i = 1, 2, 3$

$$\begin{pmatrix} g_1 & - & g_3 \\ f_1 & g_2 & - \\ - & f_2 & f_3 \end{pmatrix} :$$

$$\begin{pmatrix} \begin{pmatrix} (f_1 \circ_{1.0,0} g_1) \\ \coprod_{1.2\ \ .0} \\ (f_2 \circ_{0.2,0} g_2) \end{pmatrix} \\ \coprod_{1.2,3} \\ (f_3 \circ_{0.0,3} g_3) \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} f_1 \\ \coprod_{1.2,0} \\ f_2 \end{pmatrix} \\ \coprod_{1.2,3} \\ f_3 \end{pmatrix} \circ_1 \circ_2 \circ_3 \begin{pmatrix} \begin{pmatrix} g_1 \\ \coprod_{1.2,0} \\ g_2 \end{pmatrix} \\ \coprod_{1.2,3} \\ g_3 \end{pmatrix}$$

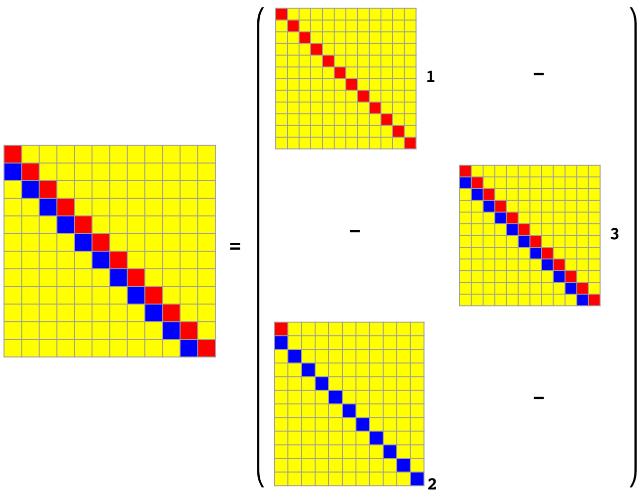
$CA_i = (f, g)_i :$

$(CA_1 \amalg_{1.2,0} CA_2) \Longrightarrow (CA_1 \amalg_{1.2,0} CA_2) \amalg_{1.2,3} CA_3$

$$(CA_1 \amalg_{1.2,0} CA_2) \amalg_{1.2,3} CA_3 = \begin{pmatrix} CA_1 & - \\ - & CA_3 \\ CA_2 & - \end{pmatrix}$$

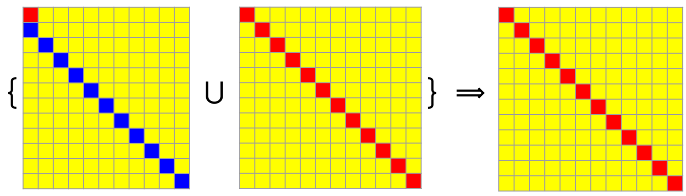
Reduction and mediation

```
Union :      (CA 1 ⋃ CA2)        = CA 1 (* seriel *)
Mediation :  (CA 1 ∐ 1.2,0 CA2)  = CA 3 (* parallel *)
```



Mediation : $CA^{(3,3)} = (CA_1 \amalg_{1.2,0} CA_2) \amalg_{1.2,3} CA_3$

Union of CA_1 and CA_2



(* **seriel** *)

Union : $(CA_1 \bigcup CA_2) = CA_1$

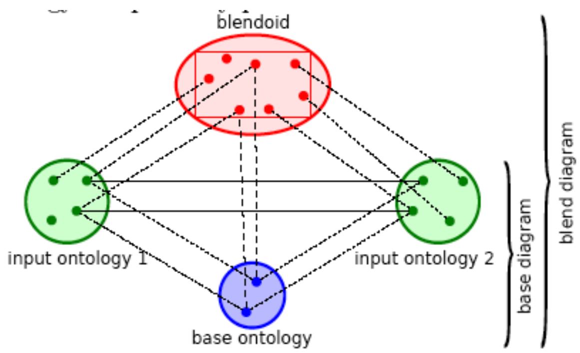
Yellow represents the background system of both, the blue and the red sub-system.
In the terms 'blending theory' (Goguen), the yellow system is the *'base ontology'*, while the red and the blue systems are the *'input ontologies'* of the process of blending, resulting in the mediation, i.e. the 'blendoid', as the interaction of the red and the blue system on the background of yellow system.

It shouldn't be difficult to understand that the sketched method of separation and mediation works for more complex cases too.

Super-additivity and concept blending

One reason while blending is not an immanent feature of logic, semiotics and computation might be the fact that it happens only on the level of semantics. Like the term "*houseboat*" and "*boathouse*", they are syntactically not produced by blending but by concatenation. The meaning of the composits is achieved by blending, and there is no simple concatenation of the meanings "boat" and "house" to "*houseboat*" and "*boathouse*" and super-additively to "*amphibious* vehicle". Both meaning of "boat" and "house" are *per se* autonomous.

> *"A classic example for this is the blending of the concepts house and boat, yielding as most straightforward blends the concepts of a houseboat and a boathouse, but also an amphibious vehicle."* (Kutz, Hois)



http : // memristors.memristics.com/Dissemination %20 as %20 Blending/Dissemination %20 as %20 Blending.html

http : // osl.cs.illinois.edu/media/papers/tosic - 2005 - parallel_vs _sequential _threshold _cellular _automata.pdf

"We shall illustrate the concept of sequential interleaving semantics of concurrency with a simple exercise.
Let' s consider the following question from a sophomore parallel programming class :
Find an example of two instructions such that, when executed in parallel, they give a result not obtainable from any corresponding sequential execution sequence.

A possible answer : Assume x = 0 initially and consider the following two programs

x ← x + 1; x ← x + 1
vs.
x ← x + 1 || x ← x + 1

where " || " stands for the parallel, and ";" for the sequential composition of instructions or programs, respectively.
Sequentially, one always gets the same answer : x = 2.
In parallel (when the two assignment operations are executed synchronously), however, one gets x = 1. It appears, therefore, that no sequential ordering of operations can reproduce parallel computation - at least not at the granularity level of high - level instructions as above.

Indeed, if we informally define Φ(P) to be the set of possible behaviors of program P, then the example above only shows that, for S1 = S2 = (x←x+1),

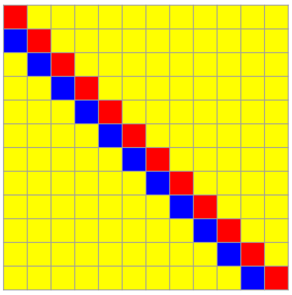
$\Phi(S1||S2) \nsubseteq \Phi(S1;S2) \cup \Phi(S2;S1)$ (1)

However, it turns out that, in this particular example, it indeed is the case that

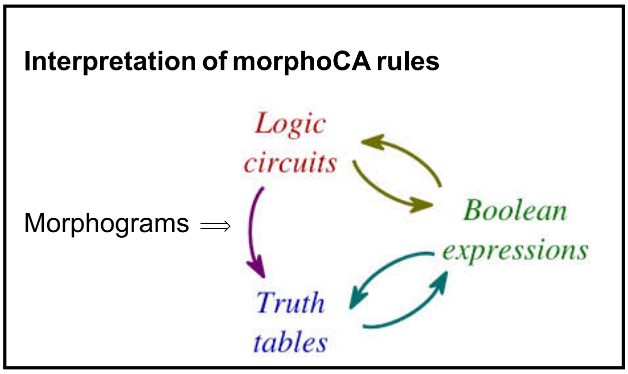
$\Phi(S1||S2) \subseteq \Phi(S1;S2) \cup \Phi(S2;S1) \cup \Phi(S1) \cup \Phi(S2)$ (2)

and no finer granularity is necessary to model Φ(S1||S2), *assuming that, in some of the sequential interleavings, we allow certain instructions not to be executed at all."* (Tosić)

Φ(S1||S2) ::
Φ(S1;S2) ∪ Φ(S2;S1) ∪ Φ(S1) ∪ Φ(S2) :
Φ(S1;S2) ∪ Φ(S2;S1) : double serial composition,
Φ(S1) ∪ Φ(S2)   : super-additivity



Minimization of morphoCA functions



http : // www.toves.org/books/logic/

MG - composition -> value-interpretation -> functional reduction -> MG-interpretation, with such formal reductions of morphoCAs a context of technical realizations seems to be sketched.

MG – reduction scheme

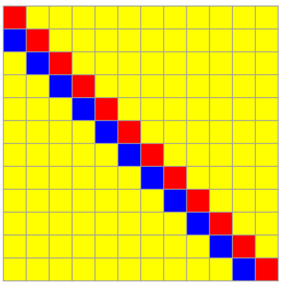
MG – composition ⟶ value – interpretation
↓
MG – reduction ⟵ functional reduction

The proposed functional reduction of morphoCA is not changing its morphogrammatic base, i.e. its morphograms, but the symbolic interpretation of the morphograms only. Therefore, functionally reduced morphoCAs are still morphoCAs, and nothing else.

ruleDM[{1, 2, 12, 4, 5}]

```
ruleDM[{1, 2, 12, 4, 5}]
```

```
ArrayPlot[CellularAutomaton[
  {
   {0, 0, 0} → 0,   (*[1]*)
   {0, 0, 1} -> 0,
 (*   {1, 1, 0} -> 1,
   {1, 0, 1} -> 2,*)
   {1, 0, 0} -> 1,
   {0, 1, 0} -> 2,
  (* {0, 1, 1} -> 0,
   {1,1,1} → 1,*)

   (*   {1, 1, 2} -> 1,
     {2, 2, 1} -> 2,
    {1, 2, 1} -> 0, (*[12]*)
        {2, 1, 2} -> 0,
    {1, 2, 2} -> 1, (*[4]*)
   {2, 1, 1} -> 2,*)

   {0, 0, 2} -> 0,  (*[2]*)
  (*   {0, 2, 0} -> 1,
   {0, 2, 2} -> 0,
    {2, 0, 0} -> 2,
    {2, 0, 2} -> 1,*)
  (*  {2, 2, 0} -> 2, *)

   {0, 2, 1} -> 0, (*[5]*)
 (*       {0, 1, 2} -> 0,
         {1, 2, 0} -> 1,
         {1, 0, 2} -> 1,
    {2, 0, 1} -> 2,*)
   {2, 1, 0} -> 2
  (*  {2,2,2} → 2*)
  },
  {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green},
 Mesh → True, ImageSize → 300]
```



The decomposition of ruleDM[{1, 2, 12, 4, 5}] into polylogical sub-systems might be generalized towards the following scheme.

$$(CA_1 \amalg_{1.2,0} CA_2) \amalg_{1.2,3} CA_3 = \begin{pmatrix} CA_1 & - \\ - & CA_3 \\ CA_2 & - \end{pmatrix}$$

Full rule scheme

$$\left(\left(\begin{pmatrix} \begin{pmatrix} < A[1] \mid A[2] \mid A[3] >_1 \\ \amalg_{1.2} \\ < A[1] \mid A[2] \mid A[3] >_2 \end{pmatrix} \\ \amalg_{1.2,3} \\ < A[1] \mid A[2] \mid A[3] >_3 \end{pmatrix} \backslash \begin{pmatrix} \begin{pmatrix} < A[t, f] \mid A[f, t] \mid A[t, f] >_1 \\ \amalg_{1.2} \\ < A[t, f] \mid A[f, t] \mid A[t, f] >_2 \end{pmatrix} \\ \amalg_{1.2,3} \\ < A[t, f] \mid A[f, t] \mid A[t, f] >_3 \end{pmatrix}\right)\right)$$

$$\left(\left(\left(\begin{matrix}\langle A[t,f] \mid A[f,t] \mid A[t,f]\rangle_1 \\ \amalg_{1.2} \\ \langle A[t,f] \mid A[f,t] \mid A[t,f]\rangle_2\end{matrix}\right) \\ \amalg_{1.2,3} \\ \langle A[t,f] \mid A[f,t] \mid A[t,f]\rangle_3\right) \left(\left(\begin{matrix}\langle A[1]_1 \mid A[2]_2 \mid A[3]_3\rangle_1 \\ \amalg_{1.2} \\ \langle A[1]_1 \mid A[2]_2 \mid A[3]_3\rangle_2\end{matrix}\right) \\ \amalg_{1.2,3} \\ \langle A[1]_1 \mid A[2]_2 \mid A[3]_3\rangle_3\right)\right)$$

Frame scheme

$$\left(\left(\left(\begin{matrix}\langle A[1] \mid A[2] \mid A[3]\rangle_1 \\ \amalg_{1.2} \\ \langle A[1] \mid A[2] \mid A[3]\rangle_2\end{matrix}\right) \\ \amalg_{1.2,3} \\ \langle A[1] \mid A[2] \mid A[3]\rangle_3\right) \backslash \left(\left(\begin{matrix}\langle A[t,f] \mid A[f,t] \mid A[t,f]\rangle_1 \\ \amalg_{1.2} \\ \langle A[t,f] \mid A[f,t] \mid A[t,f]\rangle_2\end{matrix}\right) \\ \amalg_{1.2,3} \\ \langle A[t,f] \mid A[f,t] \mid A[t,f]\rangle_3\right)\right)$$

Transjunctional part **mediative part**

$$\left(\left(\left(\begin{matrix}\langle A[t,f] \mid A[f,t] \mid A[t,f]\rangle_1 \\ \amalg_{1.2} \\ \langle A[t,f] \mid A[f,t] \mid A[t,f]\rangle_2\end{matrix}\right) \\ \amalg_{1.2,3} \\ \langle A[t,f] \mid A[f,t] \mid A[t,f]\rangle_3\right) \left(\left(\begin{matrix}\langle A[1]_1 \mid A[2]_2 \mid A[3]_3\rangle_1 \\ \amalg_{1.2} \\ \langle A[1]_1 \mid A[2]_2 \mid A[3]_3\rangle_2\end{matrix}\right) \\ \amalg_{1.2,3} \\ \langle A[1]_1 \mid A[2]_2 \mid A[3]_3\rangle_3\right)\right)$$

As an example ruleDM[{1, 2, 12, 4, 5}] might be decomposed:

Transjunctional part:

< A[1]	A[2]	A[3] >₁
(0	0	0) → 0
(0	0	1) → 0
(1	0	0) → 1

$\amalg_{1.2}$

< A[1]	A[2]	A[3] >₂
(0	0	2) → 0

$\amalg_{1.2,3}$

$\langle A[1] \mid A[2] \mid A[3]\rangle_3$

$\backslash$

$$\left(\left(\begin{matrix}\langle A[t,f] \mid A[f,t] \mid A[t,f]\rangle_1 \\ \amalg_{1.2} \\ \langle A[t,f] \mid A[f,t] \mid A[t,f]\rangle_2\end{matrix}\right) \\ \amalg_{1.2,3} \\ \langle A[t,f] \mid A[f,t] \mid A[t,f]\rangle_3\right)$$

Transjunctional part **mediative part**

Transjunctional part:

< A[t]	A[f]	A[t] >₁
(0	1	0) → 2

$\amalg_{1.2}$

$\langle A[t,f] \mid A[f,t] \mid A[t,f]\rangle_2$

$\amalg_{1.2,3}$

$\langle A[t,f] \mid A[f,t] \mid A[t,f]\rangle_3$

Mediative part:

< A[t]₁	A[f]₂	A[f]₃ >₁
(0	2	1) → 0
(2	1	0) → 2

$\amalg_{1.2}$

$\langle A[f]_1 \mid A[t]_2 \mid A[3]_3\rangle_2$

$\amalg_{1.2,3}$

$\langle A[1]_1 \mid A[2]_2 \mid A[3]_3\rangle_3$

The decomposed scheme for ruleDM[{1, 2, 12, 4, 5}] resumed

```
ruleDM[{1, 2, 12, 4, 5}] =
```

< A[1]	A[2]	A[3] >₁
(x	x	x) → x
(x	x	y) → x
(y	x	x) → y

$\amalg_{1.2}$ | _

< A[1]	A[2]	A[3] >₂
(x	x	z) → x

$\square_{[1;(1,2,3)_1]}$ | _

< A[t]	A[f]	A[t] >₁
(x	y	x) → z

< A[t]₁	A[f]₂	A[f]₃ >₁
(x	z	y) → x
(z	y	x) → z

Reduced example of ruleDM[{1, 2, 12, 4, 5}]

```
ruleDM[{1, 2, 12, 4, 5}] =
```

< A[1]	A[2]	A[3] | A[0] >₁
(0	0	0) → 0
(0	0	1) → 0
(1	0	0) → 1

$\amalg_{1.2}$ –

<

A[1]	A[2]	A[3] | A[0]
(0	0	2) → 0

$>_2$

$\square_{[1;(1,2,3)_1]}$ –

(

< A[t]	A[f]	A[t] | A[0] >₁
(0	1	0) → 2

< A[t]₁	A[f]₂	A[f]₃ | A[0] >₁
(0	2	1) → 0
(2	1	0) → 2

)

Application of the reduced ruleDM[{1, 2, 12, 4, 5}]

```
ArrayPlot[CellularAutomaton[
  {
   {0, 0, 0} → 0,  (*[1]*)
   {0, 0, 1} -> 0, (*[2]*)
   {0, 0, 2} -> 0,
   {1, 0, 0} -> 1, (*[4]*)
   {0, 1, 0} -> 2, (*[12]*)
   {0, 2, 1} -> 0, (*[5]*)
   {2, 1, 0} -> 2
  },
  {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green},
 Mesh → True, ImageSize → 100]
```

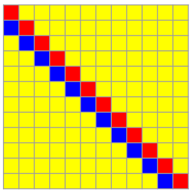


Table notation of the reduced ruleDM[{1, 2, 12, 4, 5}]

ruleDM[{1, 11, 3, 4, 5}] =

x / yz	00	01	02	10	20	11	12	21	22
0	0	0	0	2	1	1	0	0	0
1	1	2	1	1	1	1	1	0	1
2	2	2	1	2	2	2	0	2	2

Elimination of the reduced values (in blue)

ruleDM[{1, 11, 3, 4, 5}] =

x / yz	00	01	02	10	21
0	0	0	0	2	0
1	1	□	□	□	□
2	□	□	□	2	□

ruleDM[{1, 11, 3, 4, 5}] =

x / yz	00	01	02	10	21
0	t1, 3	t1	t3	v1	v3
1	f1	□	□	□	□
2	□	□	□	v2	□

x / yz	00	01	11	10
0	0	0	□	t
1	1	□	□	□

+ (0, 10) → 2

x / yz	11	12	22	21
1	□	□	□	t
2	□	□	□	□

+ (2, 10) → 2

x / yz	00	02	22	20
0	0	0	□	□
2	□	□	□	□

A different notation

A / B	0	1	2	0	1	2	0	1	2
0	0	2	1	1	1	0	2	0	0
1	1	1	0	2	1	0	1	1	1
2	1	0	2	2	1	2	2	0	2
C	0	–	–	1	–	–	2	–	–

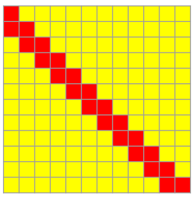
val (ruleDM[{1, 2, 12, 4, 5}]) = 1 :

A / B	0	1	2	0	1	2	0	1	2
0	□	□	1	1	1	□	□	□	□
1	1	1	□	□	1	□	1	1	1
2	1	□	□	□	1	□	□	□	□
C	0	–	–	1	–	–	2	–	–

val(ruleDM[{1,2,12,4,5}]) = 1 iff
A0B2C0 + A0B0C1 + A0B1C1 +
A1B0C0 + A1B1C0 + A1B1C1+ A1B0C2 + A1B1C2 + A1B2C2 +
A2B0C0 + A2B1C1

ruleDCl[{1, 2, 8, 4}]

```
ArrayPlot[CellularAutomaton[
  {
   {0, 0, 0} -> 0,
   {1, 0, 0} -> 1,
   {1, 1, 0} -> 1,
   {0, 1, 1} -> 0,
   {0, 1, 0} -> 1,
   {0, 0, 1} -> 0
  },
  {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green},
 Mesh → True, ImageSize → 300]
```



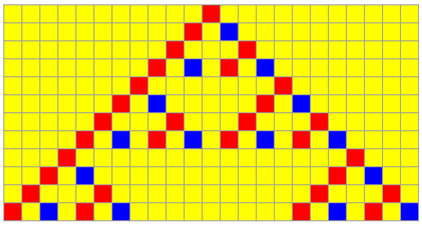
ruleDCl[{1, 2, 8, 4}] =

x / yz	00	01	11	10
0	0	0	0	1
1	1	□	1	1

Reducible and non-reducible functional representations

Example : ruleDM[{1, 7, 3, 13, 10}]

```
ArrayPlot[CellularAutomaton[
   ruleDM[{1, 7, 3, 13, 10}], {{1}, 0}, 11],
  ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green},
  Mesh → True, ImageSize → 300]
```

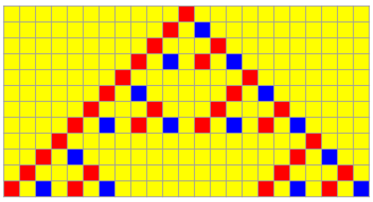


```
ruleDM[{1, 7, 3, 13, 10}]
```

```
ruleDM[{1, 7, 3, 13, 10}] =

 [1]  = {x, x, x} → x,
 [7]  = {x, x, y} → y,
 [3]  = {x, y, x} → x,
[13]  = {x, y, y} → z, z ≠ x, y
[10]  = {x, y, z} → y,
        x, y, z ∈ {0, 1, 2, 3}.
```

```
ruleDM[{1, 7, 3, 13, 10}] =

 <  A[1] | A[2] | A[3]      >1
    (x   |  x   |  x)  → x
 <  A[1] | A[2] | A[3]      >7
    (x   |  x   |  y)  → y
 <  A[1] | A[2] | A[3]      >3
    (x   |  y   |  x)  → x
          ⨿1.2
 <  A[1] | A[2] | A[3]      >13
    (x   |  y   |  y)  → z
          ⨿1.2,3
 <  A[1] | A[2] | A[3]      >10
    (x   |  y   |  z)  → y
```

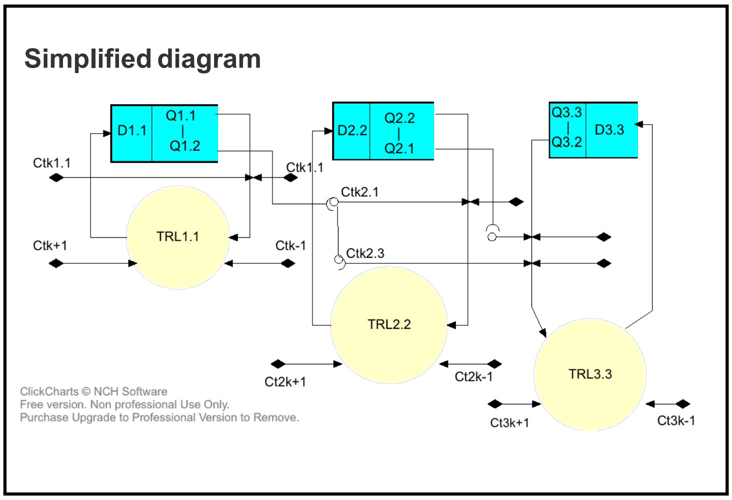
Functional reduction of ruleDM[{1,7,3,13,10}]



```
ruleDM[{1, 7, 3, 13, 10}] =

 <  < A[1] | A[2] | A[3] | A[0] >     >1    |   –
      (0   |  0   |  0)  → 0                |
 -------------------------------------------+-----------------------------------------------
    < A[1] | A[2] | A[3] | A[0] >           |   –
 <    (0   |  0   |  1)  → 1              >7|
      (0   |  0   |  2)  → 2                |
    < A[1] | A[2] | A[3] | A[0] >           |
 <    (0   |  1   |  0)  → 0              >3|
      (0   |  2   |  0)  → 0                |
 -------------------------------------------+-----------------------------------------------
    < A[t] | A[f] | A[t] | A[0] >           |   < A[t]1 | A[f]2 | A[f]3 | A[0] >
 <    (1   |  0   |  0)  → 2             >13| <   (2    |  0    |  1)  → 0            >10
      (2   |  0   |  0)  → 1                |     (1    |  0    |  2)  → 0
```

ruleDM[{1, 11, 3, 4, 5}] =

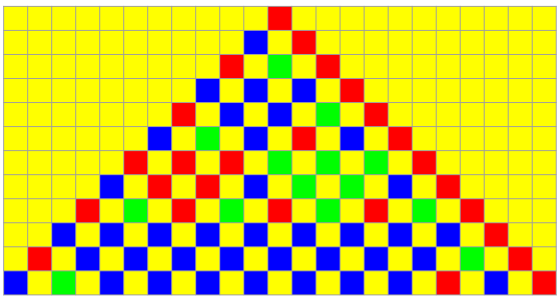
x / yz	00	01	02	10	20
0	0	1	2	0	0
1	2	□	0	□	□
2	1	0	□	□	□



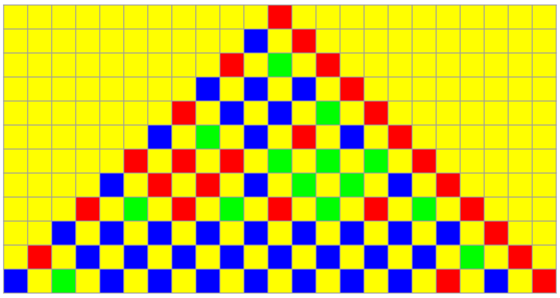
Functional reduction of ruleDM[{1,11,3,4,15}]

```
ruleDM[{1, 11, 3, 4, 15}]
```

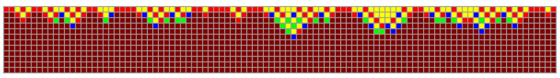
```
ArrayPlot[CellularAutomaton[ ruleDM[{1, 11, 3, 4, 15}],
   {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green},
 Mesh → True, ImageSize → 300]
```



```
ArrayPlot[CellularAutomaton[
  {
   {0, 0, 0} → 0,
   {0, 0, 1} → 2,
   {0, 0, 2} → 1,

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

   {1, 0, 0} → 1,

   {0, 1, 2} → 3,
   {2, 0, 1} → 3,
   {1, 0, 2} → 3,
   {1, 0, 3} → 2,
   {3, 0, 1} → 2,
   {3, 0, 2} → 1,
   {2, 0, 3} → 1
  },
  {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green},
 Mesh → True, ImageSize → 300]
```

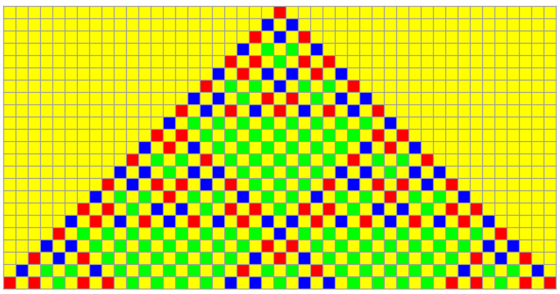


Random seed



Functional reduction of ruleDM[{1,11,3,13,15}]

```
ArrayPlot[CellularAutomaton[
   ruleDM[{1, 11, 3, 13, 15}],
  {{1}, 0}, 22],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green},
 Mesh → True, ImageSize → 300]
```



```
ArrayPlot[CellularAutomaton[
  {
   {0, 0, 0} → 0,   (* [1] *)

   {0, 0, 1} → 2, {0, 0, 2} → 1,  (* [11] *)

   {0, 1, 0} → 0,  (* [3] *)
   {1, 0, 1} → 1,
   {0, 2, 0} → 0,
   {2, 0, 2} → 2,
   {3, 0, 3} → 3,
   {0, 3, 0} → 0,

   {1, 0, 0} → 2,  (* [13] *)
   {2, 0, 0} → 1,

   {1, 0, 2} → 3,  (* [15] *)
   {1, 0, 3} → 2,
   {2, 0, 1} → 3,
   {2, 0, 3} → 1,
   {3, 0, 1} → 2,
   {3, 0, 2} → 1
  },
  {{1}, 0}, 22],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green},
 Mesh → True, ImageSize → 300]
```

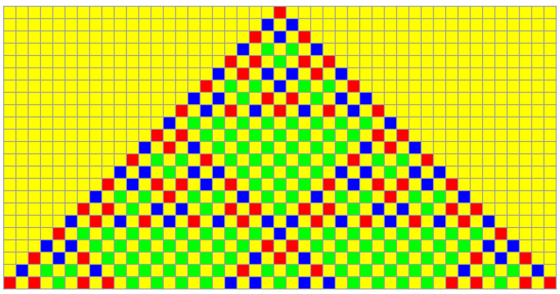
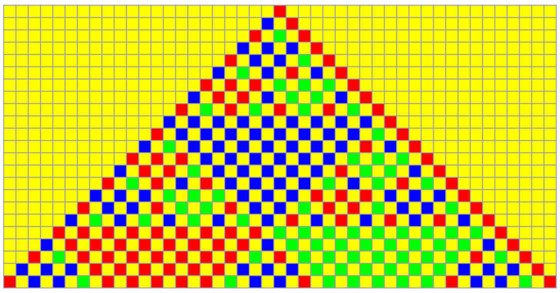


table (reduction (ruleDM[{1, 11, 3, 4, 15}])) =

x / yz	00	01	02	03	10	20	30
0	0	2	1	□	0	0	0
1	2	1	3	2	□	□	□
2	1	3	2	□	□	□	□
3	□	2	1	3	□	□	□

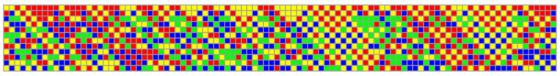
Functional reduction of ruleDM[{1,11,3,4,15}]

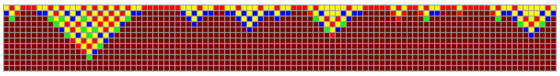
```
ArrayPlot[CellularAutomaton[
   ruleDM[{1, 11, 3, 4, 15}],
   {{1}, 0}, 22],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green},
 Mesh → True, ImageSize → 300]
```



Random seed

```
ArrayPlot[CellularAutomaton[ ruleDM[{1, 11, 3, 4, 15}],
   RandomInteger[1, 100], 11],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green},
 Mesh → True, ImageSize → 300]
```

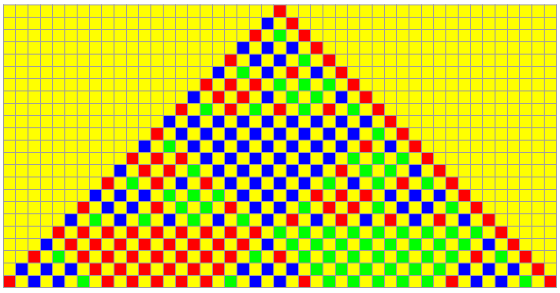




ruleDM[{1, 11, 3, 4, 15}]

Reduction : ruleDM[{1, 11, 3, 4, 15}]

```
ArrayPlot[CellularAutomaton[
  {
   {0, 0, 0} → 0,

   {0, 0, 1} → 2,
   {0, 0, 2} → 1,

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

   {1, 0, 0} → 1,

   {1, 0, 2} → 3,
   {1, 0, 3} → 2,
   {2, 0, 3} → 1,
   {2, 0, 1} → 3,
   {3, 0, 2} → 1,
   {3, 0, 1} → 2
  },
  {{1}, 0}, 22],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green},
 Mesh → True, ImageSize → 300]
```



ruleDM[{1, 11, 3, 4, 15}] is reducible to table:

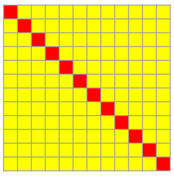
table (reduction (ruleDM[{1, 11, 3, 4, 15}])) =

x / yz	00	01	02	03	10	20	30
0	0	2	1	□	0	0	0
1	1	1	3	2	□	□	□
2	□	3	2	1	□	□	□
3	□	2	1	3	□	□	□

Example ruleDCl[{1, 2, 3, 4}]

`ruleDCl[{1, 2, 3, 4}]`

```
ArrayPlot[CellularAutomaton[
   ruleDCl[{1, 2, 3, 4}],
   {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green},
 Mesh → True, ImageSize → 300]
```



```
ruleDCl[{1, 2, 3, 4}] =

 [1] = {x, x, x} → x,
 [2] = {x, x, y} → x,
 [3] = {x, y, x} → x,
 [4] = {x, y, y} → x,
       x, y ∈ {0, 1, 2, 3}.
```

```
              ruleDCl[{1, 2, 3, 4}] =
             <  A[1] | A[2] | A[3] | A[0]  >1
                (x   |  x   |  x)  →  x
             <  A[1] | A[2] | A[3] | A[0]  >2
                (x   |  x   |  y)  →  x
II1.2,3,4    <  A[1] | A[2] | A[3] | A[0]  >3
                (x   |  y   |  x)  →  y
             <  A[1] | A[2] | A[3] | A[0]  >4
                (x   |  y   |  y)  →  x
```

Reduction

ruleDCl[{1, 2, 3, 4}] ⟹

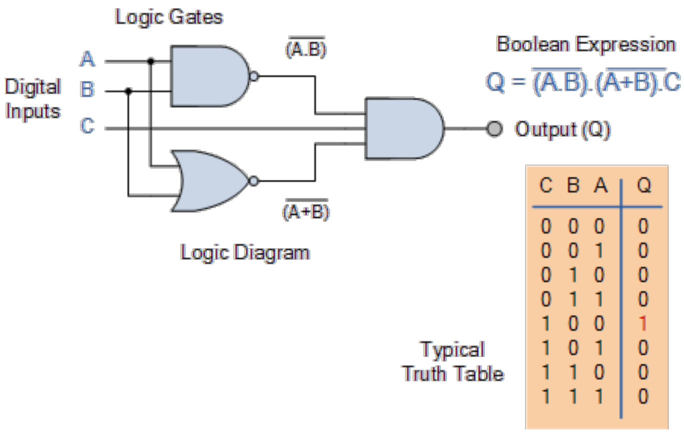
x / yz	00	01	10
0	0	0	0
1	1	-	-

Reductions by Karnaugh maps

> *“Karnaugh maps reduce logic functions more quickly and easily compared to Boolean algebra. By reduce we mean simplify, reducing the number of gates and inputs. We like to simplify logic to a lowest cost form to save costs by elimination of components. We define lowest cost as being the lowest number of gates with the lowest number of inputs per gate.”*

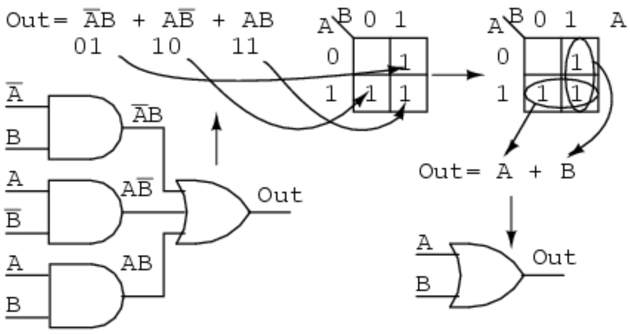
http : // www.allaboutcircuits.com/textbook/digital/chpt - 8/karnaugh - maps - truth - tables - boolean - expressions/

x' y' z + y z' + x y



http : // www.electronics - tutorials.ws/combination/comb_ 1. html

Further example

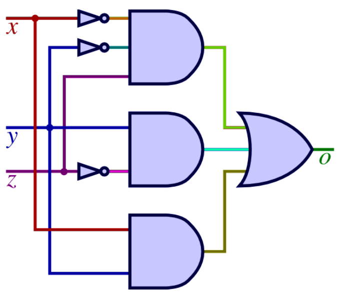


http : // www.allaboutcircuits.com/textbook/digital/chpt - 8/karnaugh - maps - truth - tables - boolean - expressions/

http : // www.ijser.org/researchpaper %5 CFormulation - and - Design - of - Useful - Logic - Gates - Using - Quater-nary - Algebra.pdf

Towards Karnaugh maps

x / yz	00	01	11	10
0	0	1	0	1
1	0	0	1	1

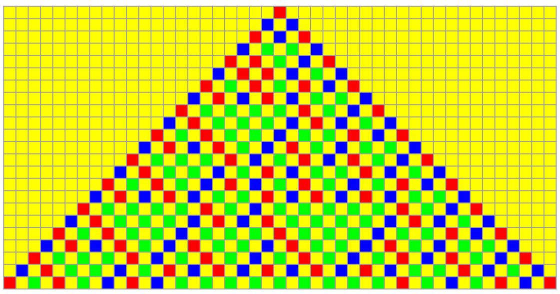


http : // www.toves.org/books/logic/

Examples for morphoCA$^{(3,3)}$

Example1: ruleDM[{1, 11, 3, 13, 15}]

```
ArrayPlot[CellularAutomaton[
  {
   {0, 0, 0} → 0,

   {0, 0, 1} → 2,
   {0, 0, 2} → 1,

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

   {1, 0, 0} → 2,
   {2, 0, 0} → 1,

   {1, 0, 2} → 3,
   {1, 0, 3} → 1,
   {2, 0, 1} → 3,
   {2, 0, 3} → 1,
   {3, 0, 1} → 2,
   {3, 0, 2} → 2
  },
  {{1}, 0}, 22],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green},
 Mesh → True, ImageSize → 300]
```



ruleDM[{1, 11, 3, 13, 15}] is minimized manually to the table :

ruleDM[{1, 11, 3, 13, 15}] ⟹

x / yz	00	01	10	02	03	20	30
0	**0**	2	**0**	1	□	0	0
1	2	**1**	□	3	1	□	□
2	1	3	□	2	1	□	□
3	□	2	□	2	3	□	□

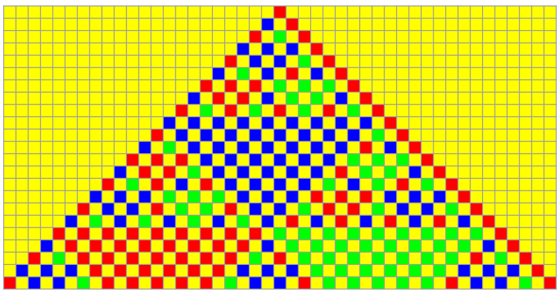
Example2: ruleDM[{1, 11, 3, 4, 15}]

Equality test: ruleDM[{1,11,3,4,15}] equal reduct(ruleDM[{1,11,3,4,15}])

```
A == With[{dropZeros = # /. {x__, 0 ..} :> {x} &},
   MatrixForm[Map[Reverse, dropZeros[#] & /@ Map[Reverse, dropZeros[#] & /@
        CellularAutomaton[ ruleDM[{1, 11, 3, 4, 15}], {{1}, 0}, 11]]]]];
B == With[{dropZeros = # /. {x__, 0 ..} :> {x} &},
   MatrixForm[Map[Reverse, dropZeros[#] & /@ Map[Reverse, dropZeros[#] & /@
        CellularAutomaton[ {
          {0, 0, 0} → 0,

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

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

          {1, 0, 0} → 1,

          {1, 0, 2} → 3,
          {1, 0, 3} → 2,
          {2, 0, 1} → 3,
          {2, 0, 3} → 1,
          {3, 0, 1} → 2,
          {3, 0, 2} → 1},
         {{1}, 0}, 11]]]]];

         A == B

         True
```

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

```
ArrayPlot[CellularAutomaton[
  ruleDM[{1, 11, 3, 4, 15}],
  {{1}, 0}, 22],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green},
 Mesh → True, ImageSize → 300]
```



ruleDM[[{1, 11, 3, 4, 15}] is minimized manually to the table :

ruleDM[{1, 11, 3, 4, 15}]

x / yz	00	01	10	02	03	20	30
0	**0**	2	**0**	1	–	0	0
1	1	**1**	1	3	2	–	–
2	–	3	–	2	1	–	–
3	–	2	–	2	3	–	–

Not reduced table for ruleDM[{1, 11, 3, 4, 15}].

ruleDM[{1, 11, 3, 4, 15}]

ruleDM[{1, 11, 3, 4, 15}] =

x / yz	00	01	02	03	10	20	30	11	12	13	21	22	23	31	32	33
0	0	2	1	2	0	0	0	0	3	2	3	0	1	2	1	0
1	1	1	3	2	2	3	2	1	0	2	1	1	0	1	0	1
2	2	3	2	1	3	1	1	2	2	0	0	2	0	0	2	2
3	3	2	1	3	2	1	2	3	0	3	0	3	3	0	1	3

Irreducible rules

Irreducible rules are playing the same role for morphoCAs as the irreducible binary functions like NAND, XOR for binary reductions. With NAND or NOR, all other two-valued binary function are defined. Because they are not reducible they are used as elementary devies in electronic circuit consturctions.

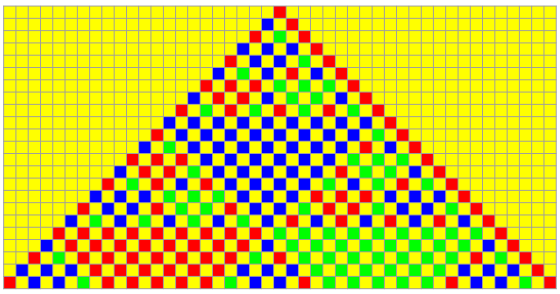
The question for morphic patterns arises: *How many irreducible patterns exist for morphoCA$^{(3,3)}$?*

Do they have the same property to define all other functions?

With *ruleDM[{1, 11, 8, 4, 15}]* non-reducible, its *dual* function might be irreducible too. (?)

```
ruleDM[{1, 11, 8, 4, 15}], yes
ruleDM[{1, 11, 3, 4, 15}], no
ruleDM[{1, 11, 8, 9, 15}]
ruleDM[{1, 11, 3, 9, 15}], no
ruleDM[{6, 11, 8, 4, 15}],
ruleDM[{6, 11, 3, 4, 15}]
ruleDM[{6, 11, 8, 9, 15}]
ruleDM[{6, 11, 3, 9, 15}]
```

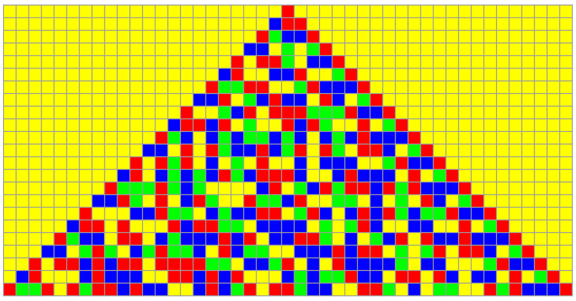
ruleDM[{1, 11, 3, 4, 15}]



reductions : {1, 1, 1} → 1, {2, 2, 2} → 2, {3, 3, 3} → 3

Example3 : ruleDM[{1, 11, 8, 4, 15}]

ruleDM[{1, 11, 8, 4, 15}]

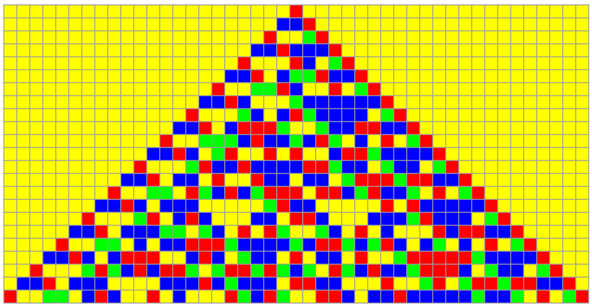


ruleDM[{1, 11, 8, 4, 15}] =

x / yz	00	01	02	03	10	20	30	11	12	13	21	22	23	31	32	33
0	0	2	1	2	1	2	3	0	3	2	3	0	1	2	1	0
1	1	0	3	2	2	3	2	1	1	2	2	1	0	3	0	1
2	2	3	2	1	3	1	1	2	1	0	0	2	0	0	3	2
3	3	2	1	0	2	1	2	3	0	1	0	3	2	0	1	3

ruleDM[{1, 11, 8, 4, 15}] can not be minimized without damaging the original result.

Example4 : ruleDM[{1, 11, 12, 4, 15}]



ruleDM[{1, 11, 12, 4, 15}] =

x / yz	00	01	02	03	10	20	30	11	12	13	21	22	23	31	32	33
0	0	2	1	2	2	1	2	0	3	2	3	0	1	2	1	0
1	1	2	3	2	2	3	2	1	1	2	0	1	0	2	0	1
2	2	3	1	1	3	1	1	2	0	0	0	2	0	0	1	2
3	3	2	1	1	2	1	2	3	0	1	0	3	1	2	1	3

ruleDM[{1, 11, 12, 4, 15}] can not be minimized without damaging the original result.

http : // www.isaet.org/images/extraimages/S1213024.pdf

http : // www.ijser.org/researchpaper %5 COptimization - of - Ternary - Combinational - System.pdf

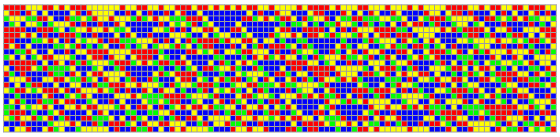
ruleDM[{1, 11, 12, 4, 15}] =

x / yz	00	01	02	03	10	20	30	11	12	13	21	22	23	31	32	33
0	0	2	1	2	2	1	2	0	3	2	3	0	1	2	1	0
1	1	2	3	2	2	3	2	1	1	2	0	1	0	2	0	1
2	2	3	1	1	3	1	1	2	0	0	0	2	0	0	1	2
3	3	2	1	1	2	1	2	3	0	1	0	3	1	2	1	3

ruleDM[{1, 11, 12, 4, 15}] =

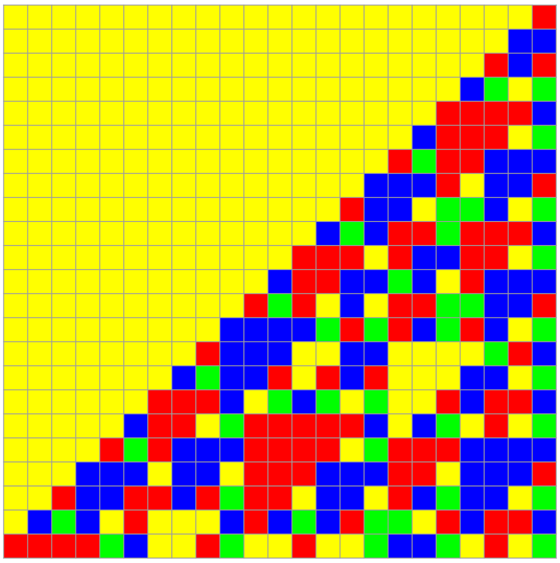
x / y	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
0	0	2	1	□	□	□	□	□	□	□	□	□	□	□	□	□
1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
2	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
3	2	□	□	□		□	□	□		□	□	□		□	□	□
z	0				1				2				3			

Random seed

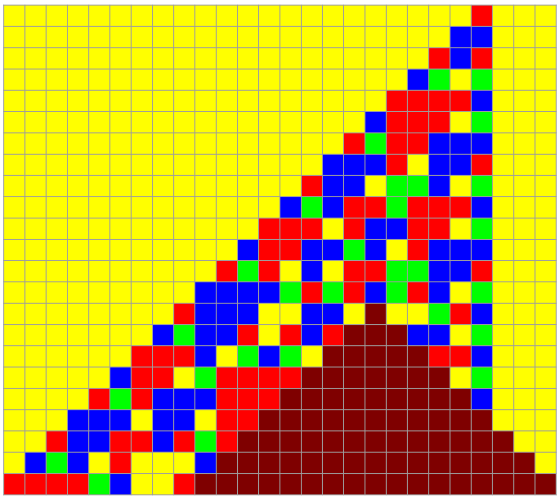


Example: ruleDM[{1, 11, 12, 9, 15}] : irreducible

```
ruleDM[{1, 11, 12, 9, 15}]
```



Without {1, 2, 3} -> 0:



ruleDM[{1, 11, 12, 4, 15}] =

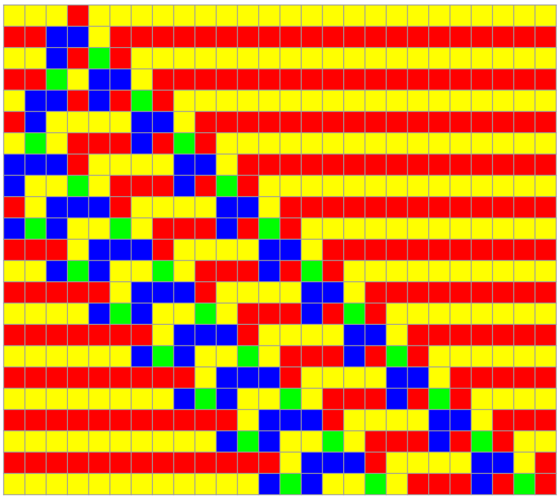
x / yz	00	01	02	03	10	20	30	11	12	13	21	22	23	31	32	33
0	0	2	1	2	2	1	2	1	3	2	3	2	1	2	1	3
1	0	2	3	2	2	3	2	1	1	2	0	2	0	2	0	3
2	0	3	1	1	3	1	1	1	0	0	0	2	0	0	1	3
3	0	2	1	1	2	1	2	1	0	1	0	2	1	2	1	3

ruleDM[{1, 11, 12, 9, 15}] =

x / y	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
0	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
2	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
3		□	□	□		□	□	□		□	□	□		□	□	□
z	0				1				2				3			

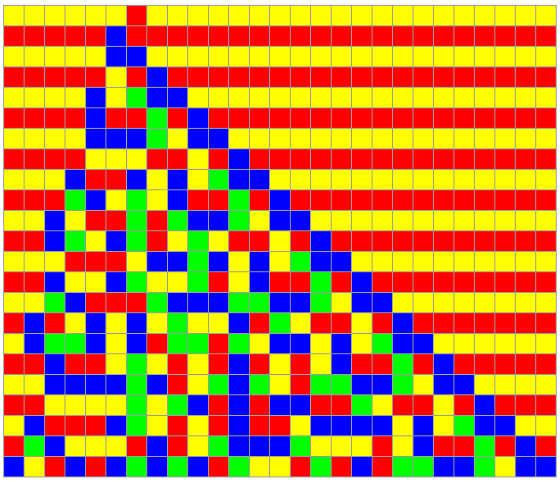
Reducible examples

ruleDM[{6, 11, 12, 9, 15}]



Reduct :
{2, 3, 1} → 0, {2, 2, 2} → 1,
{2, 2, 2} → 3, {3, 3, 3} → 1, {3, 3, 3} → 2,
{3, 3, 3} → 0, {3, 3, 2} → 1, {3, 3, 0} → 2,
{3, 3, 1} → 0, {2, 3, 3} → 3

ruleDM[{6, 11, 8, 4, 15}]

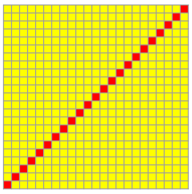


reductions : {3, 3, 3} → 1, {1, 1, 1} → 2

Mapping morphoCA rules onto computational diagrams

ruleDCl[{1, 2,3,4}]

```
ArrayPlot[CellularAutomaton[
  {{0, 0, 0} → 0,
   {0, 0, 1} → 1,
   {0, 1, 0} → 0,
   {1, 0, 0} → 0}, {{1}, 0}, 22],
 ColorRules -> {1 -> Red, 0 -> Yellow},
 Mesh → True, ImageSize → 100]
```



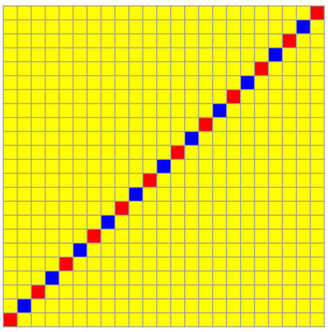
ruleDCl[{1, 2, 3, 4}]

x / yz	00	01	10
0	0	1	0
1	0	□	□

ruleDM[{1, 11, 3, 9, 15}]

ruleDM[{1, 11, 3, 9, 15}]

```
ArrayPlot[CellularAutomaton[
  {{0, 0, 0} → 0,
   {0, 0, 1} → 2, {0, 0, 2} → 1,
   {0, 1, 0} → 0, {0, 2, 0} → 0,
   {1, 0, 0} → 0, {2, 0, 0} → 0}, {{1}, 0}, 22],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue},
 Mesh → True, ImageSize → 300]
```



ruleDM[{1, 11, 3, 9, 15}]

x / yz	00	01	02	10	20
0	0	2	1	0	0
1	0	□	□	□	□
2	0	□	□	□	□

A0B0C0 + A1B0C0 + A2B0C0 + A0B0C1 + A0B0C2 + A0B1C0 + A0B2C0

A0B0C0 + A0B0C2 + A0B1C0 + A0B2C0 + A1B0C0 + A2B0C0 + A0B0C1

A0 (B0C0 + B0C2 + B1C0 + B2C0) + A1B0C0 + A2B0C0 + A0B0C1

A0B0C0 + A1B0C0 + A2B0C0 + A0B0C1 + A0B0C2 + A0B1C0 + A0B2C0

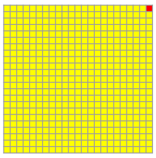
A0B0C0 + A0B0C2 + A0B1C0 + A0B2C0 + A1B0C0 + A2B0C0 + A0B0C1

A0 (B0C0 + B0C2 + B1C0 + B2C0) +(A1+A2+A0(C0 + C1))

Multi-layer structure of morphoCAs

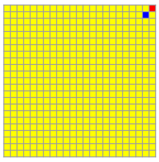
Example: ruleDM[{1, 11, 3, 9, 15}] step-wise realization

start (init) : yellow, red

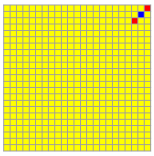


step 37 : {0, 0, 0} → 0, {0, 1, 0} → 0, {1, 0, 0} → 0

construction: blue, yellow, red, yellow

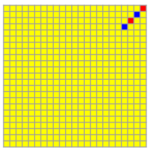


step 44 : {0, 0, 1} → 2

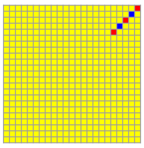


step 66 : {0, 0, 2} → 1, {0, 2, 0} → 0, {2, 0, 0} → 0

iteration of construction



step 88 : {0, 0, 1} → 2



step 110 : {0, 0, 2} → 1, {0, 2, 0} → 0, {2, 0, 0} → 0

Polylogical interpretation of the reduction

S1 = {0, 1}, S2 = {1, 2}, S3 = {0, 2}

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

      Subrules            Subsystems

ruleDM[1]  : {0, 0, 0} → 0, S1, 3 : yellow
ruleDM[3]  : {0, 1, 0} → 0, S1
ruleDM[9]  : {1, 0, 0} → 0, S1
ruleDM[3]  : {0, 2, 0} → 0, S3
ruleDM[9]  : {2, 0, 0} → 0, S3
ruleDM[11] : {0, 0, 1} → 2, S1 → S2, 3 : blue
ruleDM[11] : {0, 0, 2} → 1, S3 → S1, 2 : red
```

ruleDM[{1, 11, 3, 9, x}]

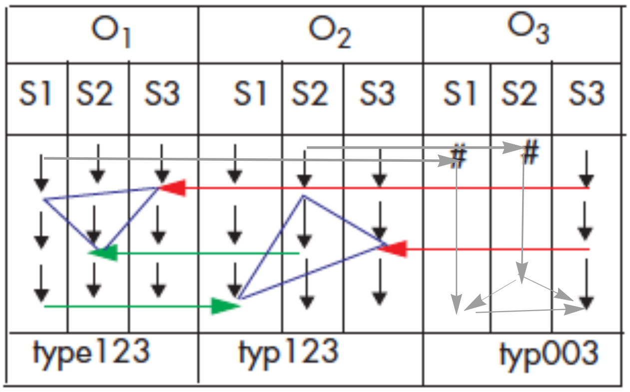
x / yz	00	10	20	01	02
0	[1] : $0_{1,3}$	[8] : 0_1	[8] : 0_3	[11] : $2_{2,3}$	[11] : $1_{1,2}$
1	[9] : 0_1	–	–	–	–
2	[9] : 0_3	–	–	–	–

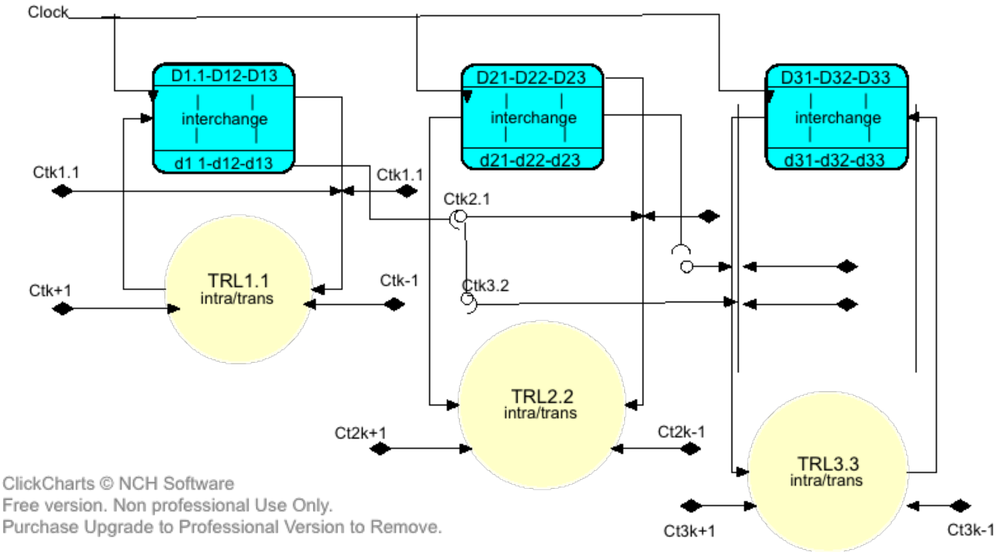
Systems: ruleDM[{1,11,3,9,15}]

Syst1 –> Syst1 || Syst3, 2,
Syst3 –> Syst3 || Syst1, 2
Syst2 –> Ø

Internal mappings: ruleDM[{1,11,3,9,15}]

Syst1 : Mem1 – Log1 – Mem1 || Mem3 , 2
∐
Syst3 : Mem3 – Log3 – Mem3 || Mem1, 2
Syst2 –> Mem2





Again, another formal approach

$\mathcal{R}^{(3,3)} = (\rho^1 \amalg \rho^2) \amalg \rho^3$

ruleDM[{1, 11, 3, 9, 15}]

$\rho^{1.1}$ =

x / yz	00	01	11	10
0	0_1	♯2	□	0_1
1	0_1	□	□	□

+ (0, 01) → 2 : MEM2, 3

$\rho^{2.3} / \rho^{3.1}$ =

x / yz	11	12	22	21
1	□	□	□	□
2	□	□	□	□

$\rho^{3.3}$ =

x / yz	00	02	22	20
0	0_3	♯1	□	0_3
2	0_3	□	□	□

+ (0, 02) → 1 : MEM1, 2

Junctional : ruleDM[1] : $(0, 0, 0)_1 \to 0_1$,
Transjunctional : ruleDM[11] : $(0, 0, 1)_1 \to 2_{2,3}$ || $(0, 0, 2)_3 \to 1_{1,2}$

The automaton at the level of a single layer of the multilayer morphoCAs is not necessarily running on its own without the inclusion of its neighbor components. Multi-layer moprhocAs are defined by their interaction with the partial automata of their neighbor layers and are therefore depending on the definitions of those partial CAs of the complementing layers of the whole morphoCA.

In this case it is supposed that the clocks of the different automata of different layers are synchronized.

$$\mathcal{R}^{(3,3)} = (\rho^1 \amalg \rho^2) \amalg \rho^3, \quad \mathbf{c} = (\mathbf{c}_{k+1}, \mathbf{c}_k, \mathbf{c}_{k-1})$$

$$\mathcal{R}^{(3,3)}\left(\text{ruleDM}[\{1, 11, 3, 9, 15\}]\right) =$$

$$\left(\left(\begin{array}{c}\rho^{1.1} \\ \hline \rho^{1.2} \\ \hline \rho^{1.3}\end{array}\right) \amalg_{1.2} \left(\begin{array}{c}\rho^{2.1} \\ \hline \rho^{2.2} \\ \hline \rho^{2.3}\end{array}\right)\right) \amalg_{1.2,3} \left(\begin{array}{c}\rho^{3.1} \\ \hline \rho^{3.2} \\ \hline \rho^{3.3}\end{array}\right)$$

$$\mathbf{c} = (\mathbf{c}_{k+1}, \mathbf{c}_k, \mathbf{c}_{k-1})$$

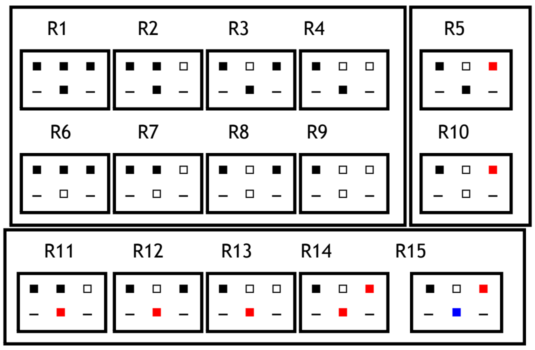
$$\mathcal{R}^{(3,3)}\left(\text{ruleDM}[\{1, 11, 3, 9, 15\}]\right) =$$

$$\left(\left(\begin{array}{c}\rho^{1.1}(\mathbf{c}) \\ \hline - \\ \hline \rho^{1.3}(\mathbf{c})\end{array}\right) \amalg_{1.2} \left(\begin{array}{c}\rho^{2.1}(\mathbf{c}) \\ \hline - \\ \hline \rho^{2.3}(\mathbf{c})\end{array}\right)\right) \amalg_{1.2,3} \left(\begin{array}{c}\rho^{3.1}(\mathbf{c}) \\ \hline - \\ \hline \rho^{3.3}(\mathbf{c})\end{array}\right)$$

$$\mathcal{R}^{(3,3)}\left(\text{ruleDM}[\{1, 11, 3, 9, 15\}]\right) =$$

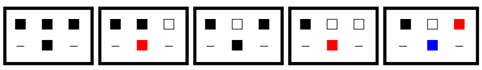
$$\left(\left(\begin{array}{c}\rho^{1.1}(\mathbf{c}_{k+1}, \mathbf{c}_k, \mathbf{c}_{k-1}) \\ \hline - \\ \hline \rho^{1.3}(\mathbf{c}_{k+1}, \mathbf{c}_k, \mathbf{c}_{k-1})\end{array}\right) \amalg_{1.2} \left(\begin{array}{c}\rho^{2.1}(\mathbf{c}_{k+1}, \mathbf{c}_k, \mathbf{c}_{k-1}) \\ \hline - \\ \hline \rho^{2.3}(\mathbf{c}_{k+1}, \mathbf{c}_k, \mathbf{c}_{k-1})\end{array}\right)\right) \amalg_{1.2,3} \left(\begin{array}{c}\rho^{3.1}(\mathbf{c}_{k+1}, \mathbf{c}_k, \mathbf{c}_{k-1}) \\ \hline - \\ \hline \rho^{3.3}(\mathbf{c}_{k+1}, \mathbf{c}_k, \mathbf{c}_{k-1})\end{array}\right)$$

Visualization of classical morphograms



Example

ruleDM[{1, 11, 3, 13, 15}] =



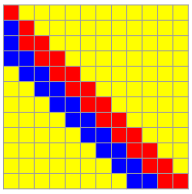
ruleDM[{1, 11, 3, 13, 15}]

x / yz	00	01	10	02	03	20	30
0	0_1	2_{11}	0_3	1_{11}	2_{11}	0_3	0_3
1	2_{13}	1_3	□	3_{15}	1_{15}	□	□
2	1_{13}	3_{15}	□	2_3	1_{15}	□	□
3	□	2_{15}	□	2_{15}	3_3	□	□

The morphoCA of ruleDM[{1, 11, 12, 4, 15}] needs, to work as a CA, additionally to the morphic skeleton values of the morphograms [{1, 11, 12, 4, 15}], some embeddings as semiotic or symbolic interpretations of the rules. That doesn't come as a surprise. The skeleton ‘values’ of the morphograms are also just interpretations of the morphograms and not their direct representations.

Further simple examples for parallelism

ruleDM[{1, 2, 12, 4, 10}]

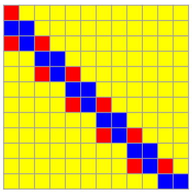


ruleDM[{1, 2, 12, 4, 10}]

x / yz	00	01	02	10	11	12	21	22
0	0	0	0	2	□	□	2	0
1	1	□	□	1	□	□	□	□
2	□	□	□	1	2	□	2	□

ruleDCM[{1, 2, 12, 13, 5}]

```
ArrayPlot[CellularAutomaton[
   {
    {0, 0, 0} → 0,

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

   {1, 2, 1} → 0, {0, 1, 0} → 2,

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

   {0, 1, 2} → 0, {2, 1, 0} → 2
   },
   {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green},
 Mesh → True, ImageSize → 100]
```



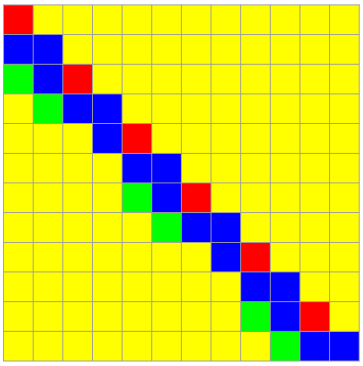
ruleDCM[{1, 2, 12, 13, 5}] =

x / yz	00	01	02	10	20	12	21	22
0	0	0	0	2	□	0	□	1
1	2	□	□	□	□	□	0	□
2	1	□	□	2	2	□	□	□

ruleDM[{1, 11, 12, 4, 15}]

x / y	0	1	2	0	1	2	0	1	2
0	0	2	□	0	□	□	□	□	□
1	2	□	□	□	□	0	□	□	□
2		□	□		□	□		□	1
z	0			1			2		

ruleDM[{1, 2, 12, 13, 5}]



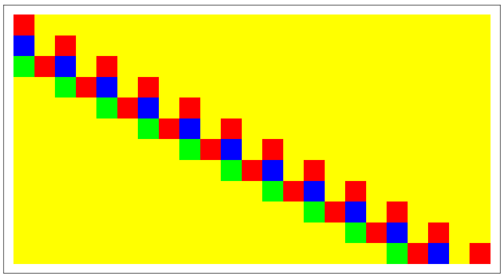
ruleDM[{1, 2, 12, 13, 5}]

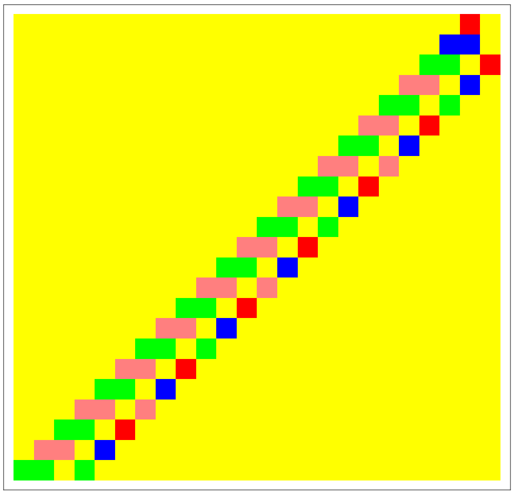
x / yz	00	01	02	03	10	20	21	22
0	0	0	0	0	2	□	0	3
1	2	□	□	□	□	□	0	□
2	1	□	□	□	2	2	□	□
3	□	□	□	□	□	□	3	0

Gray code

ruleDM[{1, 2, 12, 13, 5}]

x / yz	00	01	02	12	11	21	22	20	22
0	0	0	0	□	□	0	3	□	3
1	2	□	□	□	□	0	□	□	□
2	1	□	□	□	□	□	□	2	□
3	□	□	□	3	□	□	0	□	0





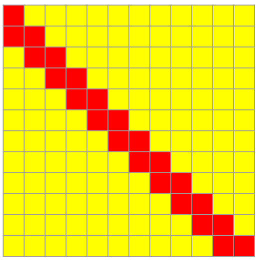
Reduction of ECA rules via sub-rule elimination

Example1

```
ruleECA[{1, 2, 8, 4, 5, 11, 13, 16}]
```

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

Reducts = {{1, 0, 1} -> 1, {1, 1, 1} -> 1}: sub-rules 11 and 16.



ruleECA[{1, 2, 8, 4, 5, 11, 13, 16}]
reduced to ruleECA[{1, 2, 8, 4, 5, 13}] :

x / yz	00	01	11	10
0	0	0	0	1
1	1	–	–	1

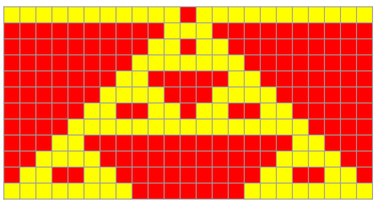
Example2

ruleECA[{6, 2, 3, 4, 10, 12, 14, 16}]

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

Reduct = {{1, 0, 1} -> 1}: sub-rule 12.

```
ArrayPlot[CellularAutomaton[
  {
   {0, 0, 0} → 1,
   {0, 0, 1} → 0,
   {0, 1, 0} → 0,
   {0, 1, 1} → 0,
   {1, 0, 0} → 0,
   {1, 1, 0} → 0,
   {1, 1, 1} → 1
  },
  {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow},
 Mesh → True, ImageSize → 200]
```



ruleECA[{6, 2, 3, 4, 10, 12, 14, 16}]
reduced to ruleECA[{6, 2, 3, 4, 10, 14, 16}] :

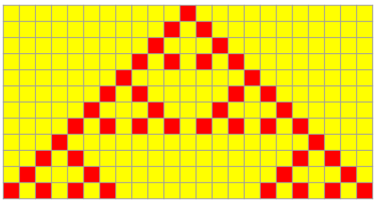
x / yz	00	01	11	10
0	0	0	0	0
1	1	–	1	0

ruleECA[{1, 7, 3, 9, 5, 12, 13, 15}]

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

Reducts : {0, 1, 1} -> 1, {1, 1, 0} -> 1, {1, 1, 1} -> 0

```
ArrayPlot[CellularAutomaton[
  {
   {0, 0, 0} → 0, {0, 0, 1} → 1,
   {0, 1, 0} → 0, {1, 0, 0} → 1,
   {1, 0, 1} → 0
  },
  {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow},
 Mesh → True, ImageSize → 200]
```



ruleECA[{1, 7, 3, 9, 5, 12, 13, 15}] reduced to
ruleECA[{1, 7, 3, 5, 12}]

x / yz	00	01	11	10
0	0	1	-	0
1	1	0	-	-

"Situations can arise where a circuit has N input signals, but not all 2^N combinations of inputs are possible. Or, if all 2^N combinations of inputs are possible, some combinations might be irrelevant."

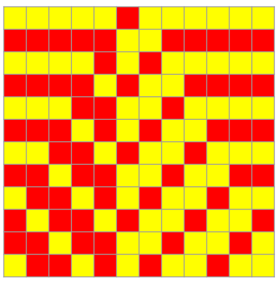
https : // www.digilentinc.com / classroom / realdigital / M4 / RealDigital_Module _ 4. pdf

Hence, there is some redundancy even in the simplest ECA rules.

Non - reducible ECA example

ruleECA[{6, 7, 3, 4, 10, 11, 13, 15}]

```
ArrayPlot[CellularAutomaton[
  {
   {0, 0, 0} → 1, {0, 0, 1} → 1, {0, 1, 0} → 0,
   {0, 1, 1} → 0, {1, 0, 0} → 0, {1, 0, 1} → 1, {1, 1, 0} → 1, {1, 1, 1} → 0
  },
  {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow},
 Mesh → True, ImageSize → 200]
```



ruleECA[{6, 7, 3, 4, 10, 11, 13, 15}] is not reducible.

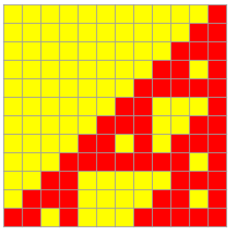
x / yz	00	01	11	10
0	1	1	0	0
1	0	1	0	1

The ECA rule 110 belongs to the balanced non-reducible rules.

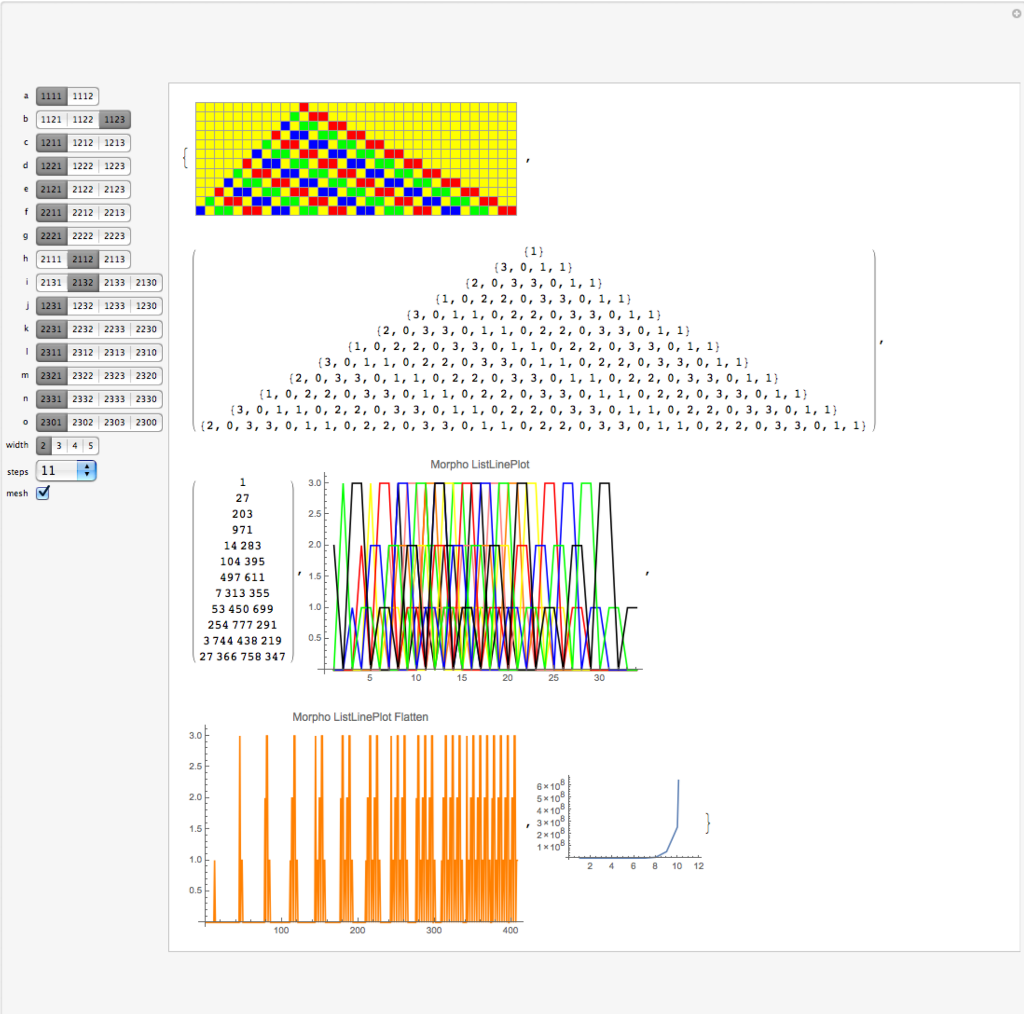
ruleECA 110 :

x / yz	00	01	11	10
0	0	1	1	1
1	0	1	0	1

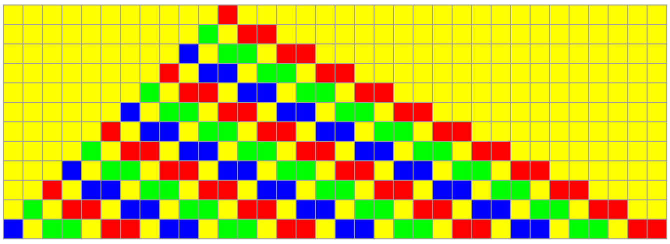
```
ArrayPlot[CellularAutomaton[
   110, {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow},
 Mesh → True, ImageSize → 200]
```



More complex examples: $CA^{(5,4,3)}$



```
ArrayPlot[
 CellularAutomaton[ruleDCKV[{1111, 1123, 1211, 1221, 2121, 2211, 2221, 2112,
    2132, 1231, 2231, 2311, 2321, 2331, 2301}], {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green, 4 → Pink, 5 → Black},
 Mesh → True, ImageSize → Medium]
```



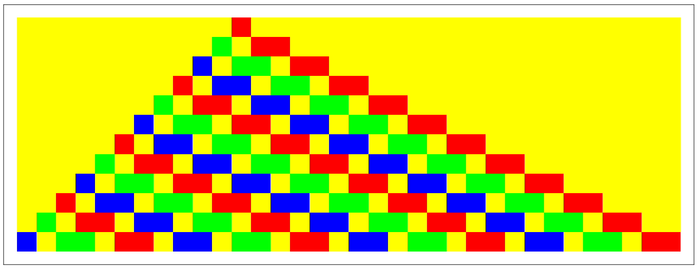
```
ruleDCKV[{1111, 1123, 1211, 1221, 2121, 2211,
  2221, 2112, 2132, 1231, 2231, 2311, 2321, 2331, 2301}]
```

Manual reduction of ruleDCK-V[{1111,1123,1211,1221,2121,2211,2221,2112,2132,1231,2231,2311,2321,2331,2301}]

Some sub - rules are redundant, the rest gets a significant reduction.

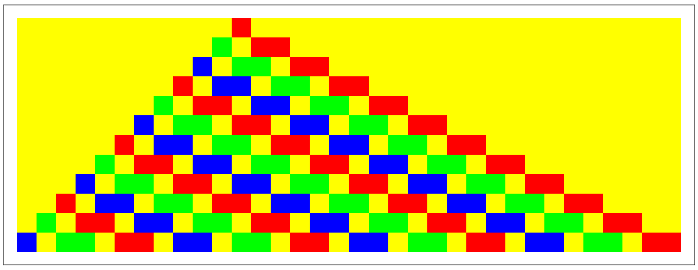
```
ArrayPlot[CellularAutomaton[{

   {0, 0, 0, 0} → 0,

   (*{1,1,1,0}→3,{2,2,2,1}→0,{3,3,3,2}→1,*){0, 0, 0, 3} → 2,
   (*{1,1,1,3}→2,{2,2,2,0}→3,{3,3,3,1}→0,*){0, 0, 0, 2} → 1,
   (*{1,1,1,2}→0,{2,2,2,3}→1,{3,3,3,0}→2,*){0, 0, 0, 1} → 3,

   (*{1,1,2,1}→1,{1,1,3,1}→1,{1,1,0,1}→1,{2,2,3,2}→2,
   {2,2,0,2}→2,{2,2,1,2}→2,{3,3,0,3}→3,{3,3,1,3}→3,{3,3,2,3}→3,*)
   {0, 0, 1, 0} → 0, {0, 0, 2, 0} → 0, {0, 0, 3, 0} → 0,

   (*{1,1,2,2}→1,{1,1,3,3}→1,*)
   {1, 1, 0, 0} → 1,
   (*{2,2,3,3}→2,{2,2,0,0}→2,{2,2,1,1}→2,{3,3,0,0}→3,
   {3,3,1,1}→3,{3,3,2,2}→3,{0,0,1,1}→0,{0,0,2,2}→0,{0,0,3,3}→0,*)

   (*{1,2,1,2}→1,{1,3,1,3}→1,{1,0,1,0}→1,{2,3,2,3}→2,{2,0,2,0}→2,{2,1,2,1}→2,
   {3,0,3,0}→3,{3,1,3,1}→3,{3,2,3,2}→3,{0,1,0,1}→0,{0,2,0,2}→0,{0,3,0,3}→0,*)

   (*{1,2,2,1}→1,{1,3,3,1}→1,{1,0,0,1}→1,{2,3,3,2}→2,
   {2,0,0,2}→2,{2,1,1,2}→2,{3,0,0,3}→3,{3,1,1,3}→3,{3,2,2,3}→3,*)
   {0, 1, 1, 0} → 0, {0, 2, 2, 0} → 0, {0, 3, 3, 0} → 0,

   (*{1,2,2,2}→1,{1,3,3,3}→1,*)
   {1, 0, 0, 0} → 1,
   (*{2,3,3,3}→2,{2,0,0,0}→2,{2,1,1,1}→2,{3,0,0,0}→3,
   {3,1,1,1}→3,{3,2,2,2}→3,{0,1,1,1}→0,{0,2,2,2}→0,{0,3,3,3}→0,*)

   (*{1,2,1,1}→2,{1,3,1,1}→3,{1,0,1,1}→0,{2,3,2,2}→3,
   {2,0,2,2}→0,{2,1,2,2}→1,{3,0,3,3}→0,{3,1,3,3}→1,{3,2,3,3}→2,*)
   {0, 1, 0, 0} → 1,
   (*{0,2,0,0}→2,{0,3,0,0}→3,*)

   (*{1,2,1,3}→2,{2,3,2,0}→3,{3,0,3,1}→0,*)
   {0, 1, 0, 2} → 1,
   (*{1,3,1,2}→3,{2,0,2,3}→0,{3,1,3,0}→1,{0,2,0,1}→2,*)
   (*{1,0,1,3}→0,{2,1,2,0}→1,{3,2,3,1}→2,{0,3,0,2}→3,*)

   (*{1,3,1,0}→3,{2,0,2,1}→0,{3,1,3,2}→1,*)
   {0, 2, 0, 3} → 2,
```

```
  (*{1,2,1,0}→2,{2,3,2,1}→3,{3,0,3,2}→0,{0,1,0,3}→1,*)
  (*{1,0,1,2}→0,
  {2,1,2,3}→1,{3,2,3,0}→2,*)
  {0, 3, 0, 1} → 3,

  (*{1,1,2,3}→1,{2,2,3,0}→2,*)
  {3, 3, 0, 1} → 3,
  (*{0,0,1,2}→0,{1,1,3,0}→1,{2,2,0,1}→2,{3,3,1,2}→3,
  {0,0,2,3}→0,{1,1,0,3}→1,{2,2,1,0}→2,{3,3,2,1}→3,{0,0,3,2}→0,
  {1,1,2,0}→1,{2,2,3,1}→2,{3,3,0,2}→3,{0,0,1,3}→0,{1,1,3,2}→1,*)
  {2, 2, 0, 3} → 2,
  (*{3,3,1,0}→3,{0,0,2,1}→0,*)
  {1, 1, 0, 2} → 1,
  (*{2,2,1,3}→2,{3,3,2,0}→3,{0,0,3,1}→0,*)

  (*{1,2,2,3}→1,{2,3,3,0}→2,{3,0,0,1}→3,{0,1,1,2}→0,{1,3,3,0}→1,{2,0,0,1}→2,
  {3,1,1,2}→3,{0,2,2,3}→0,{1,0,0,3}→1,{2,1,1,0}→2,{3,2,2,1}→3,{0,3,3,2}→0,
  {1,2,2,0}→1,{2,3,3,1}→2,{3,0,0,2}→3,{0,1,1,3}→0,{1,3,3,2}→1,{2,0,0,3}→2,
  {3,1,1,0}→3,{0,2,2,1}→0,{1,0,0,2}→1,{2,1,1,3}→2,{3,2,2,0}→3,{0,3,3,1}→0,*)

  (*{1,2,3,1}→1,{2,3,0,2}→2,{3,0,1,3}→3,{0,1,2,0}→0,{1,3,0,1}→1,{2,0,1,2}→2,
  {3,1,2,3}→3,{0,2,3,0}→0,{1,0,3,1}→1,{2,1,0,2}→2,{3,2,1,3}→3,{0,3,2,0}→0,
  {1,2,0,1}→1,{2,3,1,2}→2,{3,0,2,3}→3,{0,1,3,0}→0,{1,3,2,1}→1,{2,0,3,2}→2,
  {3,1,0,3}→3,{0,2,1,0}→0,{1,0,2,1}→1,{2,1,3,2}→2,{3,2,0,3}→3,{0,3,1,0}→0,*)

  (*{1,2,3,2}→1,{2,3,0,3}→2,{3,0,1,0}→3,{0,1,2,1}→0,{1,3,0,3}→1,{2,0,1,0}→2,
  {3,1,2,1}→3,{0,2,3,2}→0,{1,0,3,0}→1,{2,1,0,1}→2,{3,2,1,2}→3,{0,3,2,3}→0,
  {1,2,0,2}→1,{2,3,1,3}→2,{3,0,2,0}→3,{0,1,3,1}→0,{1,3,2,3}→1,{2,0,3,0}→2,
  {3,1,0,1}→3,{0,2,1,2}→0,{1,0,2,0}→1,{2,1,3,1}→2,{3,2,0,2}→3,{0,3,1,3}→0,*)

  (*{1,2,3,3}→1,{2,3,0,0}→2,*)
  {3, 0, 1, 1} → 3,
  (*{0,1,2,2}→0,{1,3,0,0}→1,{2,0,1,1}→2,{3,1,2,2}→3,
  {0,2,3,3}→0,{1,0,3,3}→1,{2,1,0,0}→2,{3,2,1,1}→3,{0,3,2,2}→0,
  {1,2,0,0}→1,{2,3,1,1}→2,{3,0,2,2}→3,{0,1,3,3}→0,{1,3,2,2}→1,*)
  {2, 0, 3, 3} → 2,
  (*{3,1,0,0}→3,{0,2,1,1}→0,*)
  {1, 0, 2, 2} → 1
  (*{2,1,3,3}→2,{3,2,0,0}→3,{0,3,1,1}→0*)

  (*{1,2,3,0}→1,{2,3,0,1}→2,{3,0,1,2}→3,{0,1,2,3}→0,{1,3,0,2}→1,{2,0,1,3}→2,
  {3,1,2,0}→3,{0,2,3,1}→0,{1,3,2,0}→1,{2,0,3,1}→2,{3,1,0,2}→3,{0,2,1,3}→0,
  {1,0,3,2}→1,{2,1,0,3}→2,{3,2,1,0}→3,{0,3,2,1}→0,{1,2,0,3}→1,{2,3,1,0}→2,
  {3,0,2,1}→3,{0,1,3,2}→0,{1,0,2,3}→1,{2,1,3,0}→2,{3,2,0,1}→3,{0,3,1,2}→0*)

 }, {{1}, 0}, 11],
ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green, 4 → Pink, 5 → Black},
ImageSize → Medium]
```



Result of reduction

```
ArrayPlot[CellularAutomaton[{

    {0, 0, 0, 0} → 0,

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

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

    {1, 1, 0, 0} → 1,

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

    {1, 0, 0, 0} → 1,

    {0, 1, 0, 0} → 1,

    {0, 1, 0, 2} → 1,

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

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

  }, {{1}, 0}, 11],
 ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green, 4 → Pink, 5 → Black},
 ImageSize → Medium]
```



Output - oriented notation

```
ArrayPlot[CellularAutomaton[{

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

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

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

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

  }, {{1}, 0}, 11],
  ColorRules -> {1 -> Red, 0 -> Yellow, 2 → Blue, 3 → Green, 4 → Pink, 5 → Black},
  ImageSize → Medium]
```