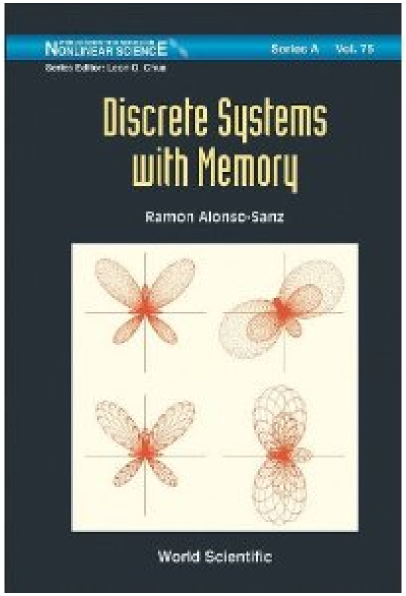
Cellular Automata with Memory

Ramón Alonso-Sanz
ramon.alonso@upm.es

REFERENCES

Conventional CA are Markovian (ahistoric, memoryless[1]): The next state of a cell depends solely on its current neighborhood ($\mathcal{N}$) configuration.

$$\sigma_i^{(T+1)} = \phi\big(\{\sigma_{j\in\mathcal{N}_i}^{(T)}\}\big) \quad \forall i$$



CA with embedded memory

$$s_i^{(T)} = s\big(\sigma_i^{(1)},\dots,\sigma_i^{(T-1)},\sigma_i^{(T)}\big) \quad \longrightarrow \quad \sigma_i^{(T+1)} = \phi\big(\{s_{j\in\mathcal{N}_i}^{(T)}\}\big) \quad \forall i$$

$$f_i^{(T)} = \phi\big(\{\sigma_{j\in\mathcal{N}_i}^{(T)}\}\big) \quad \longrightarrow \quad \sigma_i^{(T+1)} = s\big(f_i^{(1)},\dots f_i^{(T-1)},f_i^{(T)}\big) \quad \forall i$$

CA with delay memory

Extension to the standard framework where:

The mapping ϕ remains unaltered, every cell retains historic memory of its past states by means of the trait state s. So to say, cells canalize memory to the map.

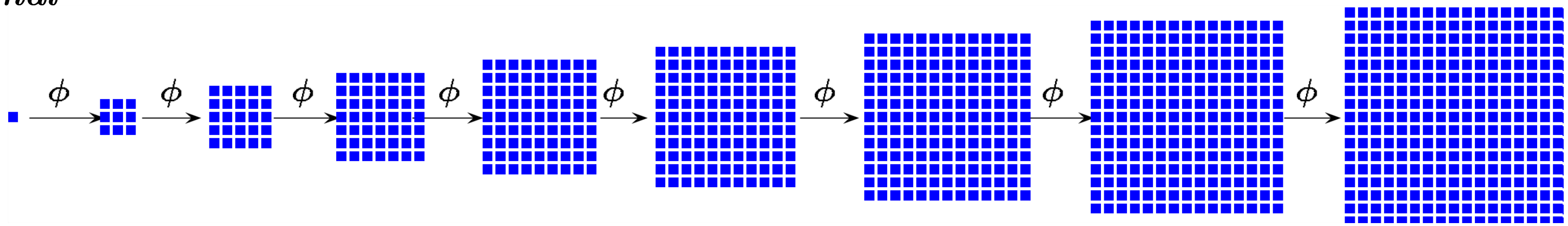
[1] $\sigma_i^{(T)} = \phi\big(\{\sigma_{j\in\mathcal{N}_i}^{(T-1)}\}\big)$ Chain like (indirect) effect of the past $\equiv$ memory-one

Ramon Alonso-Sanz

Example

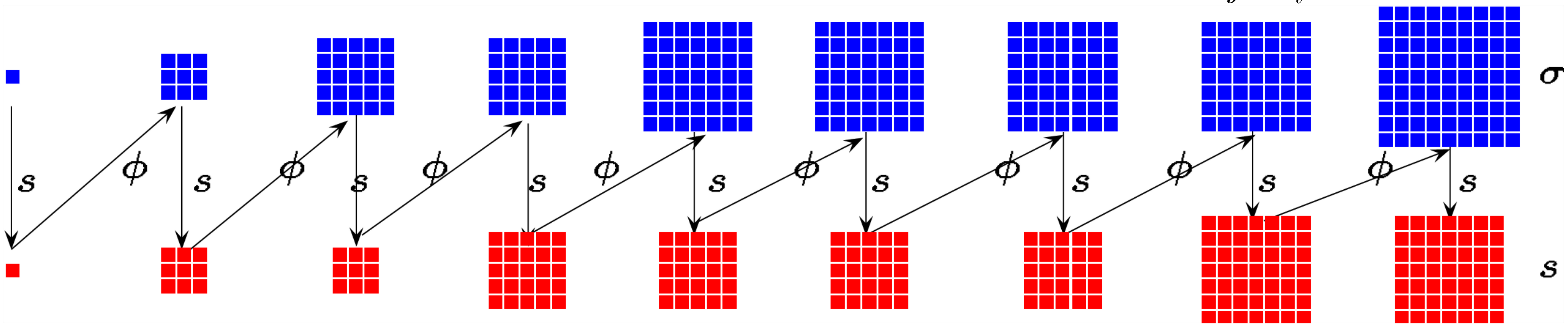
ϕ **: cell alive if any cell in its neighborhood is alive** ***(speed of light).***

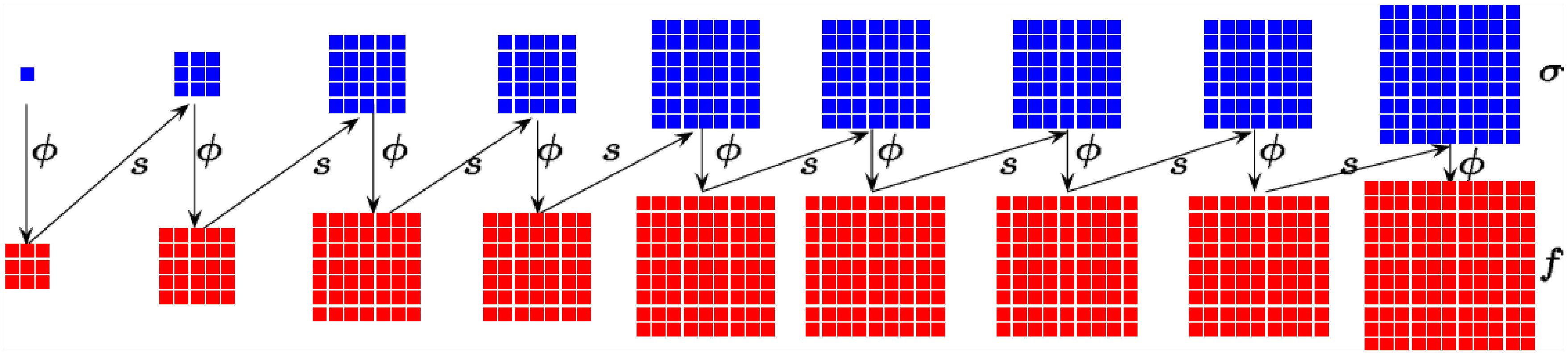
Conventional



s **:** ***Majority (most frequent, mode) memory. Last state in case of a tie.***

Embedded : $s_i^{(T)} = mode(\sigma_i^{(1)}, \ldots, \sigma_i^{(T)}) \quad \longrightarrow \quad \sigma_i^{(T+1)} = \phi\left(\left\{s_{j \in \mathcal{N}_i}^{(T)}\right\}\right) \quad \forall i$





Delay : $f_i^{(T)} = \phi\left(\left\{\sigma_{j \in \mathcal{N}_i}^{(T)}\right\}\right) \quad \longrightarrow \quad \sigma_i^{(T+1)} = mode(f_i^{(1)} \ldots, f_i^{(T)}) \quad \forall i$

MAJORITY memory $\Rightarrow$ INERTIAL effect

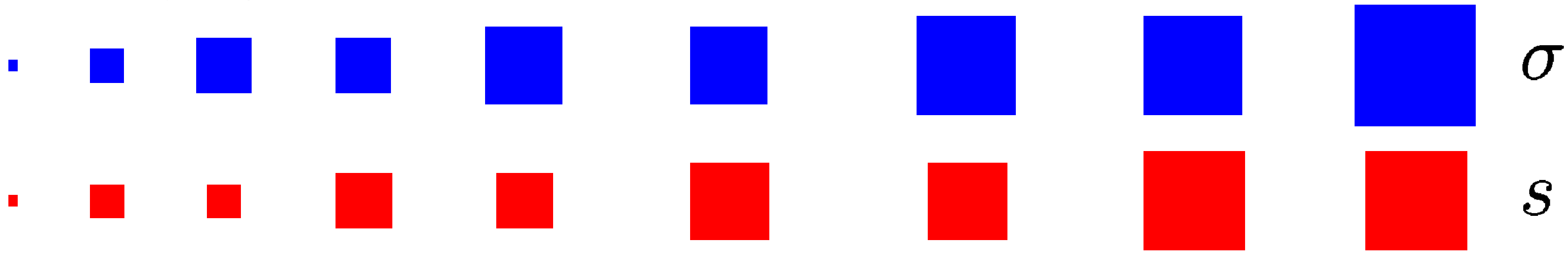
Ramon Alonso-Sanz

LIMITED TRAILILING memory of the last τ states:

Example: τ=3-Majority memory:

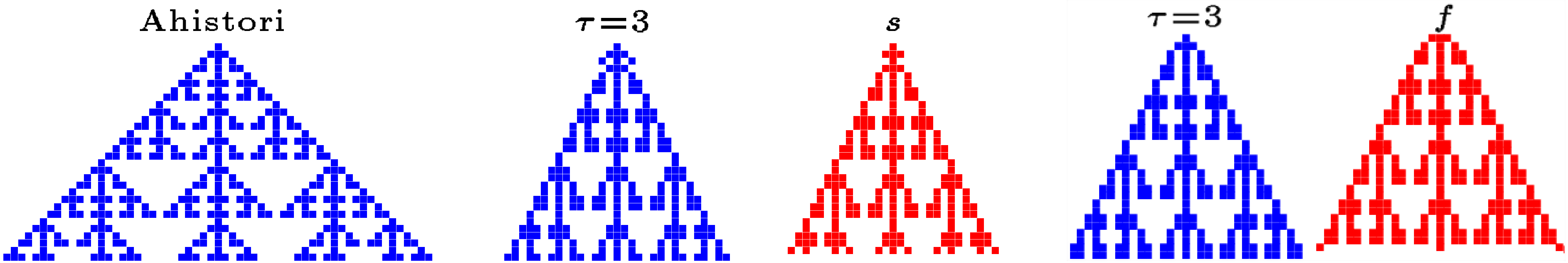
$$\mathbf{s}_i^{(T)} = mode\left(\sigma_i^{(T-2)}, \sigma_i^{(T-1)}, \sigma_i^{(T)}\right) \qquad \sigma_i^{(T+1)} = mode\left(f_i^{(T-2)}, f_i^{(T-1)}, f_i^{(T)}\right) \qquad T > 2$$

Speed of light **with τ=3-Majority Embedded memory**



1D Parity rule ϕ: cell alive iff odd number of alive cells in its neighborhood

$$\sigma_i^{(T+1)} = \sigma_{i-1}^{(T)} \oplus \sigma_i^{(T)} \oplus \sigma_{i+1}^{(T)}$$

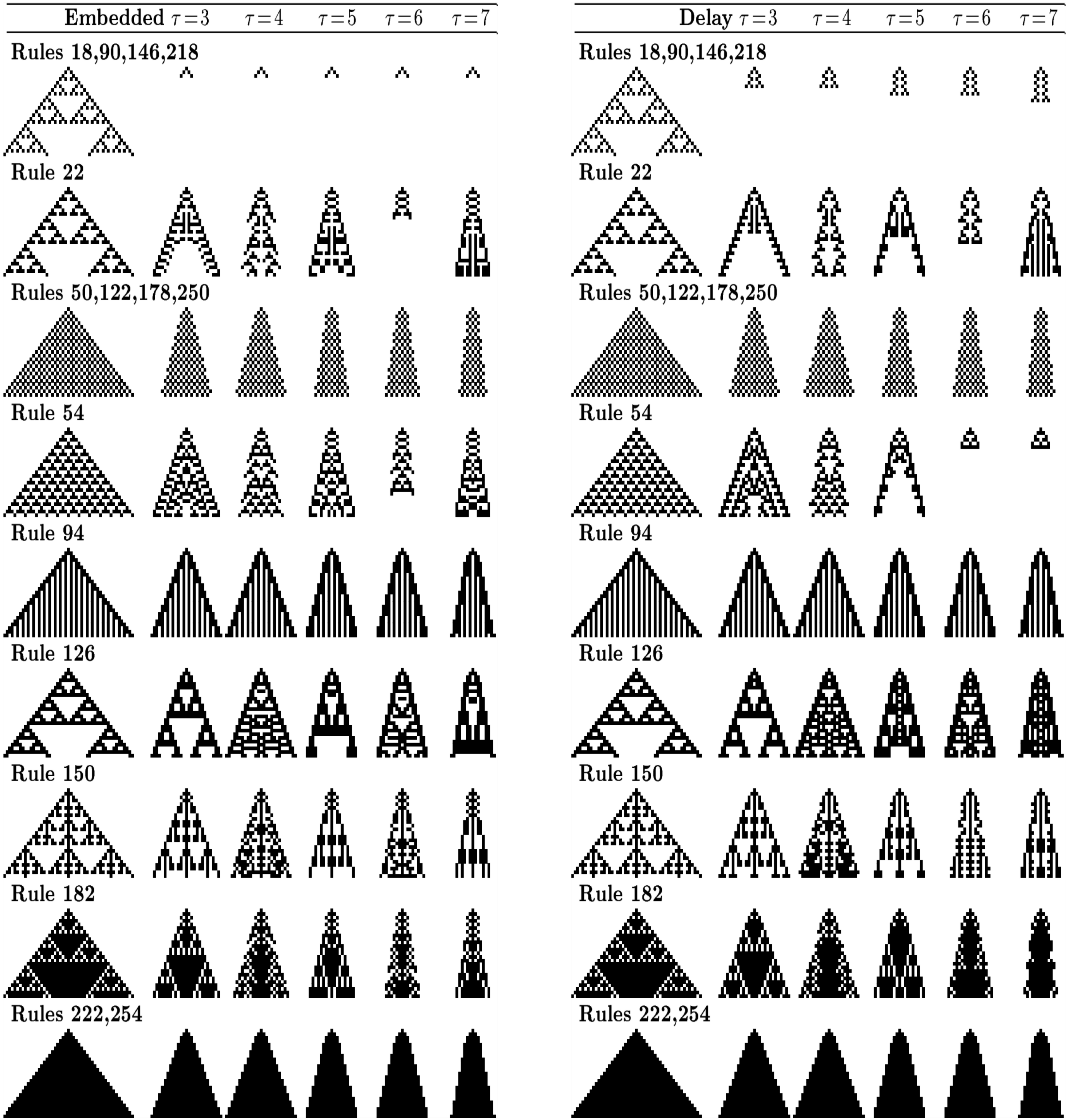


$$s_i^{(T)} = mode\left(\sigma_i^{(T-2)}, \sigma_i^{(T-1)}, \sigma_i^{(T)}\right) \qquad f_i^{(T)} = \sigma_{i-1}^{(T)} \oplus \sigma_i^{(T)} \oplus \sigma_{i+1}^{(T)}$$

$$\sigma_i^{(T+1)} = s_{i-1}^{(T)} \oplus s_i^{(T)} \oplus s_{i+1}^{(T)} \qquad \sigma_i^{(T+1)} = mode\left(f_i^{(T-2)}, f_i^{(T-1)}, f_i^{(T)}\right)$$

Ramon Alonso-Sanz

Elementary Legal Rules with Majority Memory



Ramon Alonso-Sanz

Weighted memory (unlimited trailing embedded memory)

$\boxed{\sigma_i^{(1)}} \times \alpha^{T-1}$

$\cdots\cdots\cdots$

$\boxed{\sigma_i^{(T-2)}} \times \alpha^{2}$

$\boxed{\sigma_i^{(T-1)}} \times \alpha$

$\boxed{\sigma_i^{(T)}} \times 1$

$$m_i^{(T)}(\sigma_i^{(1)},\ldots,\sigma_i^{(T)}) = \frac{\sigma_i^{(T)} + \sum_{t=1}^{T-1} \alpha^{T-t}\sigma_i^{(t)}}{1 + \sum_{t=1}^{T-1} \alpha^{T-t}} \equiv \frac{\omega_i^{(T)}}{\Omega(T)} = \frac{\sigma_i^{(T)} + \alpha\omega_i^{(T-1)}}{1+\alpha\Omega(T-1)} \; ^{*}$$

The choice of the **memory factor $0 \le \alpha \le 1$** fits the memory effect: the limit case $\alpha = 1$ is equivalent to unlimited trailing *majority* memory, whereas $\alpha << 1$ intensifies the contribution of the most recent states (short-range memory). The choice $\alpha = 0$ leads to the ahistoric model.

If $\sigma \in \{0,1\}$, trait state s by rounding the weighted mean m:

$$s_i^{(T)} = \begin{cases} 1 & if \quad m_i^{(T)} > 0.5 \\ \sigma_i^{(T)} & if \quad m_i^{(T)} = 0.5 \\ 0 & if \quad m_i^{(T)} < 0.5 \end{cases} \qquad s_i^{(1)} = \sigma_i^{(1)} \;,\; s_i^{(2)} = \sigma_i^{(2)}$$

Implementation

$k = 2:$ **α-MEMORY EFFECTIVE if $\alpha > 0.5$**

$^{*}\{\sigma_i^{(t)}, t = 1, 2, \ldots, T\}$ **NO NEEDED**

Drawback : $\alpha \to$ real numbers

Weighted memory (unlimited trailing delay memory)

$\boxed{f_i^{(1)}} \times \alpha^{T-1}$

$\dots\dots\dots\dots$

$\boxed{f_i^{(T-2)}} \times \alpha^{2}$

$\boxed{f_i^{(T-1)}} \times \alpha$

$\boxed{f_i^{(T)}} \times 1$

$$m_i^{(T)}(f_i^{(1)},\dots,f_i^{(T)}) = \frac{f_i^{(T)} + \sum\limits_{t=1}^{T-1} \alpha^{T-t} f_i^{(t)}}{1 + \sum\limits_{t=1}^{T-1} \alpha^{T-t}} \equiv \frac{\omega_i^{(T)}}{\Omega(T)} = \frac{f_i^{(T)} + \alpha\omega_i^{(T-1)}}{1+\alpha\Omega(T-1)} \; *$$

The choice of the **memory factor** $0 \leq \alpha \leq 1$ fits the memory effect: the limit case $\alpha = 1$ is equivalent to unlimited trailing *majority* memory, whereas $\alpha << 1$ intensifies the contribution of the most recent states (short-range memory). The choice $\alpha = 0$ leads to the ahistoric model.

If $\sigma \in \{0,1\}$, trait state s by rounding the weighted mean m:

$$s_i^{(T)} = \begin{cases} 1 & if \quad m_i^{(T)} > 0.5 \quad \equiv 2\omega_i^{(T)} > \Omega(T) \\ f_i^{(T)} & if \quad m_i^{(T)} = 0.5 \quad \equiv 2\omega_i^{(T)} = \Omega(T) \\ 0 & if \quad m_i^{(T)} < 0.5 \quad \equiv 2\omega_i^{(T)} < \Omega(T) \end{cases}$$

Implementation

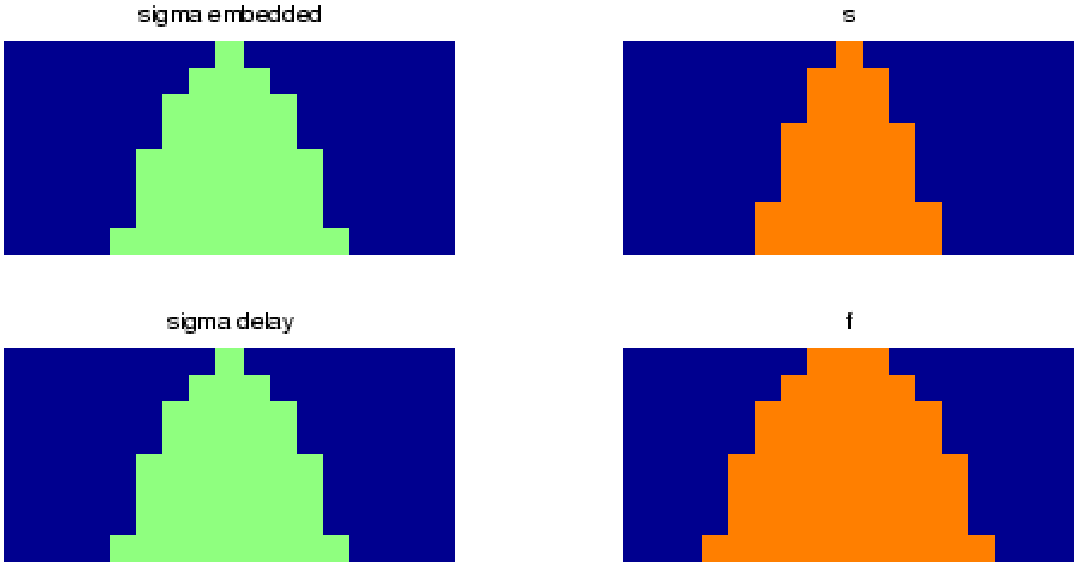
$k = 2:$ **α-MEMORY EFFECTIVE if $\alpha > 0.5$**

$^*\{f_i^{(t)}, t = 1, 2, \dots, T\}$ **NO NEEDED**

Drawback : $\alpha \rightarrow$ **real numbers**

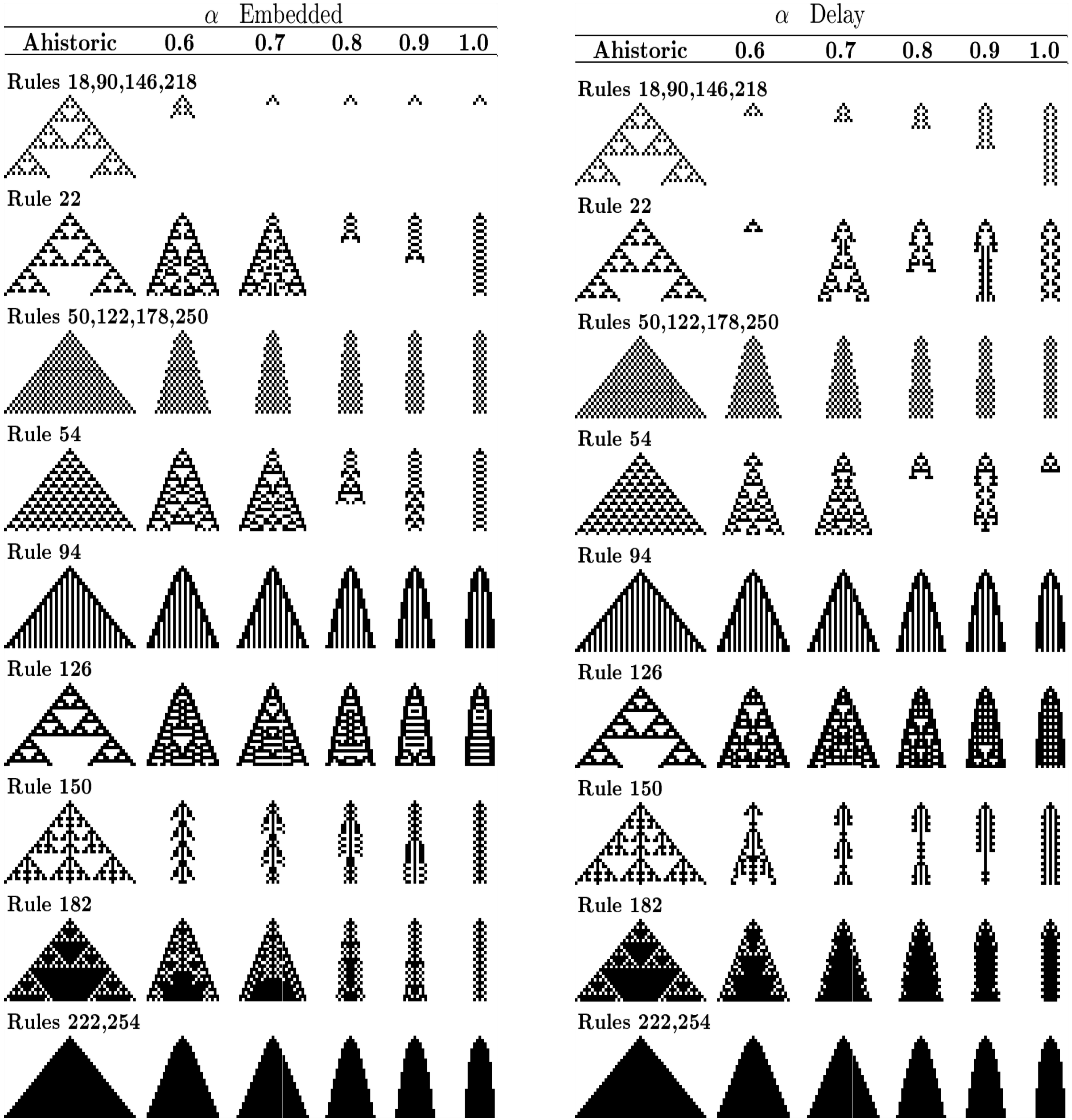
Ramon Alonso-Sanz

```
function cam
T=8;SR=254;alpha=1.0;N=2*T+1;
[srb]=binarynumber(SR);left=[N 1:N-1];right=[2:N 1];
for memo=1:2
  [SIGMA,OMEGA,omega]=init(T,N,alpha);
  switch memo
   case 1   % Embedded
   for t=1:T
    SIGMAH(t,:)=SIGMA;S=SIGMA;
    omega=(alpha*omega)+SIGMA;OMEGAX=OMEGA(t);
    for i=1:N
     if(2*omega(i)>OMEGAX)S(i)=1;end;   % memory
     if(2*omega(i)<OMEGAX)S(i)=0;end
    end
    [SIGMA]=RULE(S,N,srb,left,right);  % rule
    HS(t,:)=S;
   end
   case 2   % Delay
    [SIGMA,OMEGA,omega]=init(T,N,alpha);S=SIGMA;
     for t=1:T
    SIGMAH(t,:)=SIGMA;
    [S] =RULE(SIGMA,N,srb,left,right);  % rule
    SIGMA=S;HS(t,:)=S;
    omega=(alpha*omega)+SIGMA;OMEGAX=OMEGA(t);
    for i=1:N
     if(2*omega(i)>OMEGAX)SIGMA(i)=1;end   % memory
     if(2*omega(i)<OMEGAX)SIGMA(i)=0;end
    end
   end
  end
  subplot(3,2,2*(memo-1)+1);image(33*SIGMAH]);axis image;axis('off'):
  if(memo==1)title('sigma embedded');else;title('sigma delay');end
  subplot(3,2,2*(memo-1)+2);imagesc(33*HS,[0,44]);axis image;axis('off');
  if(memo==1)title('s');else;title('f');end
end
print camembedelay.eps -depsc

function [SIGMA,OMEGA,omega]=init(T,N,alpha);
     SIGMA(1:N)=0; SIGMA((N+1)/2:(N+1)/2)=1;
     OMEGA(1)=1.0;omega(1:N)=0;
     for t=2:T;OMEGA(t)=1+alpha*OMEGA(t-1);end

function [SIGMA]=RULE(S,N,srb,left,right);
    for i=1:N
     SIGMA(i)=srb(8-(4*S(left(i))+2*S(i)+S(right(i))));
    end

function [BN] =binarynumber(rule);
     BN(1:8)=0;irtx=rule;
     for ix=1:8
      rest=mod(irtx,2);ratio=(irtx-rest)/2;BN(8-ix+1)=rest;irtx=ratio;
     end
```



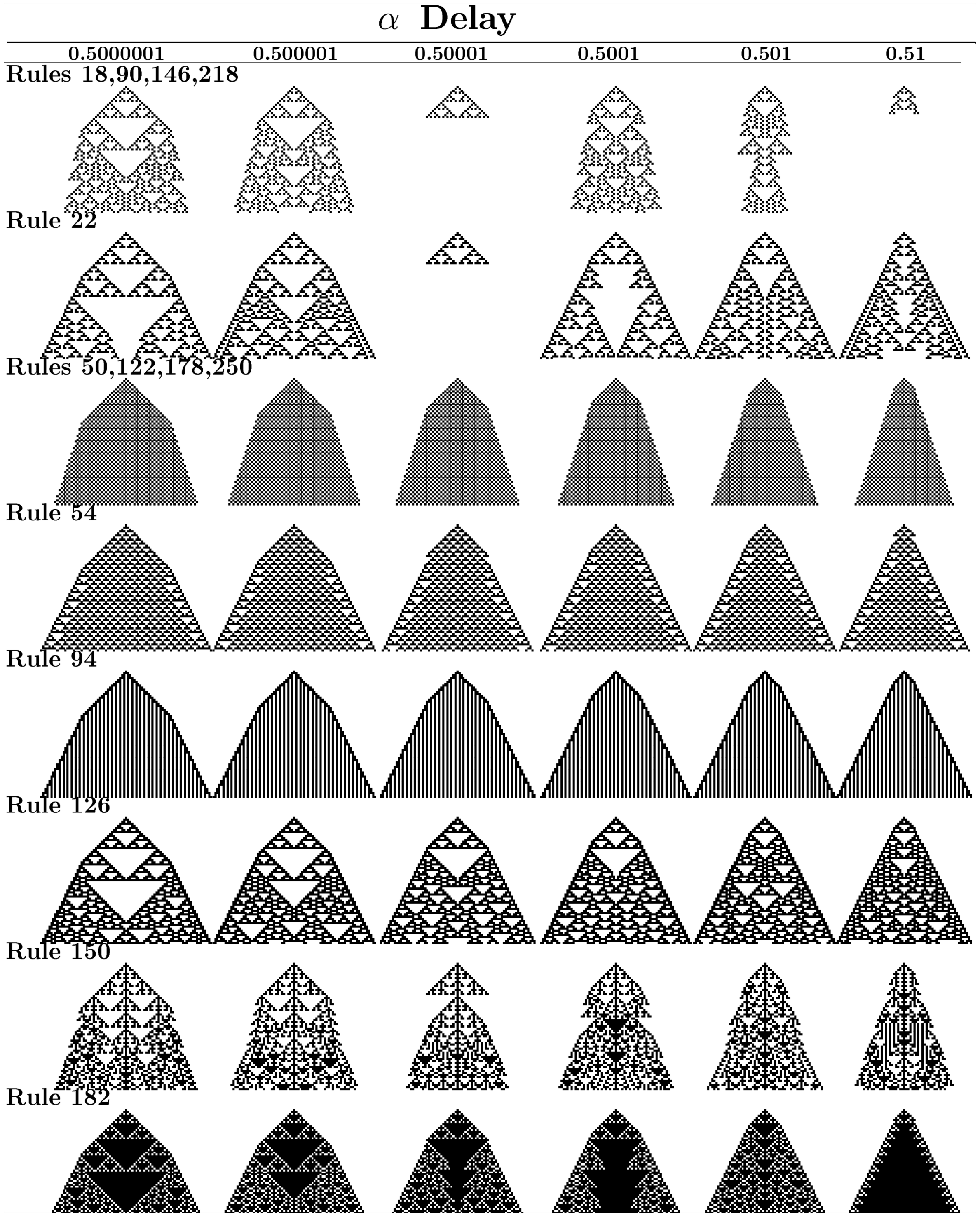
Ramon Alonso-Sanz

Elementary Legal Rules with α-Memory



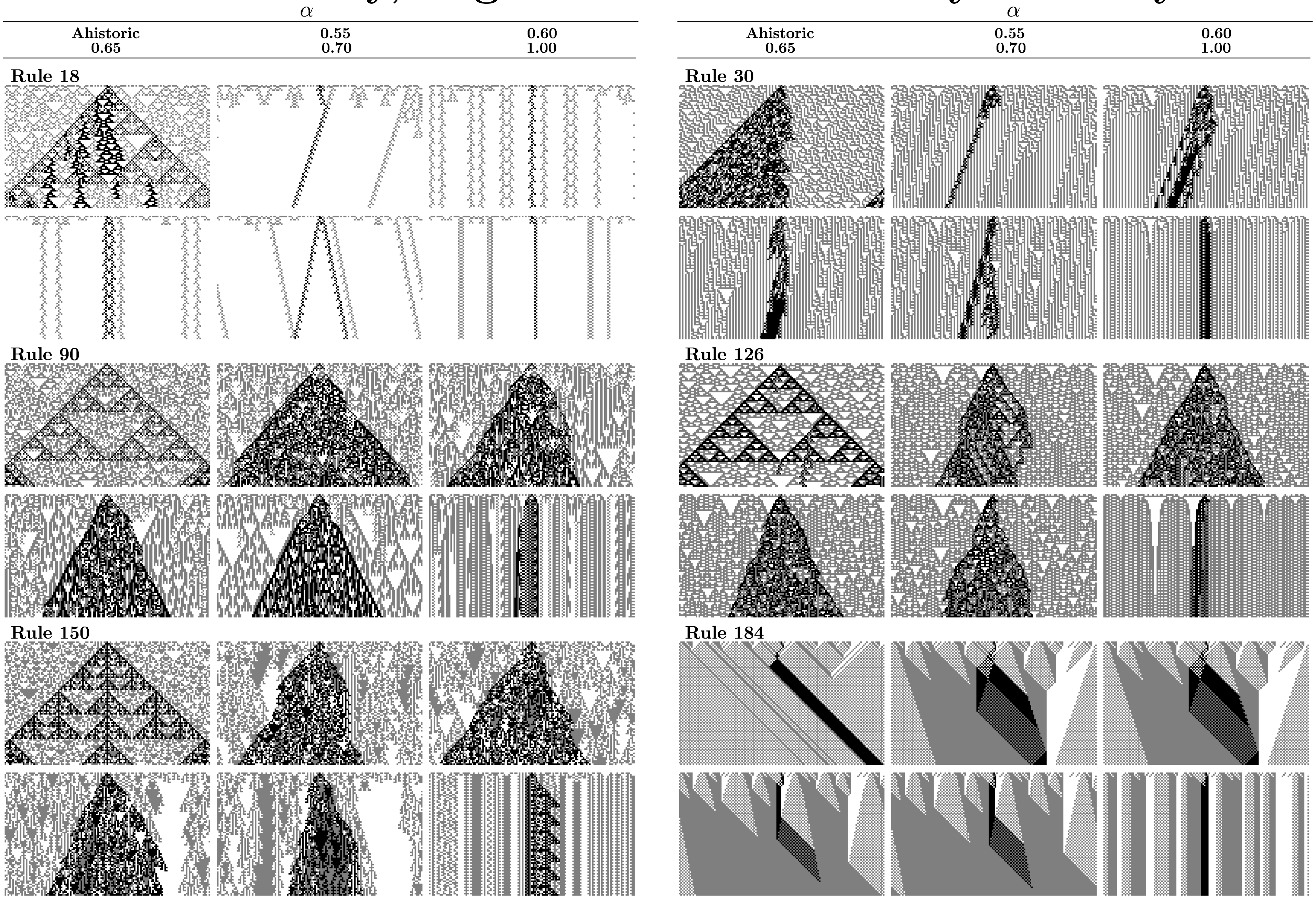
Ramon Alonso-Sanz

Elementary legal rules with low α-delay memory



Ramon Alonso-Sanz

Elementary, Legal Rules with α-delay Memory

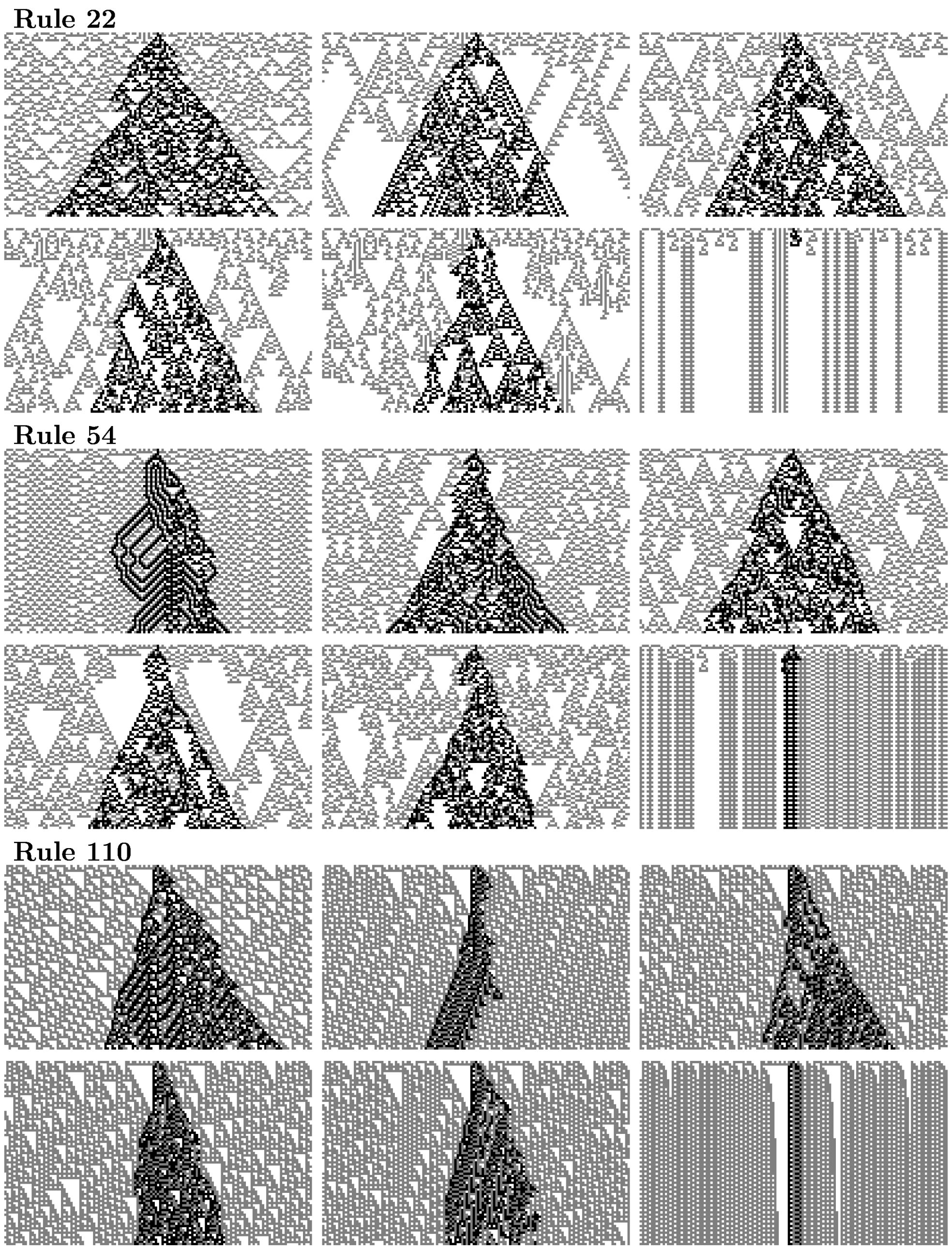


black = Damage spreading

Alonso-Sanz,R.(2013). Elementary cellular automata with memory of delay type. *LNCS*,8155,67-83.

ECA with α-delay Memory

α		
Ahistoric	0.55	0.60
0.65	0.70	1.00



black = Damage spreading

Ramon Alonso-Sanz

1D r=2 CA : neighborhood nearest and next-nearest neighbors

$$\sigma_i^{(T+1)} = \phi\big(\sigma_{i-2}^{(T)}, \sigma_{i-1}^{(T)}, \sigma_i^{(T)}, \sigma_{i+1}^{(T)}, \sigma_{i+2}^{(T)}\big)$$

In totalistic $r = 2$ rules :

$$\sigma_i^{(T+1)} = \phi\big(\sigma_{i-2}^{(T)} + \sigma_{i-1}^{(T)} + \sigma_i^{(T)} + \sigma_{i+1}^{(T)} + \sigma_{i+2}^{(T)}\big)$$

With memory :

$$\sigma_i^{(T+1)} = \phi\big(s_{i-2}^{(T)} + s_{i-1}^{(T)} + s_i^{(T)} + s_{i+1}^{(T)} + s_{i+2}^{(T)}\big)$$

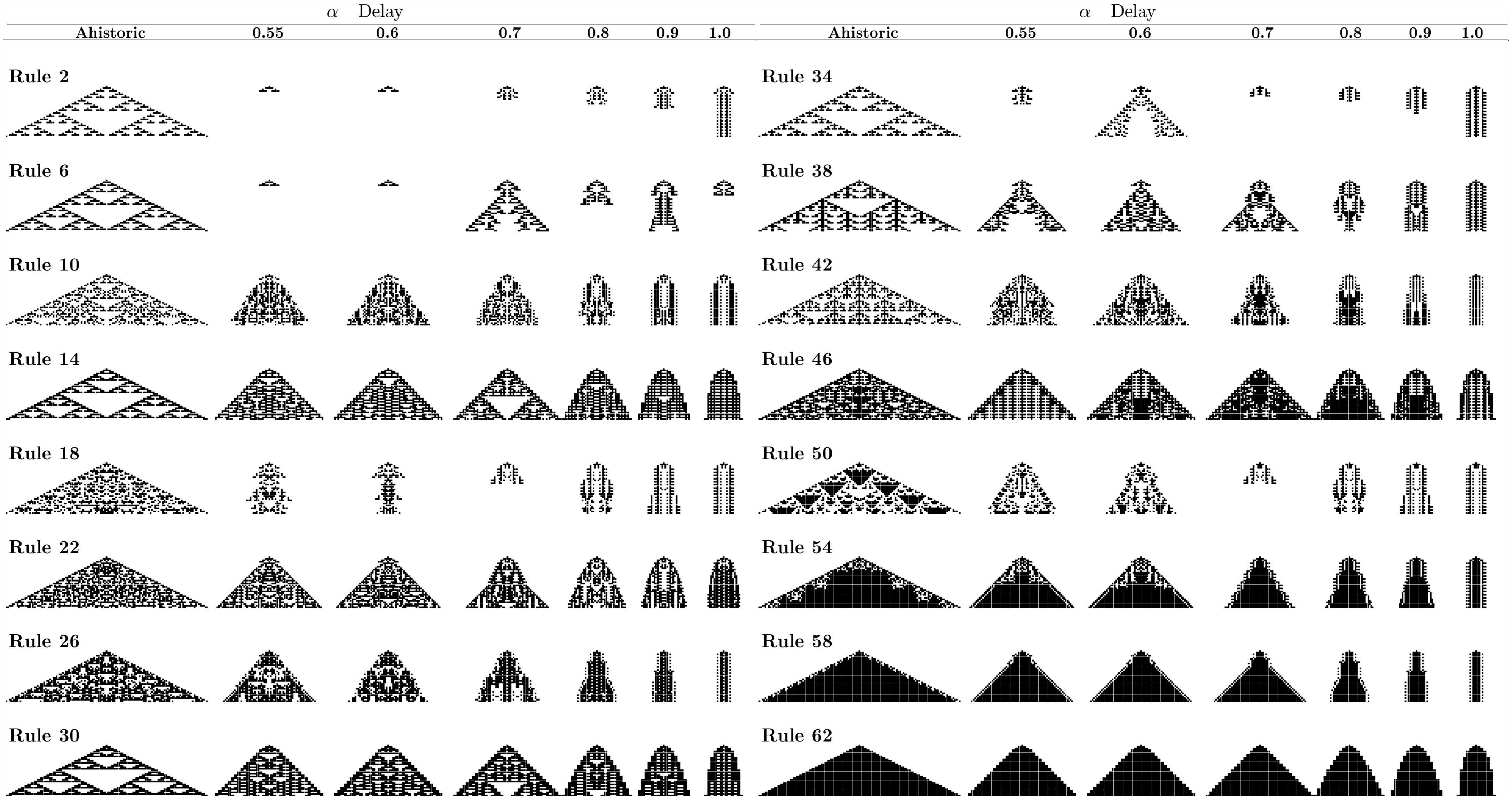
$$f_i^{(T)} = \phi\big(\sigma_{i-2}^{(T)} + \sigma_{i-1}^{(T)} + \sigma_i^{(T)} + \sigma_{i+1}^{(T)} + \sigma_{i+2}^{(T)}\big)$$

Totalistic $k=r=2$ rules are characterized by a sequence of binary values (β_s) associated with each of the six possible values of the sum (s) of the neighbors :

$$\big(\beta_5\beta_4\beta_3\beta_2\beta_1\beta_0\big)_{binary} \equiv \sum_{s=0}^{5} \beta_s 2^s_{\quad decimal} = R \in [0, 63]$$

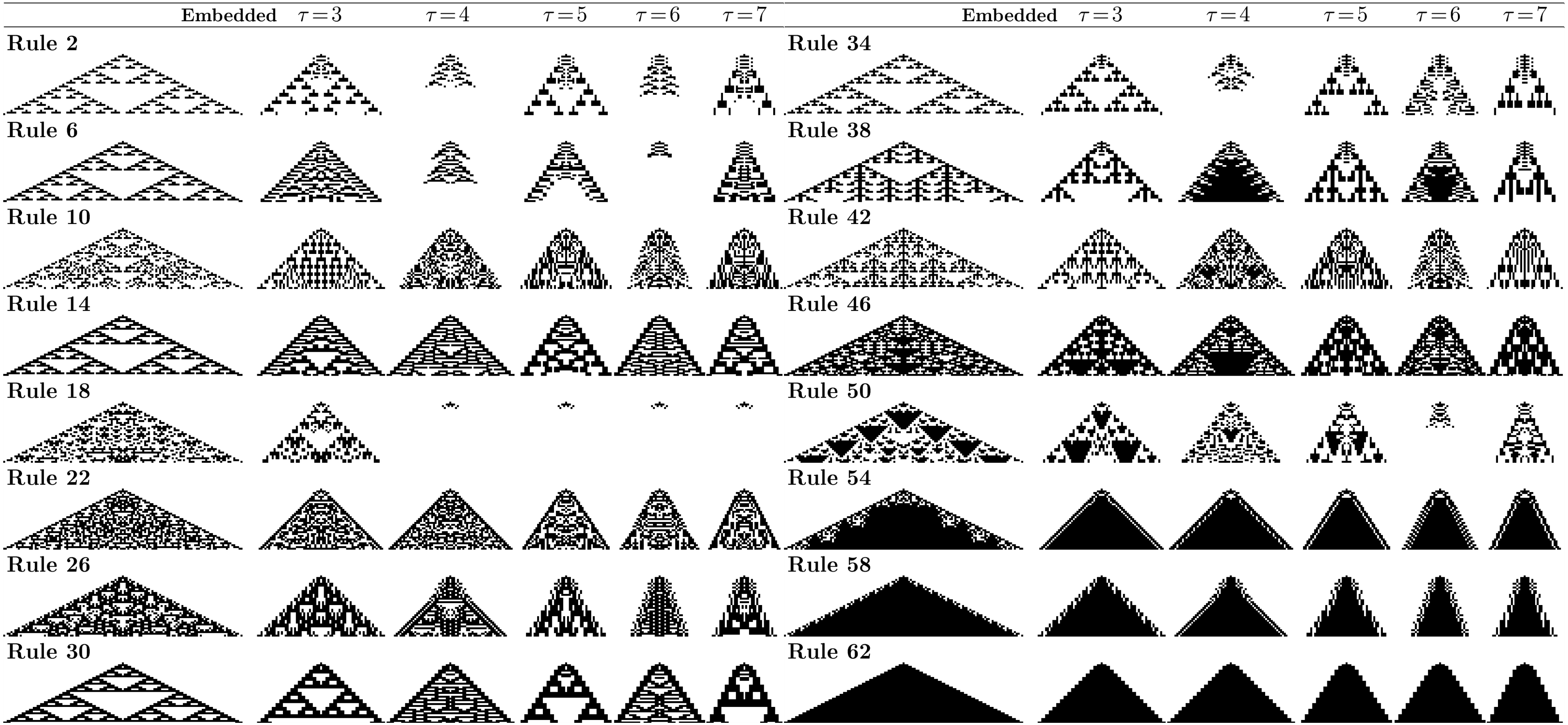
 # totalistic r=2 rules = VR_2^6=2^6=64

1D r=2 CA with α-Embedded Memory



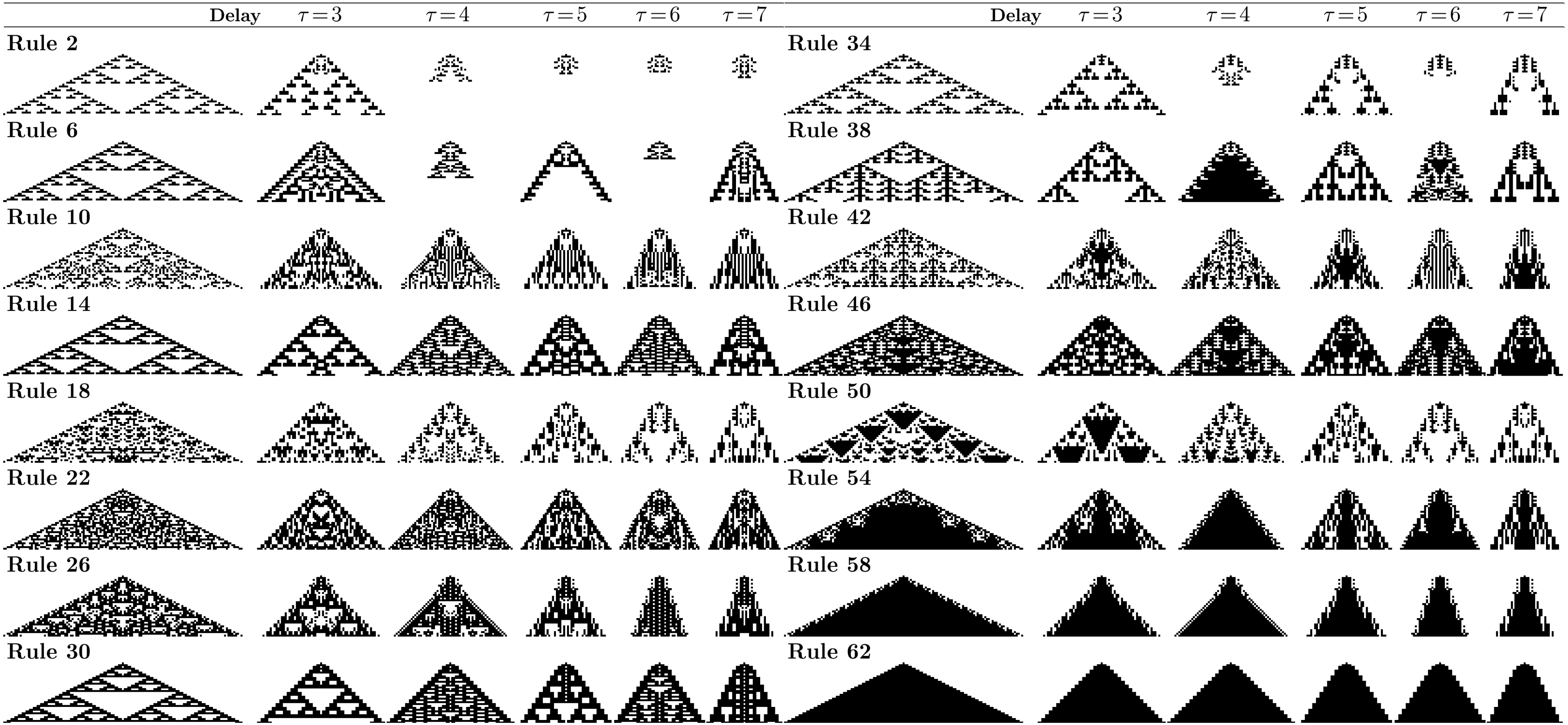
Ramon Alonso-Sanz

1D r=2 CA with Majority Embedded Memory



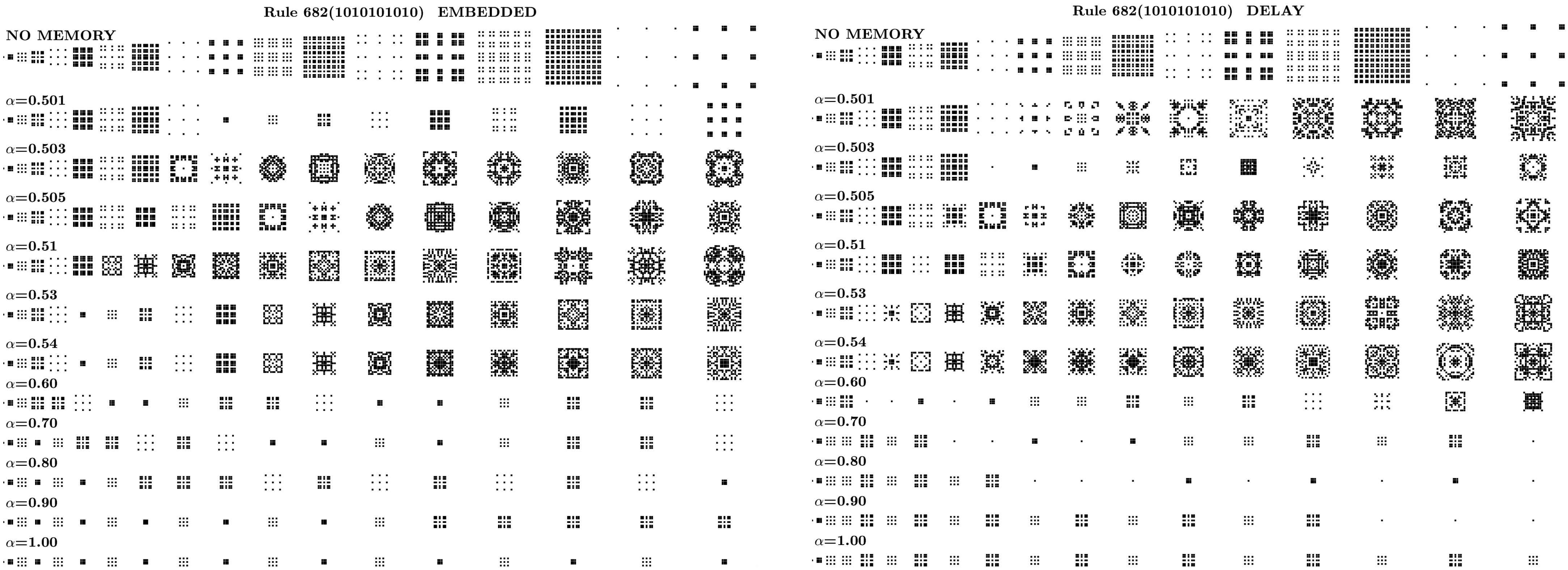
Ramon Alonso-Sanz

1D r=2 CA with Majority Delay Memory



Ramon Alonso-Sanz

The 2D PARITY rule with Memory. Moore N.

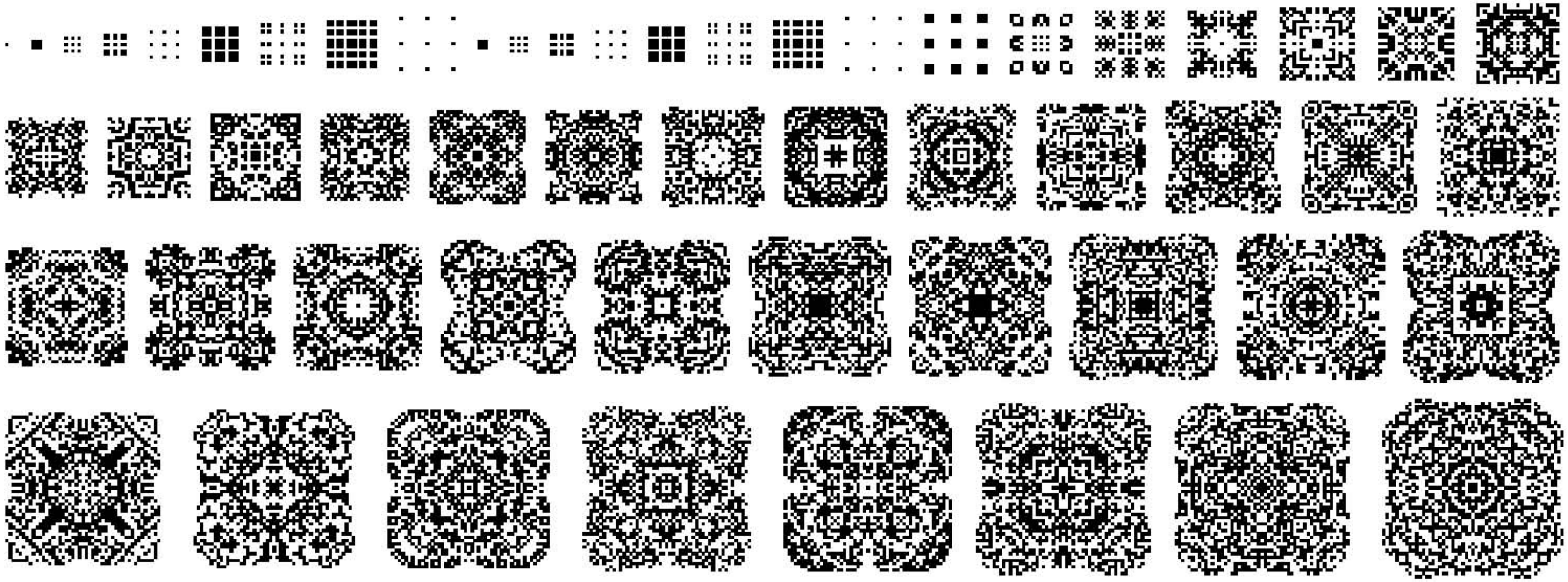


Alonso-Sanz,R.,Martin,M.(2002). Cellular Automata with Memory: patterns starting with a single site seed. *IJMPC*,13,1.

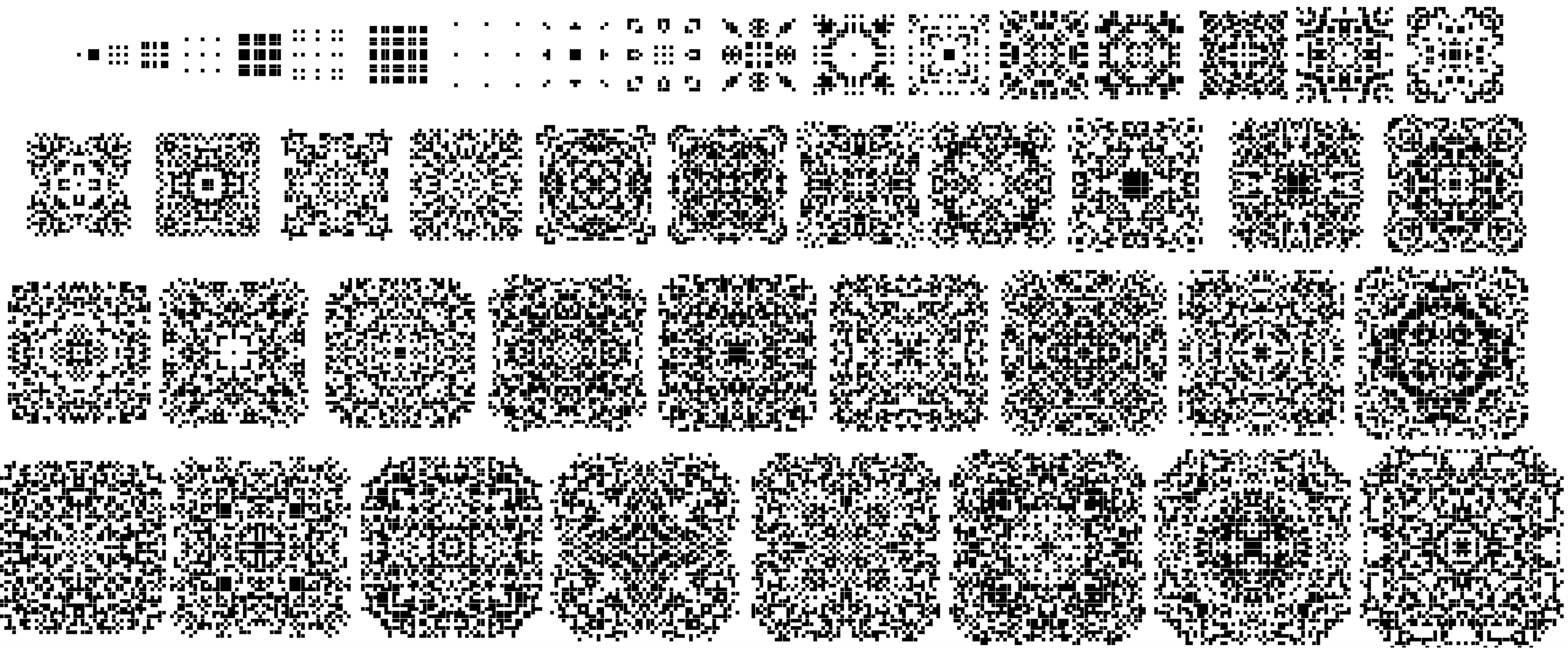
Ramon Alonso-Sanz

The 2D PARITY rule with low memory: $\alpha = 0.501$

EMBEDDED



α=0.501 **Rule 682(1010101010) DELAY**



Ramon Alonso-Sanz

Memory : Chaos (III) → Complex (IV)

Martinez,G.J.,Adamatzky,A.,Alonso-Sanz,R.(2013). Designing complex dynamics in cellular automata with memory. *IJBC*, (in press)

Martinez,G.J.,Adamatzky,A.,Alonso-Sanz,R.(2012). Complex dynamics of elementary cellular automata emerging from chaotic rules. *IJBC*,22,2.

Martinez,G.J.,Adamatzky,A.,Chen,F,Chua,L.(2012). On Soliton Collisions between Localizations in Complex Elementary Cellular Automata: Rules 54 and 110 and Beyond. *Complex Systems*, 21(2),117-142.

Martinez,G.J.,Adamatzky,A.,Alonso-Sanz,R.,Seck-Touh-Mora,J.C.(2009). Complex dynamics emerging in Rule 30 with majority memory. *Complex Systems*,18,3.

Martinez,G.J.,Adamatzky,A.,Seck-Touh-Mora,J.C.,Alonso-Sanz,R.(2010). How to make dull cellular automata complex by adding memory: Rule 126 case study. *Complexity*,15,6.

Alonso-Sanz,R.,Martin,M.(2005). One-dimensional cellular automata with memory in cells of the most recent value. *CS*,15,3.

Elementary Rules (ψ) as Memory ($\tau = 3$):

Embedded memory

$$s_i^{(T)} = \psi\left(\sigma_i^{(T-2)}, \sigma_i^{(T)}, \sigma_i^{(T-1)}\right) \quad \longrightarrow \quad \sigma_i^{(T+1)} = \phi\left(\left\{ s_{j\in\mathcal{N}_i}^{(T)} \right\}\right) \quad \forall i$$

$$f_i^{(T)} = \phi\left(\left\{ \sigma_{j\in\mathcal{N}_i}^{(T)} \right\}\right) \quad \longrightarrow \quad \sigma_i^{(T+1)} = \psi\left(f_i^{(T-2)}, f_i^{(T)}, f_i^{(T-1)}\right) \quad \forall i$$

Delay memory

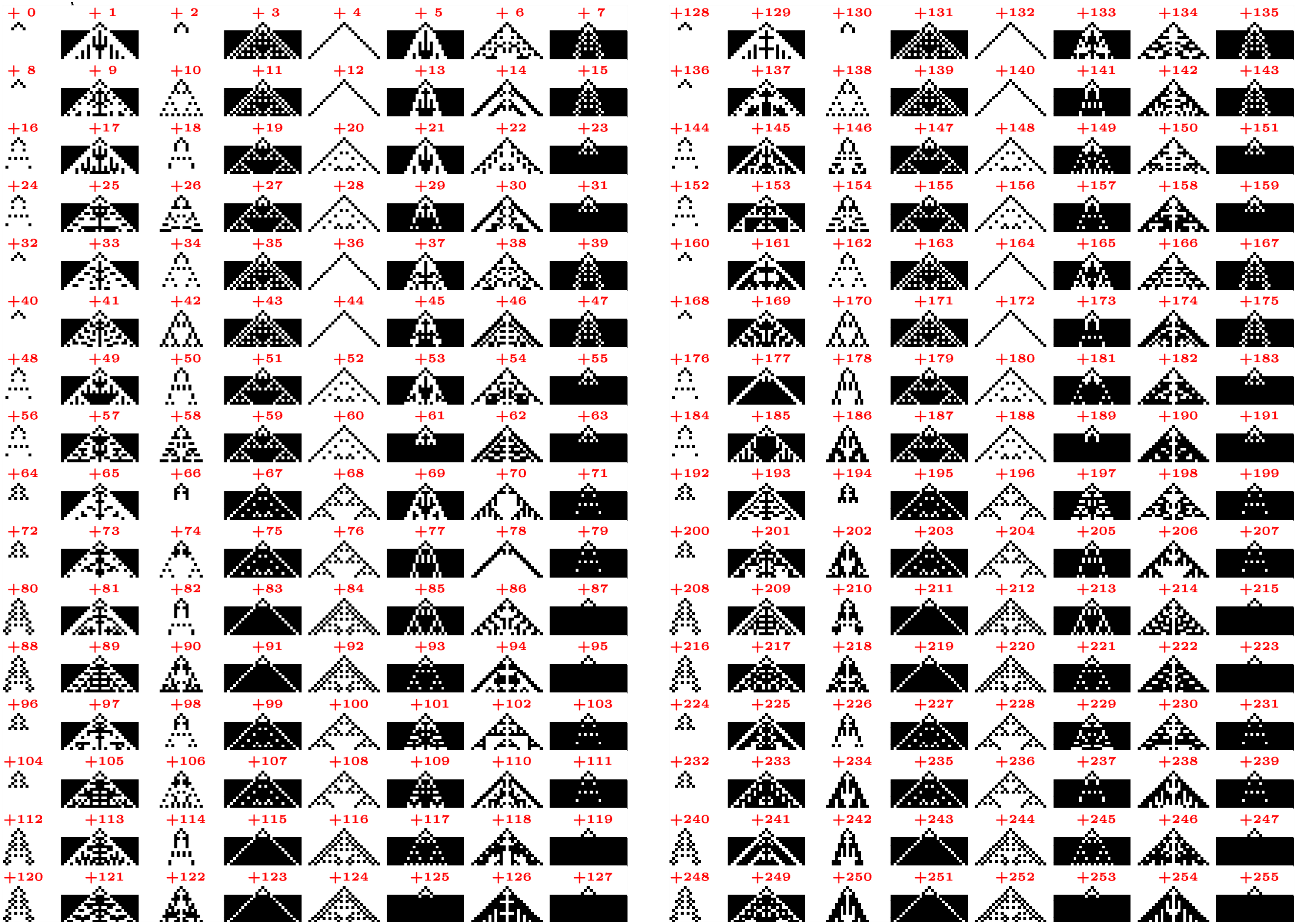
Example: ψ= Majority $\equiv$ ECA232

$$s_i^{(T)} = mode\left(\sigma_i^{(T-2)}, \sigma_i^{(T)}, \sigma_i^{(T-1)}\right) \qquad \sigma_i^{(T+1)} = mode\left(f_i^{(T-2)}, f_i^{(T)}, f_i^{(T-1)}\right)$$

Alonso-Sanz,R.,Bull,L.(2009).Elementary cellular automata with elementary memory rules in cells: The case of linear rules. *JCA*,1,1.

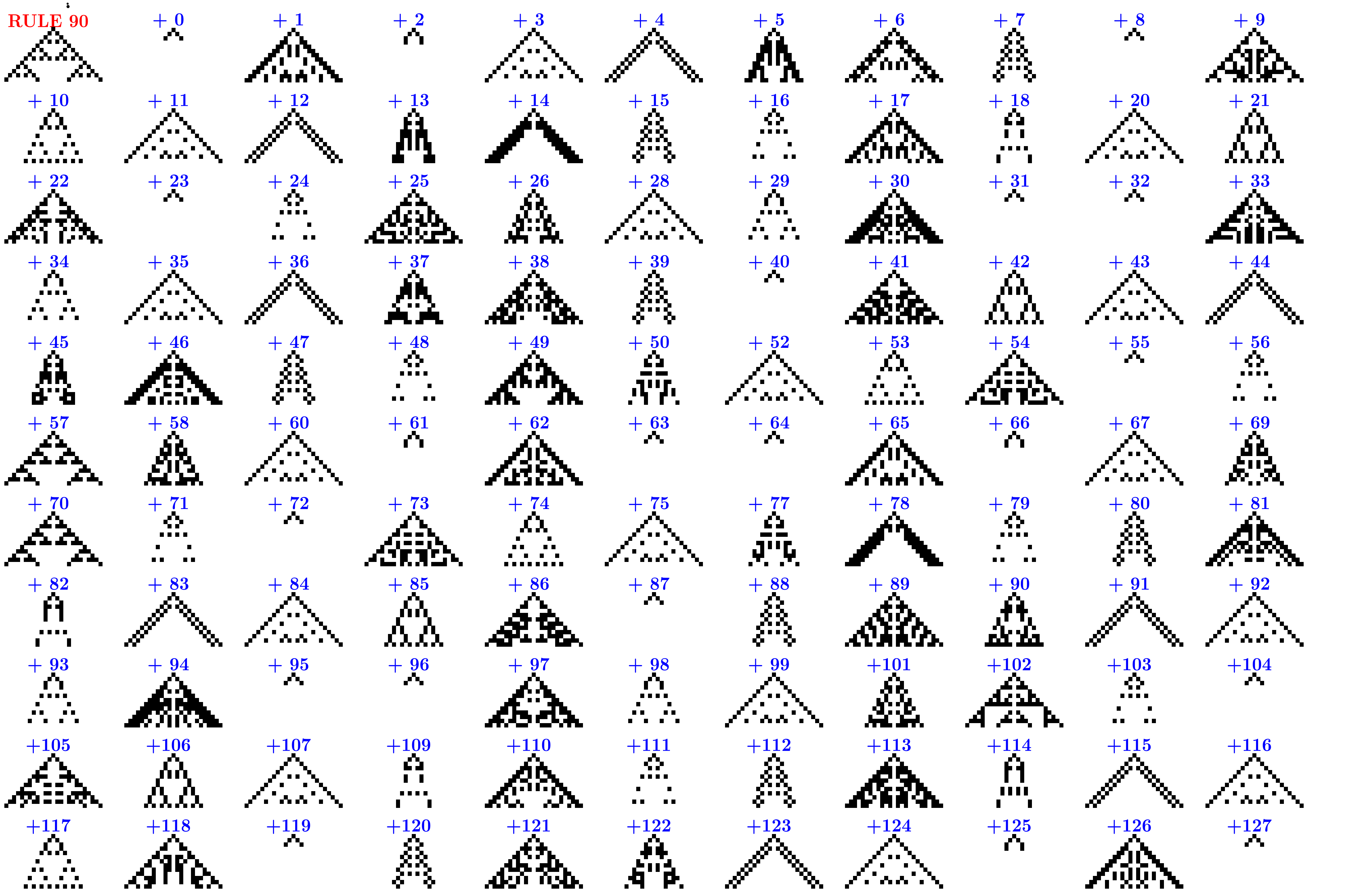
Alonso-Sanz,R.(2013). Elementary cellular automata with memory of delay type. *JCA* (in press).

The rule $\phi = 150$ with delay ECA memories



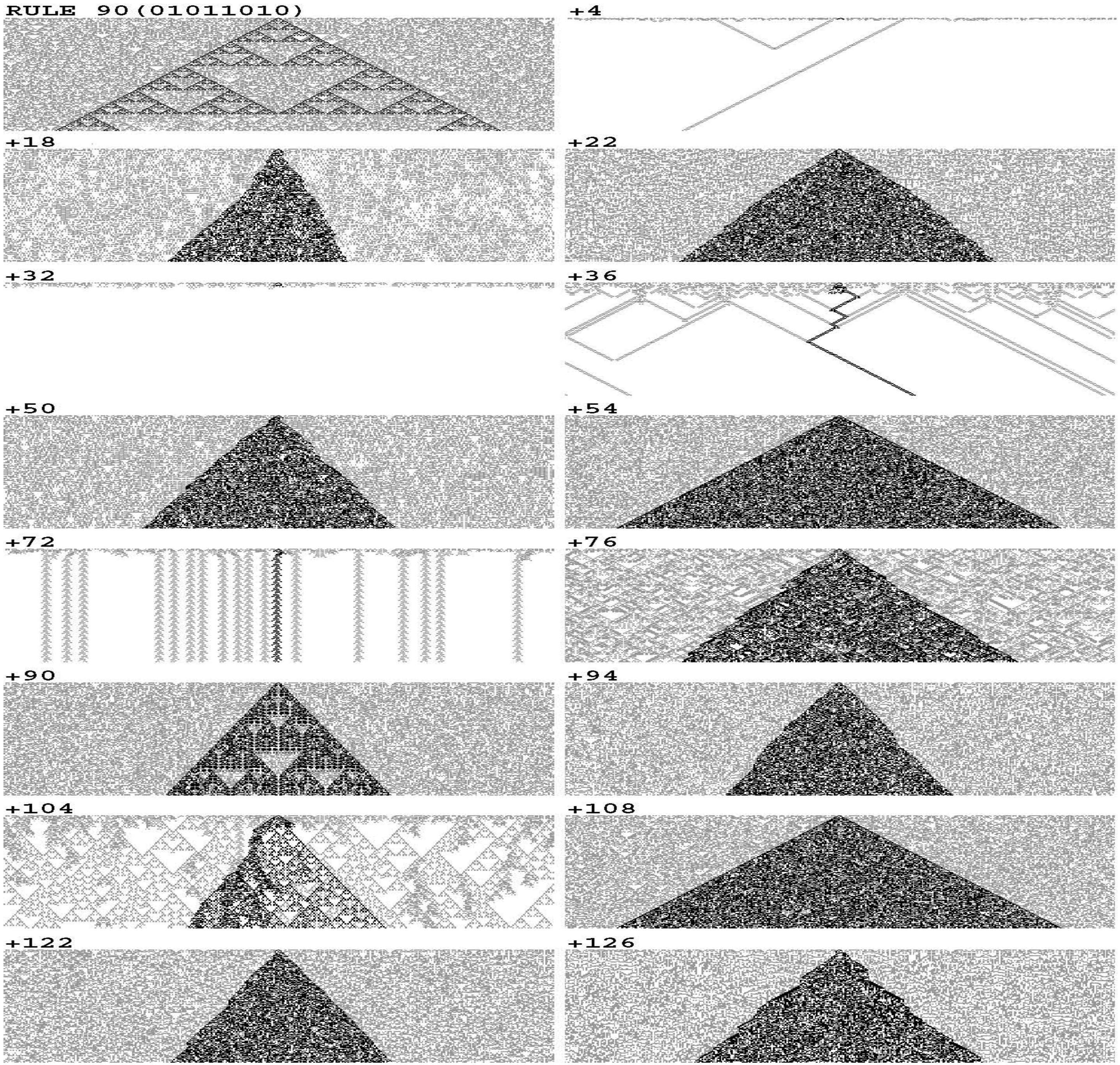
Ramon Alonso-Sanz

The rule 90 with embedded ECA memories

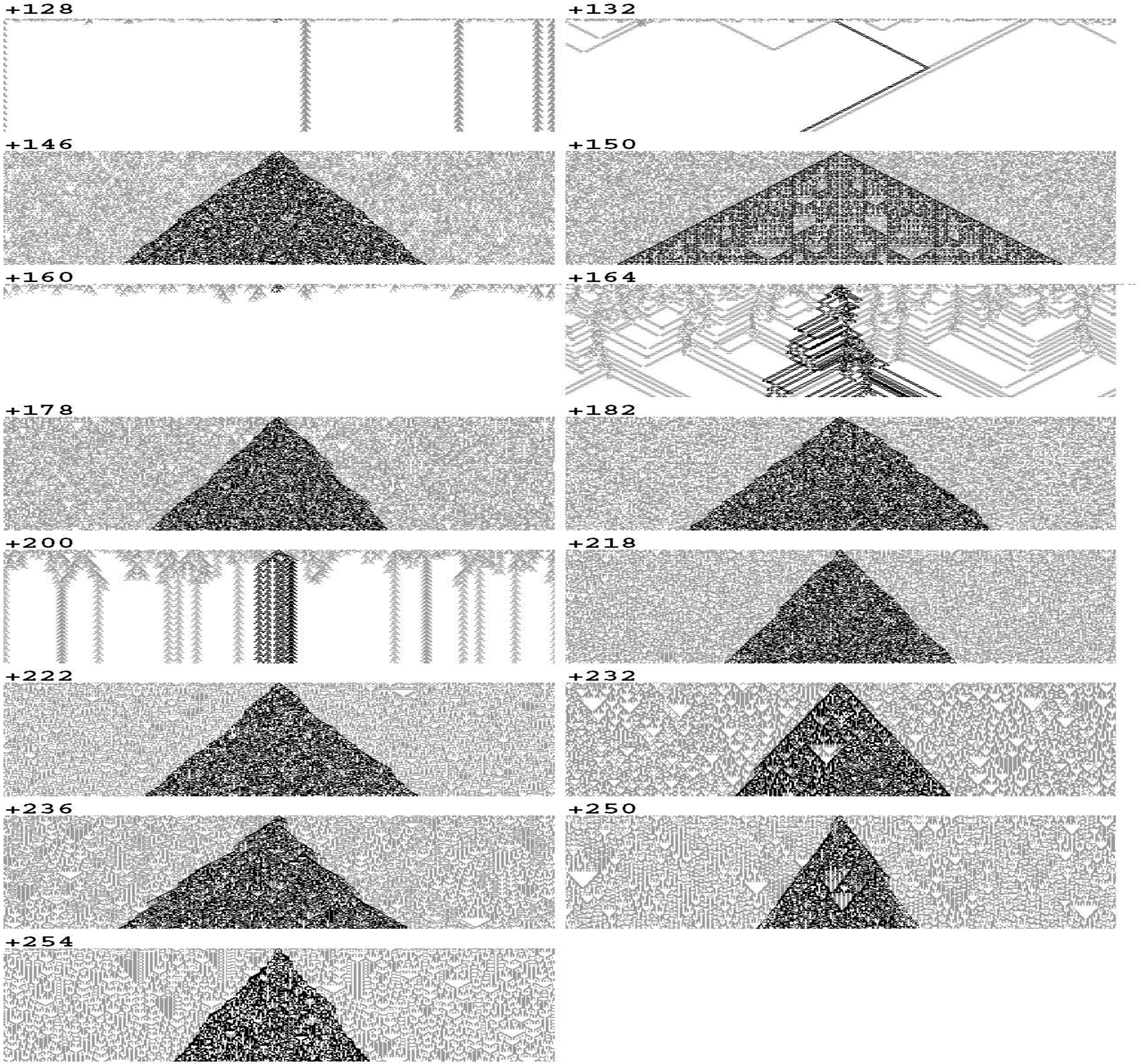


Ramon Alonso-Sanz

The rule 90 with embedded ECA memories



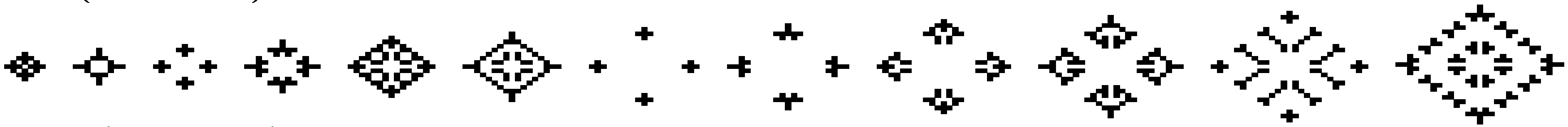
The rule 90 with embedded ECA memories



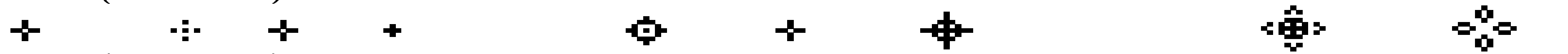
Ramon Alonso-Sanz

The Parity rule with Elementary Rules as Memory. T=4-15. vNN

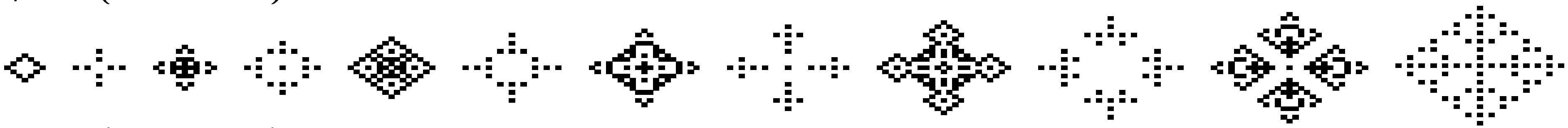
+ 4 (00000100)



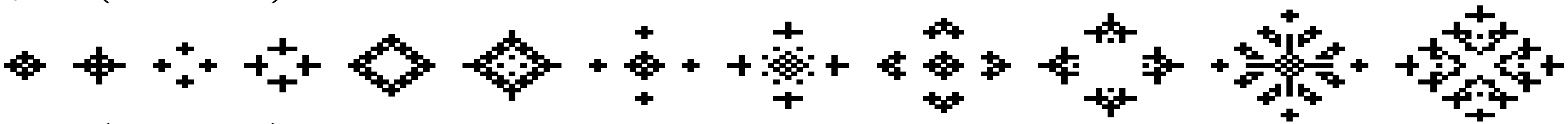
18 (00010010)



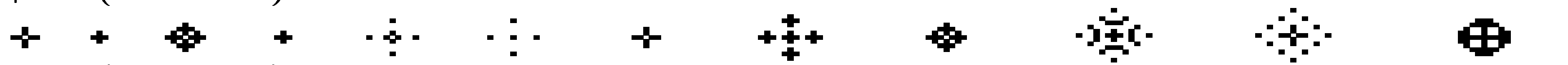
+ 22 (00010110)



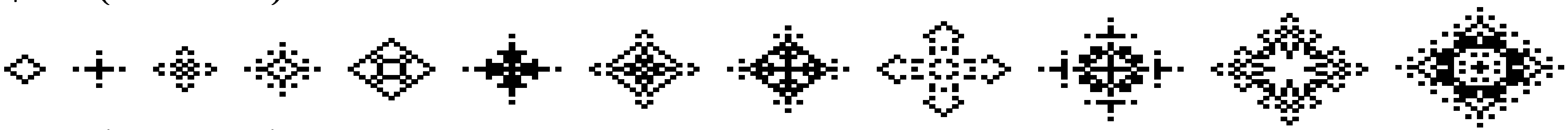
+ 36 (00100100)



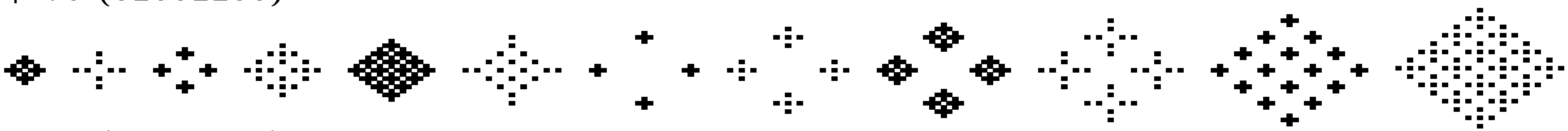
+ 50 (00110010)



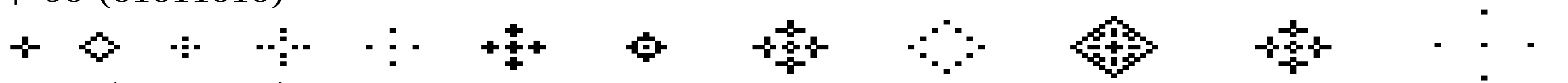
+ 54 (00110110)



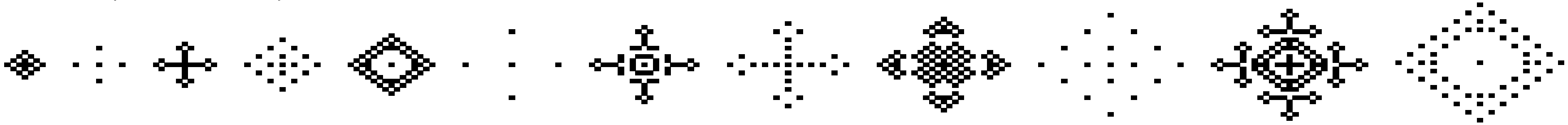
+ 76 (01001100)



+ 90 (01011010)



+150 (10010110) PARITY



Ramon Alonso-Sanz

REVERSIBLE CA (Fredkin): $\sigma_i^{(T+1)} = \phi\left(\left\{\sigma_{j\in\mathcal{N}_i}^{(T)}\right\}\right) \ominus \sigma_i^{(T-1)}$

EMBEDDED MEMORY: $\sigma_i^{(T+1)} = \phi\left(\left\{s_{j\in\mathcal{N}_i}^{(T)}\right\}\right) \ominus \sigma_i^{(T-1)}$

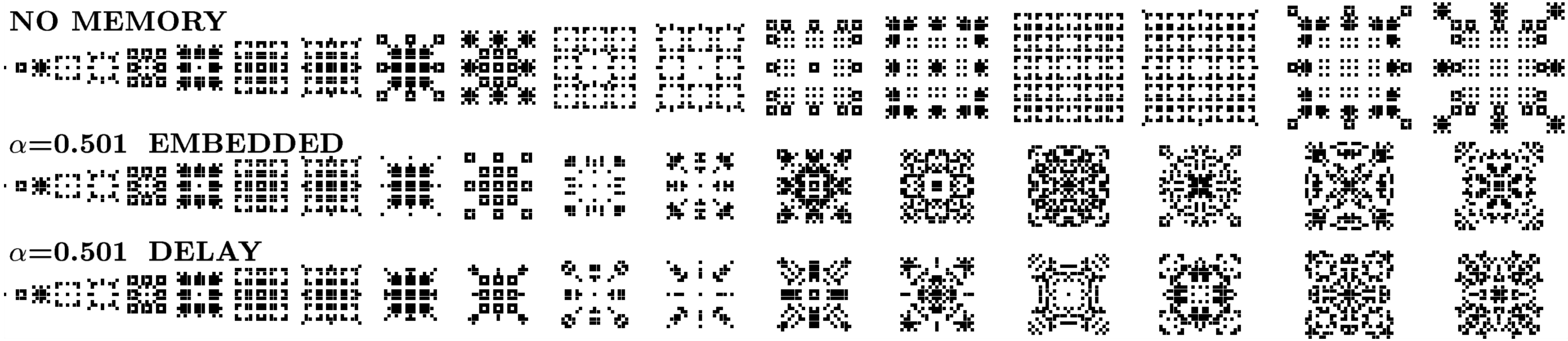
DELAY MEMORY: $\sigma_i^{(T+1)} = s\left(f_i^{(1)},\ldots,f_i^{(T)}\right) \ominus \sigma_i^{(T-1)}$

Reversion: $\sigma_i^{(T-1)} = \phi\left(\left\{\sigma_{j\in\mathcal{N}_i}^{(T)}\right\}\right) \ominus \sigma_i^{(T+1)}$

$$\sigma_i^{(T-1)} = \phi\left(\left\{s_{j\in\mathcal{N}_i}^{(T)}\right\}\right) \ominus \sigma_i^{(T+1)}$$
$$\omega_i^{(T-1)} = (\omega_i^{(T)} - \sigma_i^{(T)})/\alpha$$

$$\sigma_i^{(T-1)} = s\left(f_i^{(1)},\ldots,f_i^{(T)}\right) \ominus \sigma_i^{(T+1)}$$
$$\omega_i^{(T-1)} = (\omega_i^{(T)} - f_i^{(T)})/\alpha$$

Rule 682(1010101010)



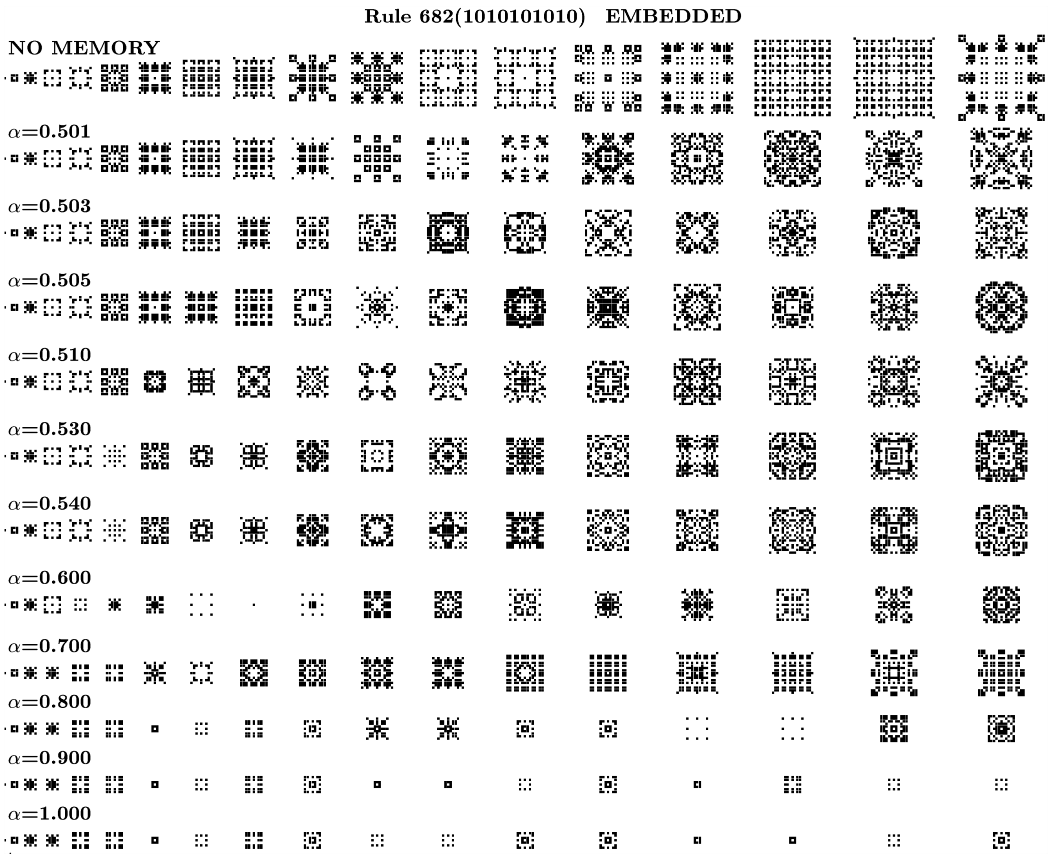
collidoscope.com/modernca/reversablerules.html *sjsu.rudyrucker.com/*

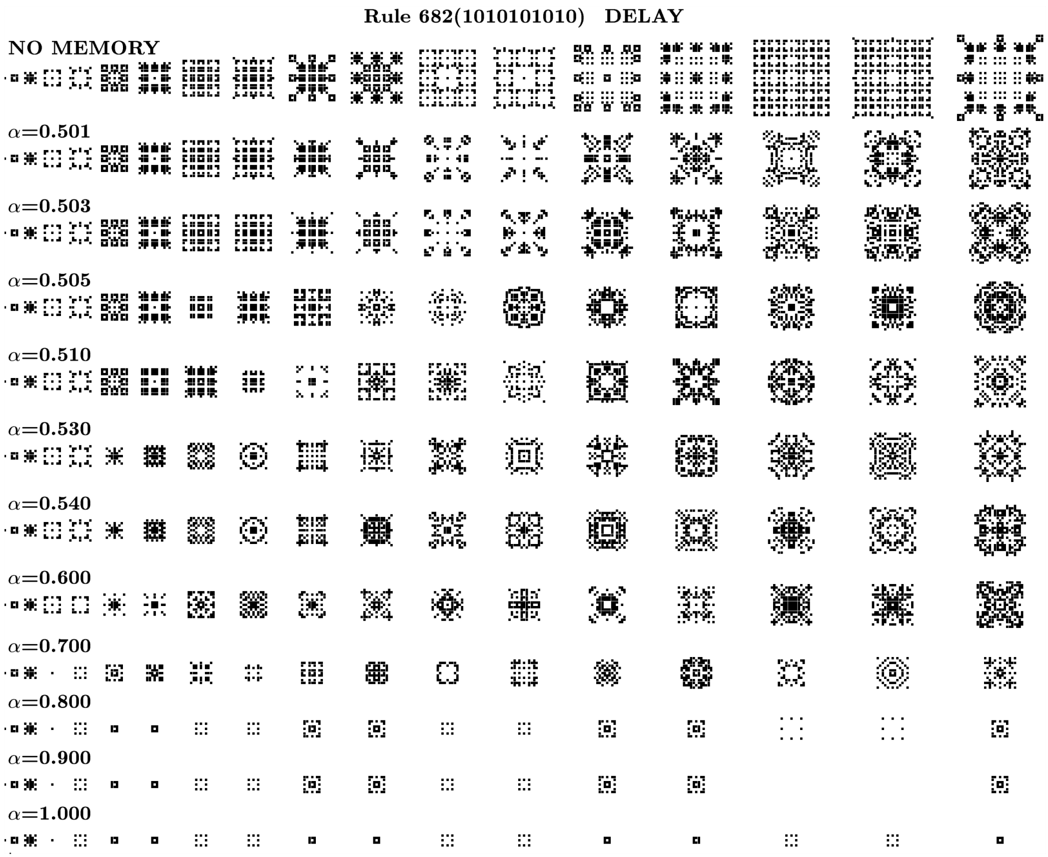
Alonso-Sanz,R.(2003). Reversible Cellular Automata with Memory: patterns starting with a single site seed. ***Physica D***, 175,1/2.
Alvarez,G. et al.(2005). A secret scheme to share color images. ***Computer Physics Comm.***, 173,1/2.

$0 \underset{\ominus}{\oplus} 0=0 \quad 1 \underset{\ominus}{\oplus} 0=0 \quad 1 \underset{\ominus}{\oplus} 1=0 \quad 0 \underset{\ominus}{\oplus} 1=1$

Ramon Alonso-Sanz

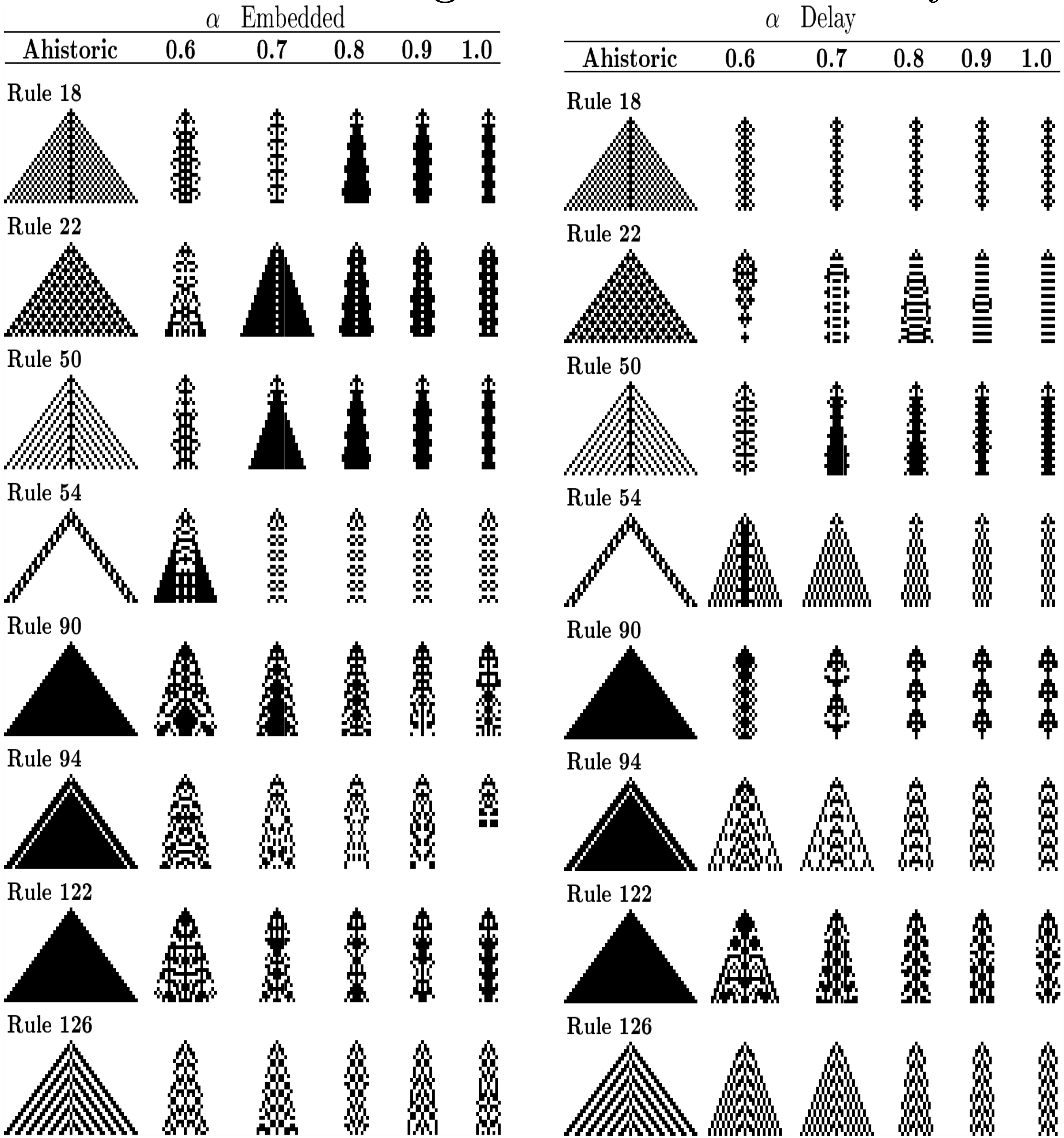
Reversible Parity Rule ($\{\sigma^{(0)}\} = \{\sigma^{(1)}\}$)





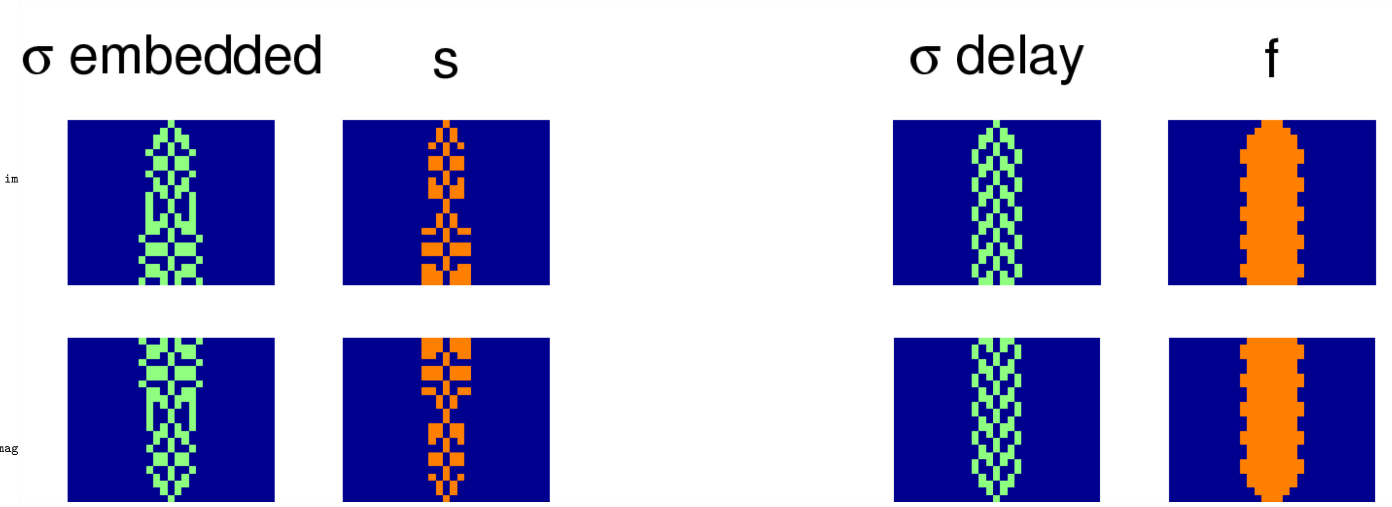
Ramon Alonso-Sanz

Reversible legal ECA with α-memory



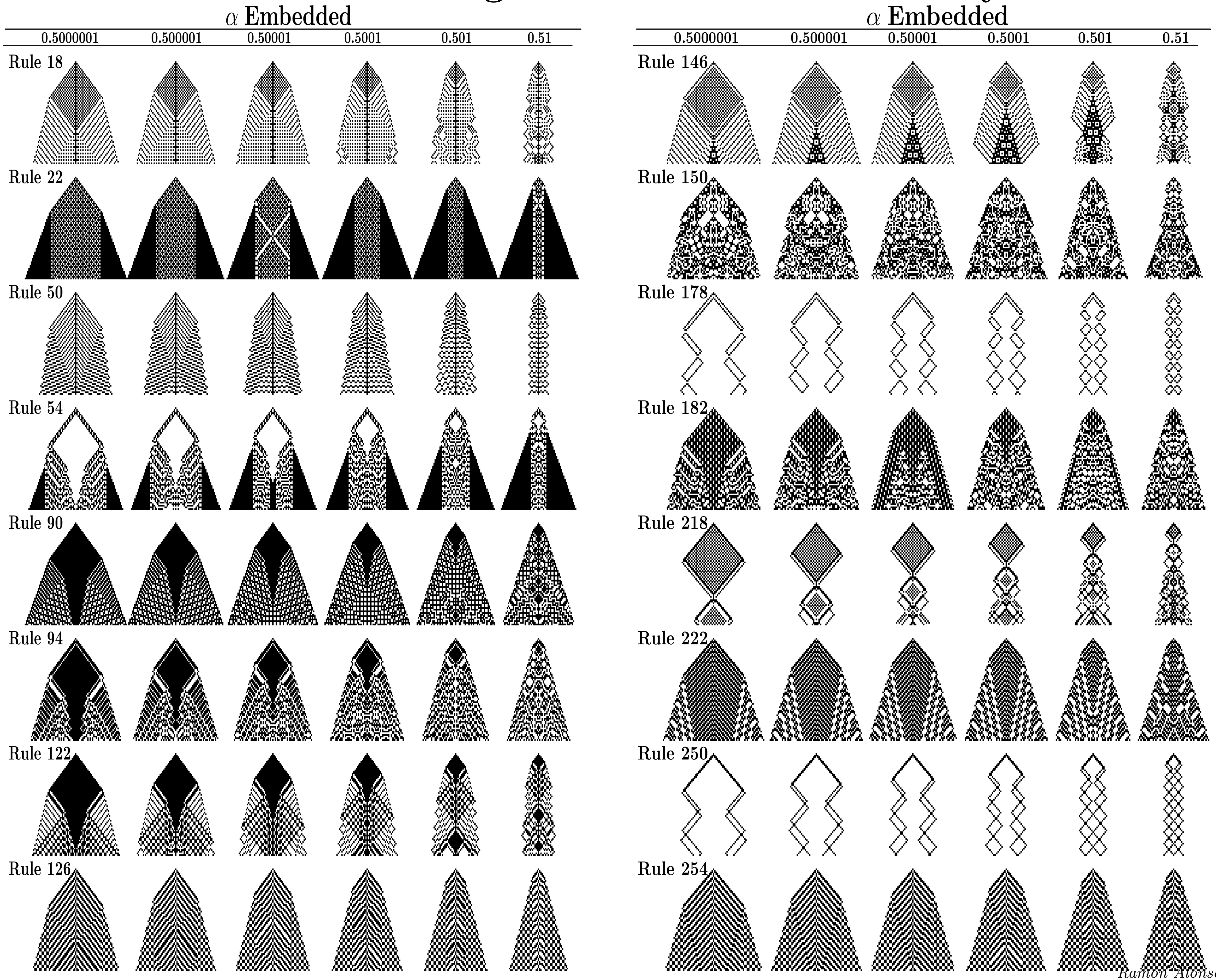
Ramon Alonso-Sanz

```
function cam
SR=254;T=8;N=2*T+1;nat=6;plus=3;
alfa(1)=0.5;alfa(2)=0.6;alfa(3)=0.7;alfa(4)=0.8;alfa(5)=0.9;alfa(6)=1.0;
[srb]=binarynumber(SR);left=[N 1:N-1];right=[2:N 1];
for memo=1:2
for nal=1:nat; alpha=alfa(nal);
    [SIGMA,OMEGA,omega]=init(T,N,alpha);XX=SIGMA;
    switch memo
     case 1    % Embedded
     for t=1:T
       SIGMAH(t,:)=SIGMA;X_ 1=XX;XX=SIGMA;if(t==T)XSIGMA=SIGMA;end
       S=SIGMA;omega=(alpha*omega)+SIGMA;OMEGAX=OMEGA(t);
       for i=1:N
         if(2*omega(i)>OMEGAX)S(i)=1;end;    % memory
         if(2*omega(i)<OMEGAX)S(i)=0;end
       end
       [SIGMA]=RULE(S,N,srb,left,right);    % rule
       SIGMA=mod(SIGMA+X_ 1,2);
     end
     subplot(5+nat*plus,nat+plus,(nat+plus)*(memo-1)+nal);image(33*SIGMAH);axis('off');axis im
                                  % reversion
     XX=SIGMA;SIGMA=XSIGMA;
     for t=1:T
       X_ 1=XX;SIGMAH(t,:)=SIGMA; XX=SIGMA;
       if(t>1)
         S=SIGMA;OMEGAX=OMEGA(T-t+1);
         if(alpha>0);omega=(omega-X_ 1)/alpha;
         for i=1:N
          if(2*omega(i)>OMEGAX)S(i)=1;end    % back-memory
          if(2*omega(i)<OMEGAX)S(i)=0;end
         end
       end
       end
       [SIGMA]=RULE(S,N,srb,left,right);    % rule
       SIGMA=mod(SIGMA-X_ 1,2);
     subplot(5+nat*plus,nat+plus,(nat+plus)*(memo)+nal);image(33*SIGMAH);axis('off');axis imag
     case 2    % delay
       [SIGMA,OMEGA,omega]=init(T,N,alpha);S=SIGMA;XX=SIGMA;
        for t=1:T
       SIGMAH(t,:)=SIGMA;X_ 1=XX;XX=SIGMAif(t==T)XSIGMA=SIGMA;end
       [S]=RULE(SIGMA,N,srb,left,right);    % rule
       SIGMA=S;HS(t,:)=S;
       omega=(alpha*omega)+SIGMA;OMEGAX=OMEGA(t);
       for i=1:N
         if(2*omega(i)>OMEGAX)SIGMA(i)=1;end   % memory
         if(2*omega(i)<OMEGAX)SIGMA(i)=0;end
       end
       SIGMA=mod(SIGMA+X_ 1,2);
     end
     subplot(5+nat*plus,nat+plus,(nat+plus)*(memo+2)+nal);image(33*SIGMAH);axis('off');axis image;
                                  % reversion
     XX=SIGMA;SIGMA=XSIGMA;
     for t=1:T
       X_ 1=XX;SIGMAH(t,:)=SIGMA;XX=SIGMA;
       [S]=RULE(SIGMA,N,srb,left,right);    % rule
       SIGMA=S;
       if(alpha>0)
         OMEGAX=OMEGA(T-t+1);
         for i=1:N
         if(2*omega(i)>OMEGAX)SIGMA(i)=1;end    % back-memory
         if(2*omega(i)<OMEGAX)SIGMA(i)=0;end
         end
         omega=(omega-S)/alpha;
         end
       SIGMA=mod(SIGMA-X_ 1,2);
       end
     end
     subplot(5+nat*plus,nat+plus,(nat+plus)*(memo+2)+nal);image(33*SIGMAH);axis('off');axis image;
end
end
print carevmemory.eps -depsc

function [SIGMA]=RULE(S,N,srb,left,right);
       for i=1:N
        SIGMA(i)=srb(8-(4*S(left(i))+2*S(i)+S(right(i))));
       end

function [SIGMA,OMEGA,omega]=init(T,N,alpha);
        SIGMA(1:N)=0; SIGMA((N+1)/2:(N+1)/2)=1;
        OMEGA(1)=1.0;omega(1:N)=0;
        for t=2:T;OMEGA(t)=1+alpha*OMEGA(t-1);end
function [BN] =binarynumber(rule);
        BN(1:8)=0;irtx=rule;
        for ix=1:8
          rest=mod(irtx,2);ratio=(irtx-rest)/2;BN(8-ix+1)=rest;irtx=ratio;
        end
```



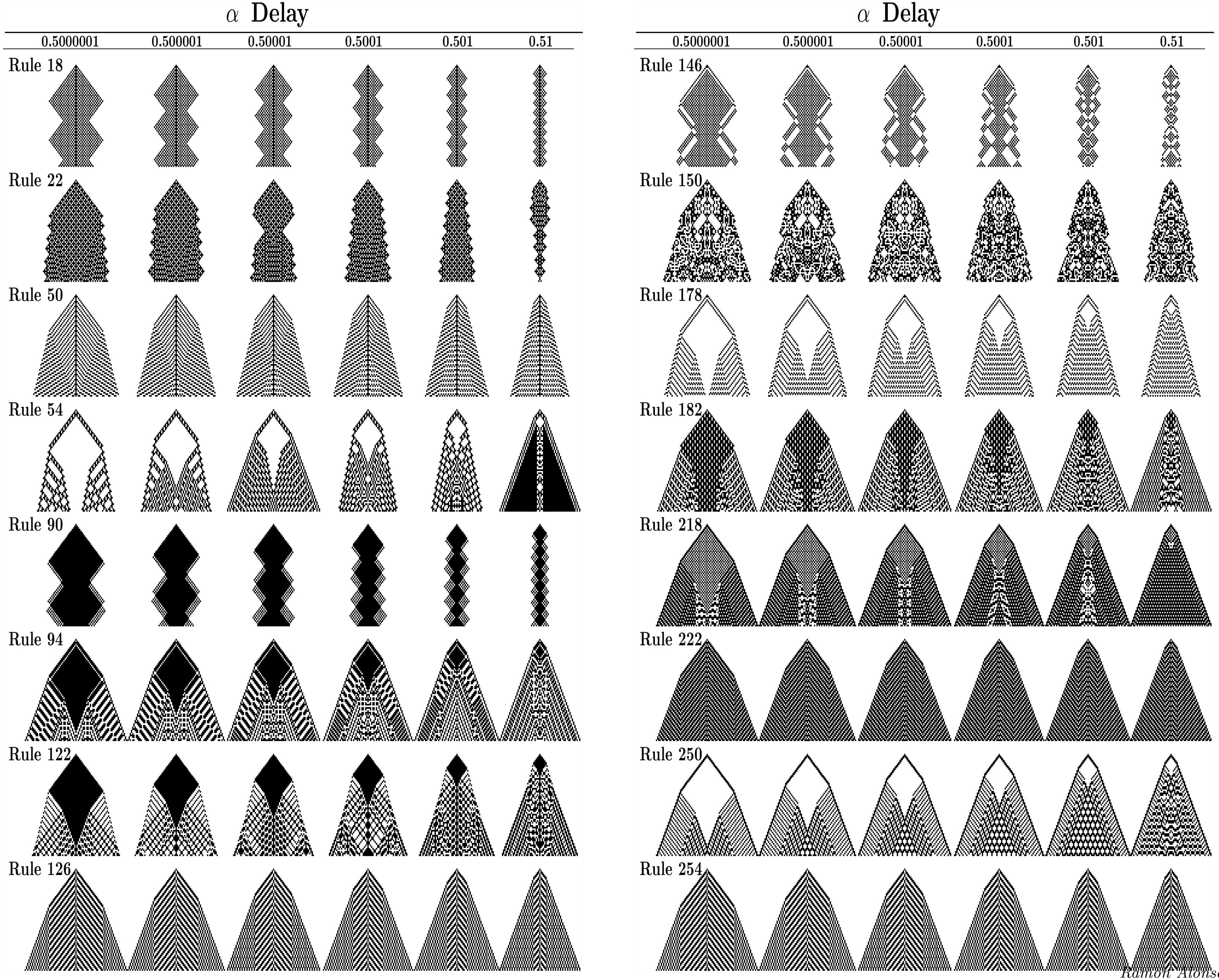
Ramon Alonso-Sanz

Reversible legal ECA with low α-memory



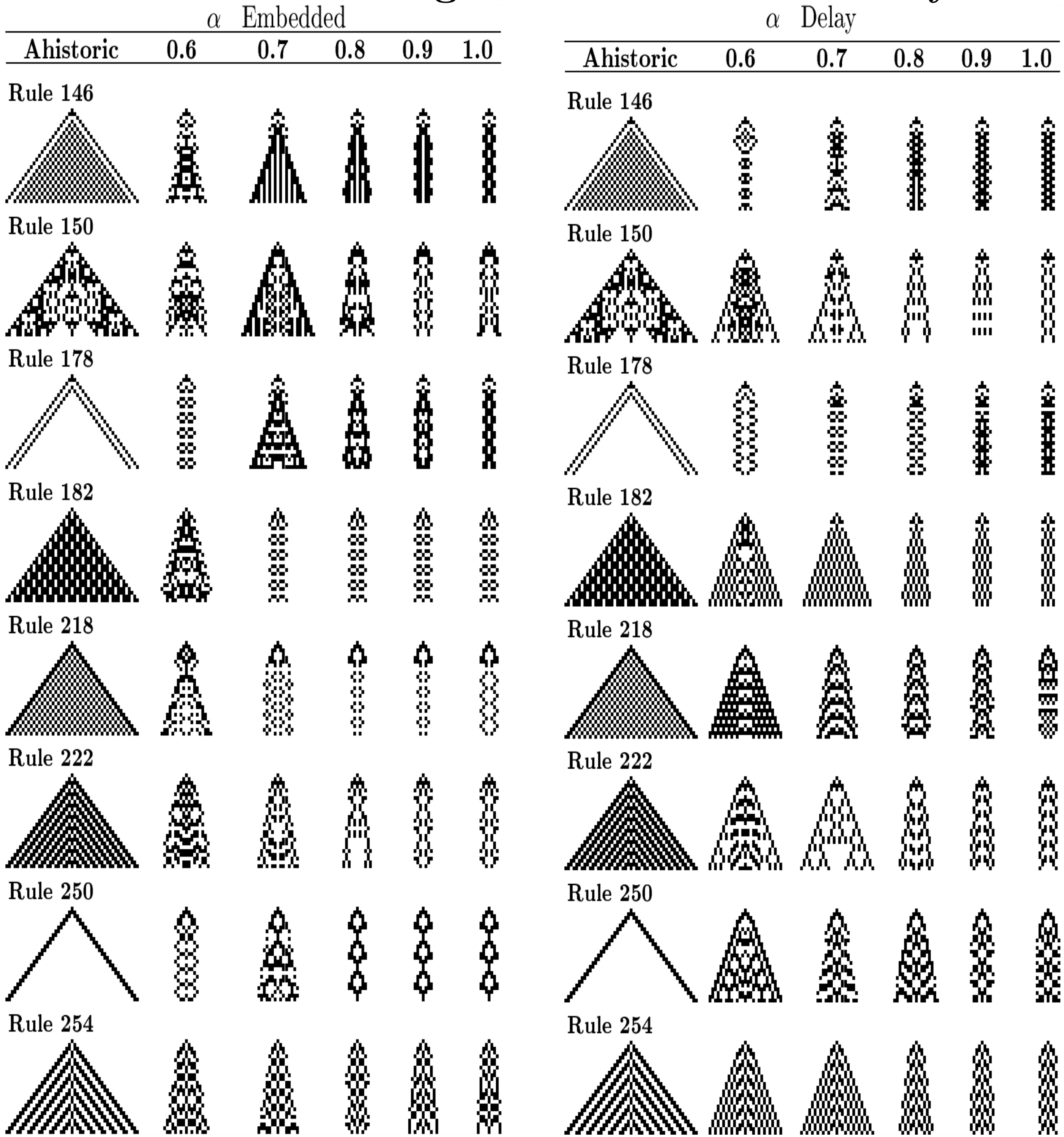
Ramón Alonso-Sanz

Reversible legal ECA with low α-memory

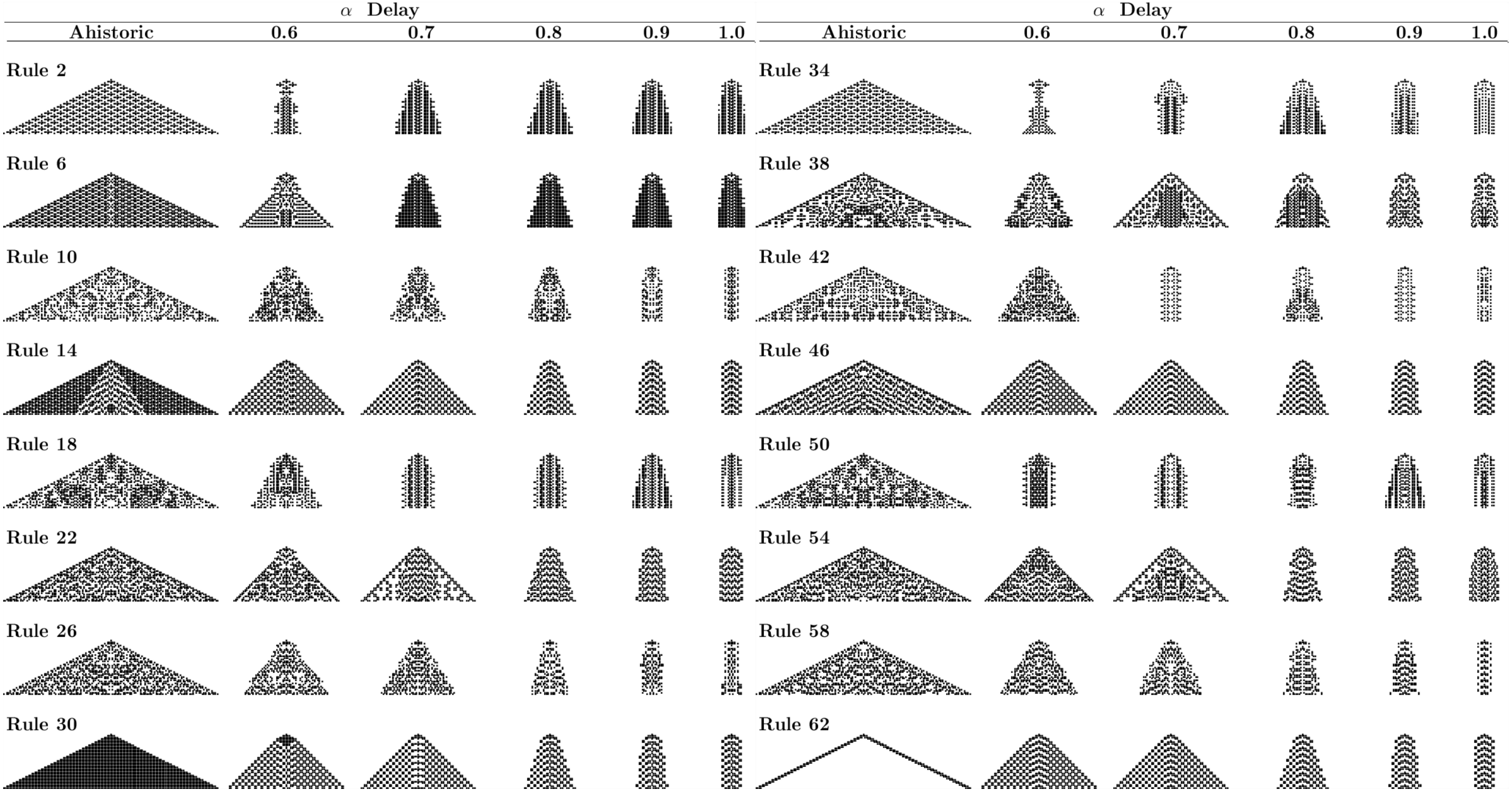


Ramón Alonso-Sanz

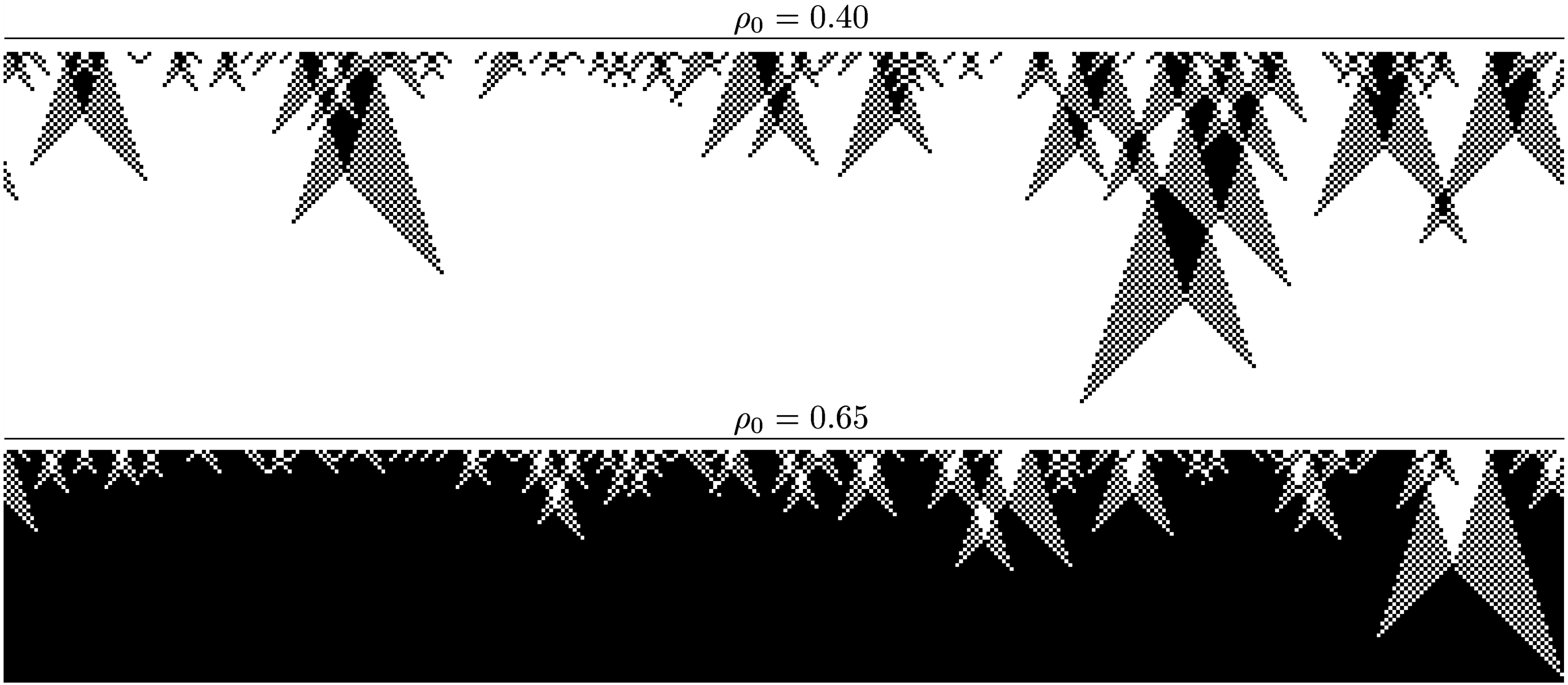
Reversible legal ECA with α-memory



1D r=2 CA

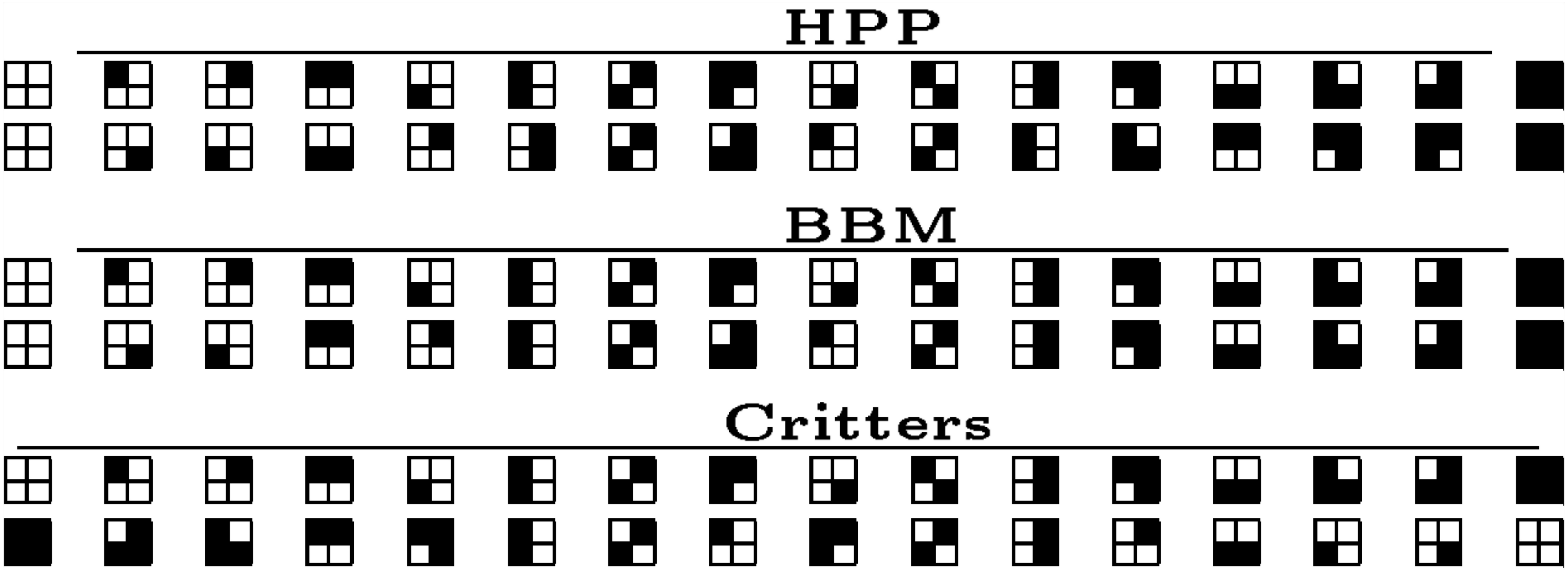


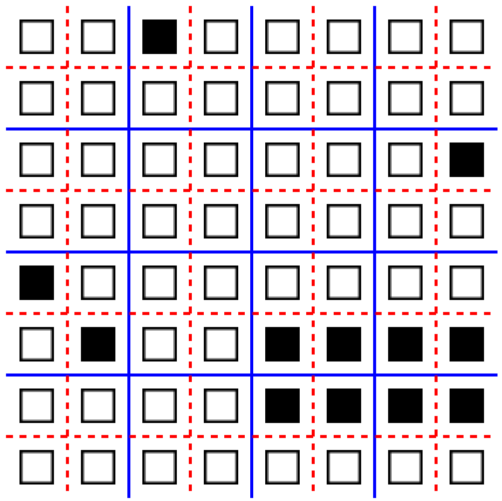
Ramon Alonso-Sanz



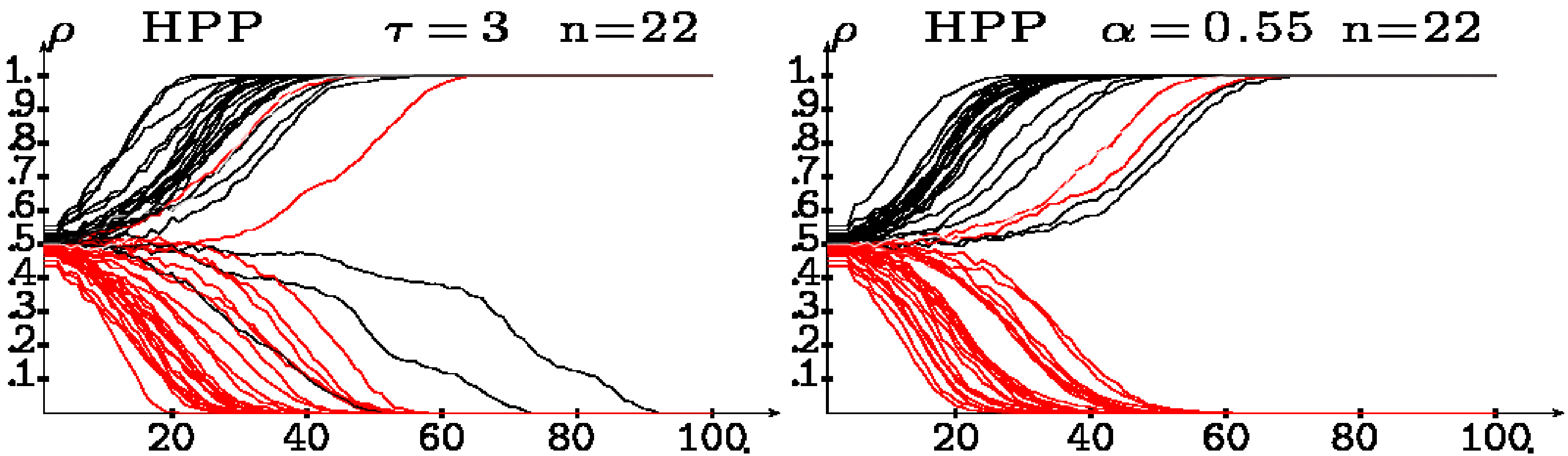
Rule III block CA with τ=3 majority memory, starting at random.

Two-dimensional block cellular automata





The HPP block CA rule with delay memory and the Density Classification Task



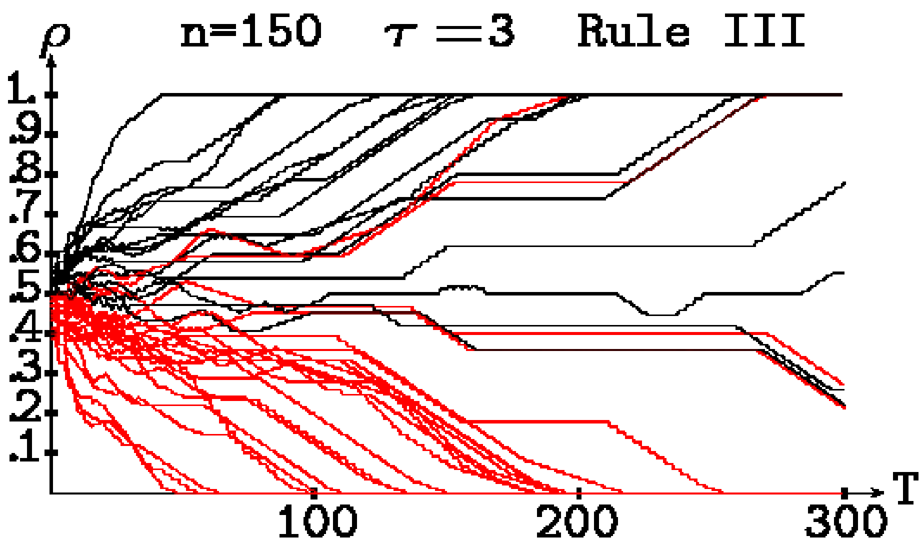
Evolution of density with low delay memory. Fifty simulations.

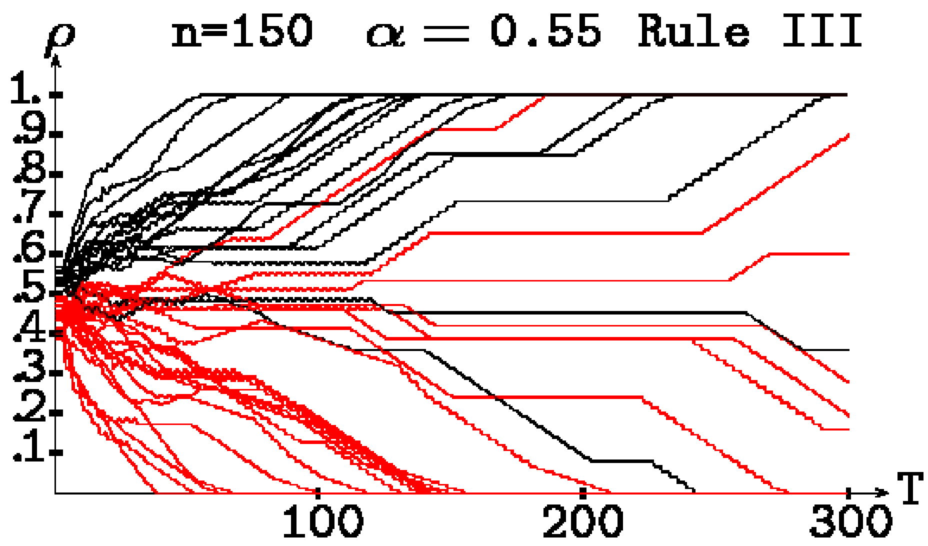
Percentage of correctly classified densities and average time up to convergence. 2D lattices of size n×n. 10^5 binomially generated ICs.

	τ =3	τ =4	τ =5	τ =6	τ =7	α =0.55	α =0.60	α =0.65	α =0.70	α =0.75
n=22	88.732 42	**92.840 41**	87.467 51	89.804 46	86.715 61	92.988 43	**94.212 40**	92.109 39	90.786 43	90.552 48
n=21	82.762 87	**87.331 87**	81.892 105	84.152 100	81.472 127	**88.601 99**	86.771 92	85.951 95	84.892 97	84.512 114
n=32	87.297 65	**91.362 62**	85.853 78	87.840 73	84.417 93	**92.378 59**	92.159 61	90.319 61	88.752 68	88.331 77
n=31	81.782 136	**84.712 136**	80.682 164	82.972 157	80.272 195	**85.770 149**	84.588 141	84.357 150	83.241 152	82.815 182
n=42	86.101 88	**89.818 85**	84.426 105	86.449 100	83.665 126	**91.442 80**	90.610 85	88.836 86	86.916 96	86.813 107
n=41	80.272 186	**83.552 186**	79.432 220	82.082 213	78.472 248	**83.792 200**	83.392 191	83.202 206	82.402 208	81.682 244

Alonso-Sanz,R.(2013). Cellular automaton with memory and the density classification task. *J. of Cellular Automata* **(in press)**

The Rule III block CA rule with delay memory and the DCT





Evolution of density with low delay memory. Fifty simulations.

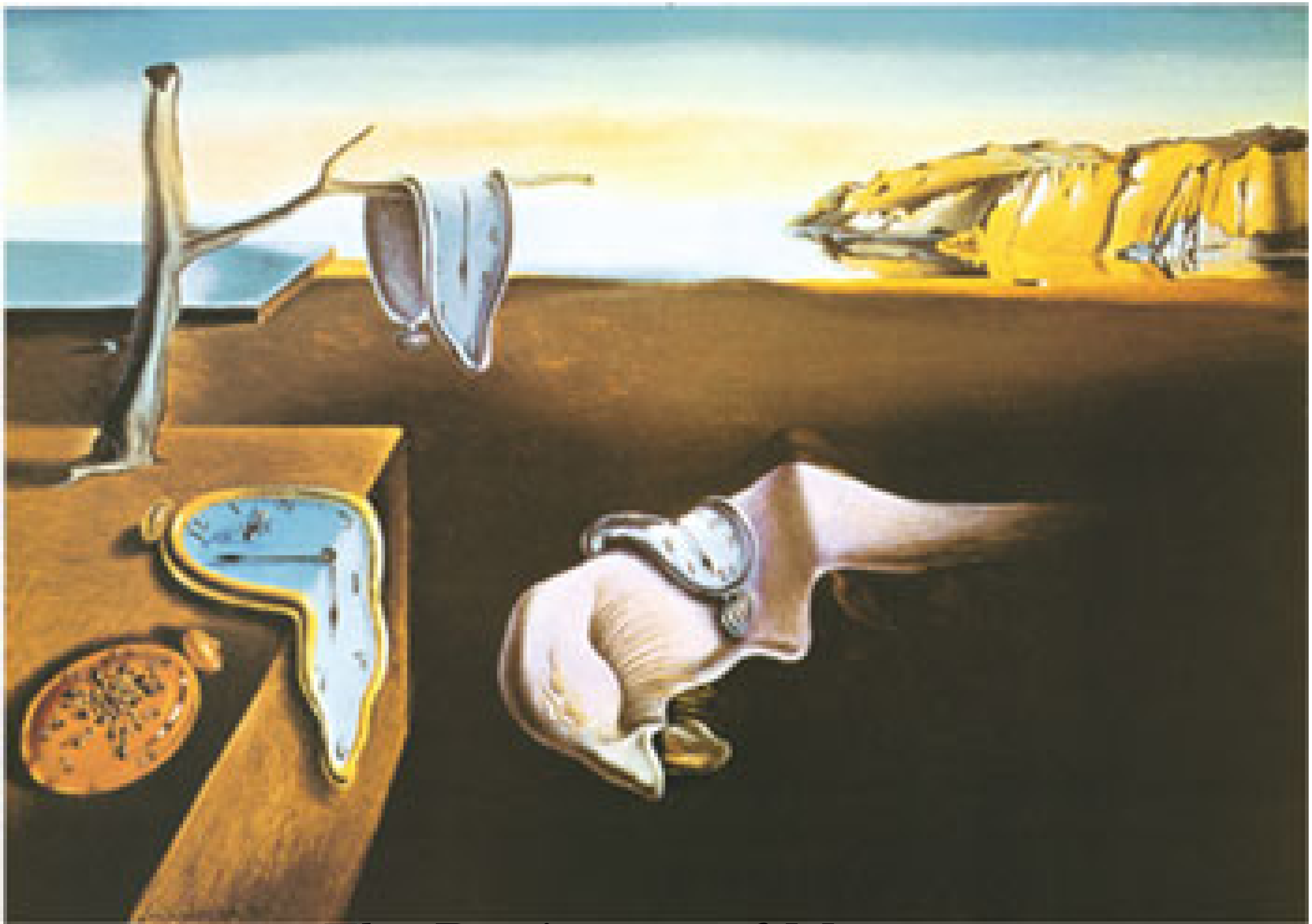
Percentage of correctly classified densities and average time up to convergence in n-size registers. 10^5 **binomially generated ICs.**

	τ =3	τ =4	τ =5	τ =6	τ =7
n=150	85.042 185	**85.669 180**	82.448 213	82.804 206	81.510 248
n=149	81.489 295	**81.921 289**	79.073 339	79.179 331	78.028 397

	α =0.51	α =0.55	α =0.60	α =0.65	α =0.70
n=150	**88.374 174**	86.385 178	86.096 180	84.329 197	83.433 205
n=149	**83.862 282**	82.462 286	82.282 288	80.371 319	79.687 328

Alonso-Sanz,R.(2013). Cellular automaton with memory and the density classification task. *J. of Cellular Automata* **(in press)**

SALVADOR DALI



The Persistence of Memory

Disintegration of the Persistence of Memory