Computing by Temporal Order:
Asynchronous Cellular Automata

Michael Vielhaber

Universidad Austral de Chile, Instituto de Matemáticas, Casilla 567, Valdivia, Chile
Hochschule Bremerhaven, FB2, An der Karlstadt 8, D–27568 Bremerhaven, Germany
vielhaber@gmail.com

Our concern is the behaviour of the elementary cellular automata with state set \( \{0, 1\} \) over the cell set \( \mathbb{Z}/n\mathbb{Z} \) (one-dimensional finite wrap-around case), under all possible temporal rules (asynchronicity).

Over the torus \( \mathbb{Z}/n\mathbb{Z} \) (\( n \leq 10 \)), we will see that the ECA with Wolfram update rule 57 maps any \( v \in \mathbb{F}_2^n \) to any \( w \in \mathbb{F}_2^n \), varying the temporal rule.

We furthermore show that all even (element of the alternating group) bijective functions on the set \( \mathbb{F}_2^n \cong \{0, \ldots, 2^n - 1\} \), can be computed by ECA-57, by iterating it a sufficient number of times with varying temporal rules, at least for \( n \leq 10 \). We characterize the non-bijective functions computable by asynchronous rules.

The thread of all this is a novel paradigm:

The algorithm is neither hard-wired (in the ECA), nor in the program or data (initial configuration), but in the temporal order of updating cells, and temporal order is pattern-universal.

**Keywords:** Cellular automata, asynchronous, update rule, universality.

1 Introduction and Notation, Asynchronicity

We consider elementary cellular automata, i.e. with state set \( S = \mathbb{F}_2 = \{0, 1\} \) and update neighborhood \((c_{i-1}, c_i, c_{i+1})\) for cell \( c_i \).

The cell index (site) \( i \) will come from \( \mathbb{Z}/n\mathbb{Z} \) for some \( n \geq 3 \), i.e. we consider the finite one-dimensional torus, indices wrap around. In Section 2, we consider patterns “How universal can a mapping on \( \mathbb{F}_2^n \) become?”; and Section 3 covers functions \( \mathbb{F}_2^n \ni v \mapsto w \in \mathbb{F}_2^n \).

The 256 ECA’s group into 88 classes under the symmetries 0/1 and left/right neighbor, see Appendix A. It is sufficient to consider one member per class.

The Wolfram rule ECA = \( \sum_{k=0}^{7} 2^k \cdot p_k \in \{0, \ldots, 255\} \) defines the behavior. A cell with neighborhood \((c_{i-1}, c_i, c_{i+1})\) \( \in \mathbb{F}_2^3 \), summing up to \( k := 4c_{i-1} + 2c_i + c_{i+1} \in \{0, \ldots, 7\} \) is replaced by \( c_i^+ := p_k \).

**Example 1** The behaviour of the ECA with Wolfram rule 57\(_{10} = 00110001_2 \) is given in Table 1. We have that \( 0c_i \mapsto c_i \), all other cases \( 0c_i, 1c_i, 0c_i, 1c_i \mapsto c_i \).

|   |   |
|---|---|
| 111 | 0 |
| 110 | 0 |
| 101 | 1 |
| 100 | 1 |

Table 1: ECA-57

\*Partially funded by HS Bremerhaven, Germany through a sabbatical leave.
1.1 State-of-the-Art

The study of asynchronous cellular automata started with Ingrerson and Buvel’s 1984 paper [2].

Lee et al. [3] give an asynchronous CA on the two-dimensional grid $\mathbb{Z} \times \mathbb{Z}$, which is Turing-universal. Fatès et al. [1] consider ECA’s with quiescent states (000 $\rightarrow$ 0, 111 $\rightarrow$ 1, i.e. with even Wolfram rule $\geq 128$). They consider fully randomized ECA’s.

A good overview is given in the thesis [4] by Sharkar.

Nevertheless, all these articles treat asynchronous CAs with randomized clocks.

Our concern is instead the (fully deterministic) behavior of a suitable ECA, with any fixed initial configuration, under all possible temporal sequences. There seems to be no work on the combined effect of all deterministic temporal rules, synchronous and asynchronous, so far.

**Definition 1. Temporal Rules — Asynchronicity Rules**

Let the set $A_{n}$ of asynchronicity rules over $\mathbb{Z}/n\mathbb{Z}$ consist of all words of length $n$ over the alphabet $\{<,=,>\}$ such that both $<$ and $>$ occur at least once. We also include the word “$=\cdots=$”, the synchronous case, and have $A_{n} = \{\{<,=,>\}^{n} \setminus (\{<,=\}^{n} \cup \{=\}^{n}) \cup \{\}^{n}\}$ with $|A_{n}| = 3^{n} - 2^{n+1} + 2$.

Given a rule $A = A_{S}0 \cdots A_{S_{n-1}}$, $A_{S_{i}}$ = “$\langle\rangle$”, “$\equiv\rangle$”, and “$\equiv\langle\rangle$”, resp., defines that cell $c_{i}$ updates after, simultaneously with, resp. before $c_{i+1}$, $0 \leq i \leq n-2$. $A_{S_{n-1}}$ refers to cell $c_{n-1}$ with respect to $c_{0}$.

For any partition $(S_{1}, \ldots, S_{m})$ of the cell sites, i.e. $\bigcup_{k=1}^{m} S_{k} = \{0, \ldots, n-1\}$, let its temporal rule be $A_{S_{i}} = <, =, >$, resp., if $i \in S_{(i)}$, $i + 1 \in S_{(i+1)}$, and $t(i)$ is $>, =, <$, resp., than $t(i + 1)$ (we say that site $i$ is “bigger” if it comes before $i+1$, hence dominates it).

With the exception of $\equiv^{n}$ (synchronous case), both $<$ and $>$ must occur at least once, since otherwise, by wrapping-around, each cell would update only after itself and the temporal rule would thus not be well-defined, e.g. $\equiv<\equiv$ leads to $c_{1}$ with $c_{2}$ after $c_{3}$ with $c_{1}$ after, and thus before, itself.

**Example 2** Let $n = 4$, and $A = "<\langle\rangle\rangle":$ Cell 0 updates after cell 1, 1 with 2, 2 before 3, and 3 before 0. Hence the temporal order is $(1,2|3|0)$, first 1 and 2 simultaneously, then 3, finally cell 0, i.e. $S_{1} = \{1,2\}, S_{2} = \{3\}, S_{3} = \{0\}$. Analogously, “$\langle\rangle\langle\rangle"$ leads to $(1,3|0,2)$, and “$\equiv\langle\rangle\langle\rangle"$ leads to $(0|1,2,3)$.

One might be inclined to partition the $n$ cells into sets $S_{1}, \ldots, S_{m} \subset \mathbb{Z}/n\mathbb{Z}$, and update those in $S_{1}$ first, then cells from $S_{2}$ and so forth. This, however, is too fine-grained:

**Theorem 1**

Consider two partitions $(S_{1}, \ldots, S_{m})$ and $(S'_{1}, \ldots, S'_{m'})$ of the cell set $\mathbb{Z}/n\mathbb{Z}$ and define functions $a, b, c, a', b', c'$ such that

$$\forall i \in \{0, \ldots, n-1\} : i - 1 \in S_{a(i)}, i \in S_{b(i)}, i + 1 \in S_{c(i)}.$$

Then, if $\text{sgn}(a(i) - b(i)) = \text{sgn}(a'(i) - b'(i))$ and $\text{sgn}(b(i) - c(i)) = \text{sgn}(b'(i) - c'(i)), \forall i$, i.e. the relative update order of cells $i - 1, i, i + 1$ is the same for $S$ and $S'$, then updating according to $S$ or according to $S'$ leads to the same result, and this is described by the following asynchronicity rule (Table 2).

**Proof.** By construction. Since the relative temporal order of cell $c_{i}$ with respect to $c_{i-1}$ and $c_{i+1}$ is the same for $(S_{k})$ and $(S'_{k})$, by $\text{sgn}(a(i) - b(i)) = \text{sgn}(a'(i) - b'(i))$ and $\text{sgn}(b(i) - c(i)) = \text{sgn}(b'(i) - c'(i))$, both partitions lead to the same overall behaviour, which is described by $A_{S}$.

The construction by the theorem shows that the $A \in A_{S}$ are sufficient to distinguish the behaviour. On the other hand, all these $A$ are necessary and can lead to different behaviour (at least for some ECA’s), since any $A_{S_{i}} \neq A_{S'_{i}}$ will lead to a different order of updating cells $c_{i}$ and $c_{i+1}$.
\[
\begin{array}{cc|cc}
\text{sgn}(a-b) & \text{sgn}(b-c) & \text{AS}_{i-1} & \text{AS}_i \\
-1 & -1 & > & > \\
-1 & 0 & > & \equiv \\
-1 & +1 & > & < \\
0 & -1 & \equiv & > \\
0 & 0 & \equiv & \equiv \\
0 & +1 & \equiv & < \\
+1 & -1 & < & > \\
+1 & 0 & < & \equiv \\
+1 & +1 & < & < \\
\end{array}
\]

Table 2: Local asynchronicity

**Example 3** For \( n = 6, "<><><>" \) requires the odd cells 1,3,5 to update before the even ones 0,2,4. There are 13 partitions of three elements, e.g. \((1,3,5),(1|3,5),(1,5|3)\), and \((5|1|3)\), and thus \(13^2 = 169\) partitions \(S_i\) for this AS.

**Definition 2.** By \(ECA_{AS}(v) = w\), we mean that the elementary CA with rule ECA maps \(v \in \{0,1\}^n\) to \(w \in \{0,1\}^n\) via the temporal sequence \(AS\).

**Example 4** \(ECA-57<><>(1000) = 1110\), in two steps: \(10\overline{0}0 \mapsto \overline{1}100 \mapsto 1110\), where underlined cells are active in the next step.

## 2 The Finite Torus \(\mathbb{Z}/n\mathbb{Z}\): Patterns

In this section, we work on the torus \(\mathbb{Z}/n\mathbb{Z}\), and consider all ECA’s for all initial configurations. We apply a fixed temporal rule \(AS \in AS_n\) repeatedly, \(\tau\) times, and ask, whether these 5 pattern universality properties hold:

\[
\begin{align*}
(0) & \exists v \in F_2^n, \forall w \in F_2^n, \exists \tau \in \mathbb{N}, \ldots \\
(i) & \forall v \in F_2^n, \forall w \in F_2^n, \exists \tau \in \mathbb{N}, \ldots \\
(ii) & \forall v \in F_2^n, \exists \tau \in \mathbb{N}, \forall w \in F_2^n, \ldots \\
(iii) & \exists \tau \in \mathbb{N}, \forall v \in F_2^n, \forall w \in F_2^n, \ldots \\
(iv) & \exists \tau_0 \in \mathbb{N}, \forall \tau \geq \tau_0, \forall v,w \in F_2^n, \ldots \\
\end{align*}
\]

All results are experimental i.e. derived from exhaustive computer simulations for the stated lengths.

We start with

\(0) \exists v \in F_2^n, \forall w \in F_2^n, \exists \tau \in \mathbb{N}, \exists AS \in AS_n : ECA_{AS}^\tau(v) = w.\) That is from some \(v\) we eventually reach any \(w\). We give the largest number of \(w\)’s reached for some \(v\), for \(n = 4, 8, 12\). To satisfy \(0), these must be \((16, 256, 4096)\).

The 3 ECA families \(0\) \((1,1,1)\), \(200\) \((1,1,1)\), and \(204\) \((1,1,1)\) are resilient to asynchronicity. They have a constant result, for all \(n\).

\(ECA-51\) \((2,2,2)\) varies between at most two results.
The next 49 ECA families are ordered by increasing image size for \( n = 12 \):

\[
\begin{array}{c@{\quad}c@{\quad}c@{\quad}c@{\quad}c@{\quad}c@{\quad}c@{\quad}c}
140 & (2,6,16), & 160 & (12,130,1182), & 164 & (13,197,2930), & 108 & (16,256,4052), \\
136 & (2,9,27), & 2 & (11,211,1477), & 24 & (15,211,2961), & 56 & (16,256,4066), \\
128 & (2,16,49), & 72 & (11,131,1499), & 34 & (13,209,2998), & 74 & (15,255,4071), \\
132 & (4,18,81), & 76 & (11,131,1499), & 130 & (14,211,3160), & 73 & (16,256,4084), \\
32 & (7,31,127), & 172 & (11,137,1506), & 94 & (16,216,3448), & 33 & (16,256,4092), \\
8 & (5,45,320), & 168 & (12,147,1601), & 152 & (14,237,3561), & 10 & (13,253,4093), \\
4 & (7,47,322), & 13 & (16,168,1792), & 138 & (13,238,3751), & 134 & (15,255,4093), \\
12 & (7,47,322), & 232 & (12,156,1830), & 104 & (14,232,3824), & 42 & (15,255,4093), \\
28 & (11,91,641), & 77 & (12,156,1830), & 162 & (16,250,3970), & 35 & (16,256,4094), \\
29 & (12,92,642), & 142 & (12,140,1848), & 170 & (16,256,3976), & 43 & (16,256,4094), \\
44 & (12,100,870), & 78 & (15,167,1851), & 15 & (16,256,3976), \\
156 & (4,64,1024), & 36 & (14,162,1943), & 150 & (12,240,4032), \\
40 & (11,119,1052), & 5 & (16,216,2542), & 1 & (16,256,4051), \\
\end{array}
\]

The 4 ECA families 6, 14, 18 [for \( n \geq 7 \)], 50 [for \( n \geq 4 \)], miss exactly one pattern, leading to \( 2^n - 1 \) in general.

Finally, the 31 ECA families

\[
3, 7, 9, 11, 19, 22, 23, 25, 26, 27, 30, 37, 38, 41, 45, 46, 54, 57, 58, 60, 62, 90,
105 \text{ [} n \not\equiv 0 \mod 4 \text{]}, 106, 110, 122, 126, 146, 154, 178 \text{ [} n \not\equiv 3 \text{]}, 184 \text{ [} n \not\equiv 3 \text{]},
\]

satisfy property \( \text{(a)} \) (for \( 3 \leq n \leq 12 \)).

(i) \( \forall v, w \in \mathbb{F}_2^n, \exists \tau \in \mathbb{N}: \text{ECA}_{A_S}^\tau(v) = w \). From the 31 families satisfying \( \text{(a)} \), most fall short for some \( v \). We give the smallest number of \( w \) reachable from some \( v \), for \( n = 4, 8, \) and \( 12 \), this should be \( (16,256,4096) \) to satisfy \( \text{(i)} \).

Eighteen ECA families are insensitive (or resilient) to asynchronicity for at least some \( v \), the same \( w \) resulting for all \( A_S \). Hence, \( (1,1,1) \) patterns are reached:

\[
22, 26, 30, 38, 46, 54, 58, 60, 62, 90, 106, 110, 122, 126, 146, 154, 178, 184.
\]

ECA family 7 reaches \( 2^n - 1 \) for \( n \not\equiv 0 \mod 3 \) and only 1 pattern for \( n \equiv 0 \mod 3 \).

ECA family 45 has \( 2^n - 1 \) patterns for odd \( n \), 1 for even \( n \).

Six ECA families get near the full \( 2^n \) for all \( w \): 3 \( (15,233,3411) \), 9 \( (12,243,3963) \), 11 \( (15,233,3515) \), 25 \( (16,251,4031) \), 27 \( (16,253,4052) \), 43 \( (12,236,3554) \).

The following 6 ECA families satisfy \( \text{(i)} \) at least for certain \( [n] \) (\( 3 \leq n \leq 12 \) considered):

19 \( [3\text{-}12] \), 23 \( [3,5,7,9,11] \), 37 \( [4,5,7,8,10,11] \), 41\( [3,5,7,12] \), 57\( [3\text{-}12] \), 105 \( [3,5,7,9,11] \) all generate \( 2^n \) patterns for these \( [n] \).

(ii) – (iv) From now on, we will consider the 6 ECA families satisfying \( \text{(i)} \): 19, 23 \( (n \not\equiv 0 \mod 2) \), 37 \( (n \not\equiv 0 \mod 3) \), 41, 57, and 105 \( (n \not\equiv 0 \mod 4) \).

(iii) \( \forall v \in \mathbb{F}_2^n, \exists \tau \in \mathbb{N}, \forall w \in \mathbb{F}_2^n, \exists \tau \in \mathbb{N}: \text{ECA}_{A_S}^\tau(v) = w \); i.e. for fixed \( v \), all \( w \) are reached at the same time.

We considered \( \tau \) up to 20000, and obtain:

ECA-23: No \( v \) has any \( \tau \leq 20000 \) to satisfy \( \text{(ii)} \).

ECA-19,-37,-41: For some \( v \), there is no \( \tau \leq 20000 \) to satisfy \( \text{(ii)} \).

ECA-57 satisfies \( \text{(ii)} \), for \( n \geq 5 \) and all \( v \). The largest \( \tau \) required is 28 for \( n = 5 \); 14 for \( n = 6 \); 10 for \( 7 \leq n \leq 13 \); and 9 for \( n = 14 \) and 15.

ECA-105 satisfies \( \text{(ii)} \) for odd \( n \geq 7 \) and all \( v \). The largest \( \tau \) required is 30 for \( n = 7 \); 16 for \( n = 9, 11, 13 \); and 8 for \( n = 15 \).

In general, the time \( \tau \) decreases with \( n \), since the number of patterns, \( 2^n \), increases slower than the number of asynchronicities, \( 3^n - 2^{n+1} + 1 \), and thus for larger \( n \), \( A_S^n \) is more likely to satisfy \( \text{(ii)} \) early on.
We first consider bijective functions on $\mathbb{F}_2^n$. In this case the equivalent group-theoretic statement is:

*Do the $ECA_{AS} \in AS_n$ (written as permutations on the set $\{0, 1, \ldots, 2^n - 1\}$) generate the full symmetric group $S_{2^n}$?*
To answer this question, we used the program GAP (Graphs, Algorithms, Programming) from RWTH Aachen (Prof. Neubüser’s group) and St. Andrews University [5]. Thank you!

We ran GAP on some subsets of only 3 asynchronicity rules to show that $ECA_{AS} \in AS_n$ generates at least the alternating group $A_{2^n}$, for $3 \leq n \leq 11$.

Trying directly to obtain the group generated by the full set $ECA_{AS} \in AS_n$ overburdens GAP already from $n = 4$ on. Therefore, in order to check for the generation of $S_{2^n}$, it is then sufficient to exhibit at least one odd permutation, which is the case for $n = 3$, with the whole $S_{2^3}$ generated — or to show that all permutations generated by $ECA_{AS} \in AS_n$ are even, which is the case for $4 \leq n \leq 11$, and thus only $A_{2^n}$, but not $S_{2^n}$, is generated in these cases.

Out of the 6 ECA families satisfying property (i), ECA-57 and ECA-105 are the only ones, which have a locally bijective update rule. Therefore, only these families must be considered. We immediately have that temporal rules avoiding the symbol “≡” are bijective, when the temporal rule is bijective, since different applications of that temporal rule do not interfere with each other. On the other hand, for $n \neq 3$, all temporal rules involving the symbol “≡” lead to non-bijective functions, see next subsection.

The rules excluding ≡ define bijective functions, whenever the ECA itself is (locally) bijective, that is the application of such an temporal rule for a single cell yields bijectivity. Those temporal rules including ≡ define the non-bijective functions. Hence, the only way to generate bijective functions for $n \geq 4$ is by using ECA-57 or ECA-105, and only applying temporal rules from $\{<, >\}$.

ECA-57: GAP tells us that the $2^n - 2$ temporal rules from $\{<, >\} \{<, >\}$ always yield at least the alternating group $A_{2^n}$, which is in fact generated already by 3 of the temporal rules, for $3 \leq n \leq 10$.

The case $S_{2^n}$ vs. $A_{2^n}$ is easiest checked by hand: Is there some odd permutation within the temporal rules? This is only the case for $n = 3$. For $4 \leq n \leq 10$, all temporal rules yield even permutations and thus can not generate the full $S_{2^n}$.

Hence, for $n = 3$, all bijective functions are generated through ECA-57 by concatenation of suitable temporal rules, while for $n \geq 4$, only the even permutations from $A_{2^n}$ (that is half of the $2^n!$ bijective functions) are generated.

ECA-105: GAP tells us that all bijective temporal rules combined generate only fairly small groups:

- $<AS_3>$ has order 24,
- $<AS_4>$ has cardinality 48,
- $<AS_5>$ has cardinality 1920,
- $<AS_6>$ has cardinality 11520,
- $<AS_7>$ has cardinality 322560,

all are far below $|S_{2^n}| = 2^n!$, the number of bijective functions.

### 3.2 Non-Bijective Functions

We now turn to non-bijective functions. Then $Im(f) \subset \mathbb{F}_2^n$ with $|Im(f)|$ strictly less than $2^n$.

We start with $n = 3$. The convex hull over all $AS \in AS_3$ has cardinality at least $2^3/2 = 20160$ for the following ECA’s, Table 3 (the other ECA with bijective update rule, ECA-105, generates only 344 functions):

| ECA  | Image Size |
|------|------------|
| ECA-25 | 22496 |
| ECA-110 | 23166 |
| ECA-30 | 25258 |
| ECA-3 | 39155 |
| ECA-57 | 40320 |
| ECA-11 | 52934 |
| ECA-62 | 62683 |

Table 3: Image size for ECAs on $\mathbb{F}_2^3$
There are \(8^8\), about 16 Mio., functions on \(\mathbb{F}_3^2\). Hence, for \(n = 3\), none of the ECA’s even generates a quarter of all functions. The case ECA-57 is special in that this ECA actually generates all bijective functions, but no non-bijective one, for \(n = 3\).

In the sequel, \(n \geq 4\), we consider only ECA-57, which has sufficiently many bijective functions, namely \(2^{n!}/2\), at least for \(4 \leq n \leq 10\). We will generate a considerable subset of all functions by suitably interleaving bijective and non-bijective temporal rules for ECA-57.

We now consider ECA-57 for a temporal rule with a single \(\equiv\) on \(\mathbb{Z}/n\mathbb{Z}, n \geq 4\).

Considering larger neighborhoods, with 2 cells changing simultaneously, also ECA-57 becomes non-surjective (we show the effect of \(\text{AS}_1 = \equiv\) on the two middle cells for all configurations of 4 adjacent cells):

| \(v\)    | \(\text{ECA-57}(v)\) |
|---------|---------------------|
| 0000    | 0110                |
| 0001    | 0101                |
| 1000    | 1110                |
| 1001    | 1101                |
| 0010    | 0000                |
| 0011    | 0011                |
| 1010    | 1100                |
| 1011    | 1111                |
| 0100    | 0010                |
| 0101    | 0011                |
| 1100    | 1010                |
| 1101    | 1011                |
| 0110    | 0100                |
| 0111    | 0101                |
| 1110    | 1000                |
| 1111    | 1001                |

Table 4: ECA-57: Effect of \(\text{AS}_1 = \equiv\)

We obtain the patterns 0011 and 0101 twice, while missing 0001 and 0111. Hence the image is smaller than the full \(2^4\) by 2, or by a factor of \(7/8\).

Extending this neighborhood of \(\equiv\) to any size \(n\), and using only \(<\) and \(>\) for the other \(n - 1\) positions, before and after the \(\equiv\) transition, ECA-57 behaves bijectively. Therefore, the whole image shrinks by just the factor \(7/8\), when applying \(\equiv\) once.

Since all temporal rules without \(\equiv\) are bijective, and inclusion of more than one \(\equiv\) shrinks the image even further, we have the following result on the functions that can be represented by ECA-57:

**Theorem 3**

*Let the patterns from \(\{<, >\}^n \setminus \{<^n, >^n\}\) generate at least the alternating group \(A_{2^n}\) (which is the case at least for \(3 \leq n \leq 10\)).

Let \(f : \mathbb{F}_2^n \to \mathbb{F}_2^n, n \geq 4\) be any non-bijective function on at least 4 symbols. Let \(\#(w) = |\{v | f(v) = w\}|\) be the number of configurations \(v\) leading to configuration \(w\). Then \(f\) is representable by ECA-57 under asynchronicity, if and only if

\[
\sum_{w \in \mathbb{F}_2^n} \frac{\#(w)}{2} \geq 2^{n-3}.
\]
We first introduce the functions \( \# \) on \( \mathbb{F}_2^n \) and \( @ \) on \( \mathbb{N}_0 \):

The multiplicity \( \#(w) \) tells us, how often \( w \) is reached, i.e. is the size of the preimage of \( \{ w \} \).

For \( k \in \mathbb{N}_0 \), let \( @(k) \in \mathbb{N}_0 \) be the number of results \( w \) appearing with multiplicity \( k \), \( @(k) = |\{ w \in \mathbb{F}_2^n : \#(w) = k \}| \). In particular, \( @(0) = 2^n - |\text{Im}(f)| \) is the number of words avoided by the image of \( f \).

We have \( \sum_k k \cdot @(k) = 2^n \).

We make use of the temporal rule \( AS^* := \{<\cdots>\cdots>\} \) which maps \( 2^{n-3} \) pairs \( (v_1,v_2) \) onto \( 2^{n-3} \) words \( w \), and otherwise is 1-to-1. Hence, for \( AS^* \), we have \( @(1) = 6 \cdot 2^{n-3} \), \( @(0) = @(2) = 2^{n-3} \). We generate \( f \) by a chain \( f = \pi_k \circ AS^* \circ \cdots \circ \pi_2 \circ AS^* \circ \pi_1 \circ AS^* \), alternating \( AS^* \) and permutations \( \pi_k \in S_{2^n} \).

The second and every further application of \( AS^* \), we will join \( 2^{n-3} - 1 \) words \( v_1 \) with \( \#(v_1) > 0 \) to \( 2^{n-3} - 1 \) words \( v_2 \) with \( \#(v_2) = 0 \), hence without changing the distribution \( @ \). We also map \( 6 \cdot 2^{n-3} \) words 1-to-1, and finally we join two multiplicities \( \#(v_1), \#(v_2) \) by mapping \( v_1,v_2 \) onto the same \( w \), the actual effect of this application of \( AS^* \). The new values \( @^+ \) are thus \( @^+ (\#(v_1)) = @(\#(v_1)) - 1, @^+ (\#(v_2)) = @ (\#(v_2)) - 1, @^+ (\#(v_1) + \#(v_2)) = @ (\#(v_1) + \#(v_2)) + 1 \), and \( @^+(k) = @(k) \) otherwise. In this way, we eventually arrive at a distribution \( @ \) as required by \( f \).

To achieve this, we permute values in between applications of \( AS^* \). In this (slow) way, we eventually get to the distribution of \( \#(w) \) required by \( f \).

The final permutation \( \pi_k \) maps the \( v \) with multiplicities \( \#(v) > 0 \) to the correct values \( w \in \text{Im}(f) \).

Since we always have two words \( v_1,v_2 \) mapping to the same \( w \) under \( AS^* \), and also two words \( w_1,w_2 \) outside \( \text{Im}(AS^*) \), any \( \pi_k \in S_{2^n} \setminus A_{2^n} \) can be extended by one of the transpositions \( (v_1,v_2) \) or \( (w_1,w_2) \) to an equivalent \( \pi_k \in A_{2^n} \).

Concerning the “only if” part, already the first application of \( AS^* \) would decrease the number of values below \( |\text{Im}(f)| \).

3.3 Examples

The superscript \( (n) \) indicates the torus size.

**INC**\(^{(3)}\) For \( n = 3 \), let \( w = v + 1 \mod 8 \). This is an odd bijective function, and hence representable for this \( n = 3 \).

**MUL-BY-3**\(^{(3)}\) For \( n = 3 \), let \( w = 3 \cdot v \mod 8 \). Same as with INC.

**MUL-BY-2**\(^{(3)}\) For \( n = 3 \), let \( w = 2 \cdot v \mod 8 \). From \( 2 \cdot 0 = 2 \cdot 4 = 0 \mod 8 \), this function is not bijective, and hence not representable by ECA-57 for \( n = 3 \).

**INC**\(^{(4)}\) For \( n = 4 \), let \( w = v + 1 \mod 16 \). As with \( n = 3 \), this is an odd bijective function. Contrary to the case \( n = 3 \), a representation by ECA-57 is not possible for \( n \geq 4 \).

**INC**\(^{(4)}\) For \( n = 4 \), let \( w = v + 1 \mod 15,15 \leftrightarrow 15 \). This is an even bijective function, and thus representable.

**MUL-2-BY-2**\(^{(4)}\) For \( n = 4 \), let \( v = a|b, 0 \leq a,b \leq 3 \) and \( w = a \cdot b \). The range is given by the multiset \( \{0^7,1,2^2,3^2,4,6^2,9\} \), where superscripts show the number of occurrences. The sum \( \sum_{w \in \mathbb{F}_2^3} [\#(w)/2] \geq 6 \geq 2^{n-3} = 2 \) is large enough (the range is sufficiently small thus) to allow shrinking by e.g. repeated application of the asynchronicity pattern “\(<=>>\)” and suitable permutations. Multiplication can thus be computed by ECA-57 through asynchronicity. How to do it exactly, is a more complicated case, see Open Problems.

**MUL-k-BY-k**\(^{(2k)}\) Without appearing \( 2 \cdot 2^k \mod 1 \) times, and \( 1 \leq a < b \leq 2 \) with a \( b \) yields \( a \cdot b = b \cdot a \) that is at least \( \binom{2^k}{2} \) pairs. Hence, we have \( \sum_{w \in \mathbb{F}_2^3} [\#(w)/2] \geq 2^k - 1 + (2^k - 1) \cdot (2^k - 2)/2 > 2^{2k-3} = 2^{n-3} \) (with \( k \geq 2 \)). All these multiplications can therefore be computed by ECA-57, using asynchronicity.
Boolean and arithmetic functions on $k$ bits, $n = 2k$:

Let $v = a/b$ with $0 \leq a, b < 2k$. Then $f_{V,1}(v) = (0 \lor a \land b)$, $f_{V,2}(v) = (a \lor b \land a \lor b)$, $f_{\land,1}(v) = (0 \lor a \land b)$, $f_{\land,2}(v) = (a \land b \land a \land b)$, $f_{\lor,1}(v) = (0 \lor a \land b)$, $f_{\lor,2}(v) = (a \lor b \land a \lor b)$, $f_{-,1}(v) = (a - b \mod 2^k)$, $f_{-,2}(v) = (0)(a-b \mod 2^k)$, can all be computed by ECA-57 under asynchronicity, for any $k$ that is any even $n$:

All these Boolean functions are commutative, $a \circ b = b \circ a$, and thus enough pairs $(v_1, v_2)$ with $f(v_1) = f(v_2)$ exist to have $\sum |\#(w)/2| \geq 2^{2k-3}$.

For $0 \leq v < 2^n$, let $f_{\text{NEG}}(v) = (-v)$ (2’s complement). This is an odd bijection with the two fixed points 0 and $2^n-1$, and $2^{n-1} - 1$ transpositions, hence computable by ECA-57 only for $n = 3$.

$f_{\text{COMP}}(v) = (v \oplus 111 \cdots 111)$ (1’s complement), on the other hand, is an even bijection, computable for all $n$.

**Example 5** A detailed description of the calculation of $\text{INC}^{(3)}$ and $\text{MUL-BY-3}^{(3)}$. The left column indicates the temporal rule and partition of $\mathbb{Z}/3\mathbb{Z}$.

| INC$(^3)$ | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|-----------|-----|-----|-----|-----|-----|-----|-----|-----|
| $<><(1|0|2)$ | 011 | 101 | 100 | 110 | 111 | 010 | 001 | 000 |
| $<><(1|2|0)$ | 010 | 110 | 011 | 000 | 001 | 101 | 100 | 111 |
| $<><(1|0|2)$ | 101 | 000 | 110 | 011 | 100 | 010 | 111 | 001 |
| $<><(0|1|2)$ | 010 | 111 | 100 | 110 | 011 | 001 | 000 | 101 |
| $<><(0|1|2)$ | 001 | 010 | 011 | 100 | 101 | 110 | 111 | 000 |

| MUL-BY-3$(^3)$ | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|
| $<><(2|1|0)$ | 101 | 010 | 111 | 100 | 110 | 011 | 001 | 000 |
| $<><(2|0|1)$ | 001 | 101 | 100 | 010 | 011 | 000 | 110 | 111 |
| $<><(2|0|1)$ | 110 | 011 | 010 | 111 | 000 | 101 | 001 | 100 |
| $<><(0|1|2)$ | 101 | 100 | 001 | 010 | 110 | 000 | 111 | 011 |
| $<><(0|1|2)$ | 000 | 011 | 110 | 001 | 100 | 111 | 010 | 101 |

Table 5: INC$(^3)$ and MUL-BY-3$(^3)$ in detail

**Further Research and Open Problems**

1. Give an algorithm to calculate the temporal sequence $(\text{AS}_1, \text{AS}_2, \ldots, \text{AS}_k)$ for a function on $\mathbb{F}_2^n$ directly from the function values, given e.g. as permutation on $\{0,1,2^{n-1}\}$, instead of searching through the full tree $\text{AS}_n^k$.

2. Consider temporal sequences that do not depend on the position, but on the rule to be applied, e.g. first update at all corresponding sites $000 \rightarrow p_0$, then $110 \rightarrow p_6$, then $010 \rightarrow p_2$ etc. There are $8! = 40320$ such temporal rules, independent of $n$.

   How do we treat actions that already had their turn, but whose neighborhood only turns up later? Update immediately upon creation, never in this round, ...?

   This could mimic chemical reactions, e.g. in cell biology, DNA expression, where some reactions are faster than others, depending on their reaction rate constant $k$.

3. As in 2., but associate a latency time with each temporal rule: As soon as the corresponding neighborhood pattern is created, wait for its latency time and then update according to the temporal rule.
4. What can we say about the alphabet \{0, 1, 2\} instead of \{0, 1\}?

There are now \(3^3 \approx 2^{43}\) ECA’s to be considered. Since there are \(3^n - 2^{n+1} + 2 = \Theta(3^n)\) asynchronicities (Definition 1) and exactly \(3^n\) configurations on \(\mathbb{Z}/n\mathbb{Z}\), an analogue of properties (o) to (iv) is now impossible due to lack of temporal rules. However, we may ask, how the number of configurations actually reached grows with \(n\). Do we ever obtain the full diversity of \(3^n - 2^{n+1} + 2\) results?

**Conclusion**

We have introduced temporal order via temporal rules as a means to diversify the behaviour of elementary cellular automata.

In particular, ECA’s with update rules 19, 41, and 57 are pattern universal for \(n \leq 12\), achieving any desired pattern transduction \(v \mapsto w\), applying iteratedly a single temporal rule. We conjecture that they are indeed uniformly pattern-universal.

ECA-57 produces any even (as permutation) bijective function on \(\{0, 1\}^n\), for \(n \leq 10\), and all non-bijective ones that join at least \(2^{n-3}\) pairs of argument values.

**Temporal order is thus a third way to encode information and algorithms, after programs (ECA’s) and data (initial configurations).**

This may have farreaching consequences, e.g. for modeling gene expression, since physico-biological processes seldomly achieve exact synchronicity.

**References**

[1] N. Fatès, E. Thierry, M. Morvan, N. Schabanel, *Fully asynchronous behavior of double-quiescent elementary cellular automata*, TCS 362, 1–16, 2006. doi:10.1016/j.tcs.2006.05.036

[2] T. Ingerson, R. Buvel, *Structure in asynchronous cellular automata*, Physica D, 10, p. 59, 1984. doi:10.1016/0167-2789(84)90249-5

[3] J. Lee, F. Peper, S. Adachi, K. Morita, S. Mashiko, *Reversible computation in asynchronous cellular automata*, Proc. Unconventional Models of Computation, LNCS 2509, 220–229, 2002. doi:10.1007/3-540-45833-6_19

[4] A. Sharkar, *Asynchronous Cellular Automata*, Thesis, Shibpur, West Bengal, India, 2011.

[5] www.gap-system.org
Appendix A – ECA Families

Each family (equivalence class under the symmetries 0/1 and L/R (c_i−1 ↔ c_i+1)) consists in up to 4 ECAs with numbers ECA=abcdefgh,

| ECA^{L/R} = aecebgfdh_{2} | ECA^{0/1} = hgfedcba_{2} | ECA^{L/R,0/1} = hdjbgceaa_{2} |
|----------------------------|-----------------------------|-------------------------------|
| 0 (255), 4 (223), 8 (239 64 253), 12 (207 68 221), 18 (183), 24 (231 66 189), 28 (199 70 157), 33 (123), 37 (91), 42 (171 112 241), 46 (139 116 209), 56 (227 98 185), 62 (131 118 145), 76 (205 76 205), 94 (133), 108 (201), 128 (254), 136 (238 192 252), 146 (182), 156 (198), 168 (234 224 248), 184 (226), | 1 (127), 5 (95), 9 (111 65 125), 13 (79 69 93), 19 (55), 25 (103 67 61), 29 (71), 34 (187 48 243), 38 (155 52 211), 43 (113), 50 (179), 57 (99), 72 (237), 77, 104 (233), 110 (137 124 193), 130 (190 144 246), 138 (174 208 244), 150, 160 (250), 170 (240), 200 (236), | 2 (191 16 247), 6 (159 20 215), 10 (175 80 245), 14 (143 84 213), 19 (55), 22 (151), 26 (167 82 181), 30 (135 86 149), 34 (187 48 243), 40 (235 96 249), 44 (203 100 217), 50 (179), 57 (99), 72 (237), 77, 104 (233), 110 (137 124 193), 130 (190 144 246), 138 (174 208 244), 150, 160 (250), 170 (240), 200 (236), | 3 (63 17 119), 7 (31 21 87), 11 (47 81 117), 15 (85), 23, 27 (39 83 53), 32 (251), 36 (219), 41 (107 97 121), 45 (75 101 89), 54 (147), 60 (195 102 153), 74 (173 88 229), 90 (165 90 165), 106 (169 120 225), 126 (129), 134 (158 148 214), 142 (212), 154 (166 210 180), 164 (218), 178, 232 |