Existence of non trivial synchronizing cellular automaton on periodic configurations

Gaétan Richard

Greyc, CNRS & Université Caen Basse Normandie

Frac d'automne 2013
LIFO, Orléans

Table of contents

1. Synchronizing CA

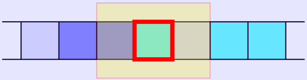
Cellular automata

A cellular automaton (CA) is a pair (Q, f) where:

- Q is the finite set of states;
- $f : Q^3 \to Q$ is the local transition function.

Configurations are elements of $Q^{\mathbb{Z}}$.

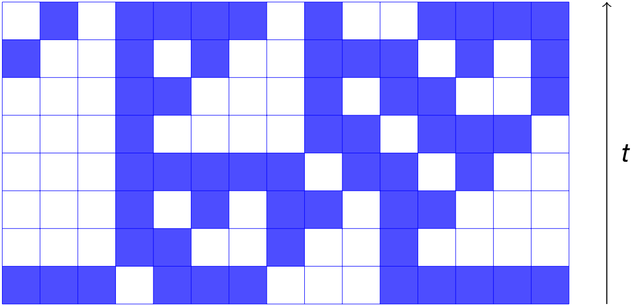
A periodic configuration of period $p \in \mathbb{N}^*$ is a configuration $c \in Q^{\mathbb{Z}}$ such that for any $i \in \mathbb{Z}$ $c_{i+p} = c_i$



One example: the "xor" automaton

Local rule: Make the xor with the left neighbourhood.
Global transition function: We denote as $F : Q^{\mathbb{Z}} \to Q^{\mathbb{Z}}$ the global transition function.

Space-time diagram:



Synchronization

Definition:

A cellular automata is synchronizing if there exists a state q such that, for any configuration $c \in Q^{\mathbb{Z}}$, there exists a constant N such that

$$F^N(c) = ^\omega q^\omega$$

Some thoughts

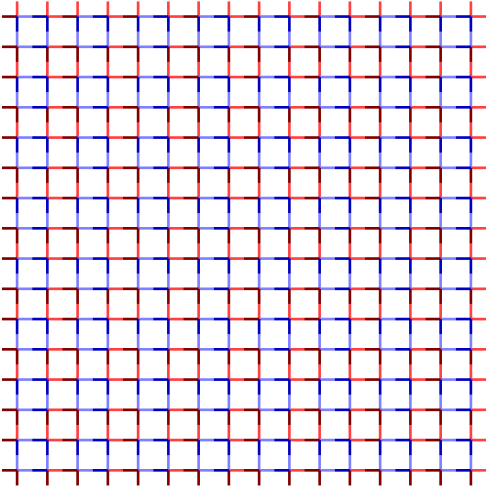
Examples: Nilpotent CAs are (trivial) synchronizing CA

Remarks: Due to the existence of universe configuration, any synchronizing CA (on all configurations) is nilpotent.

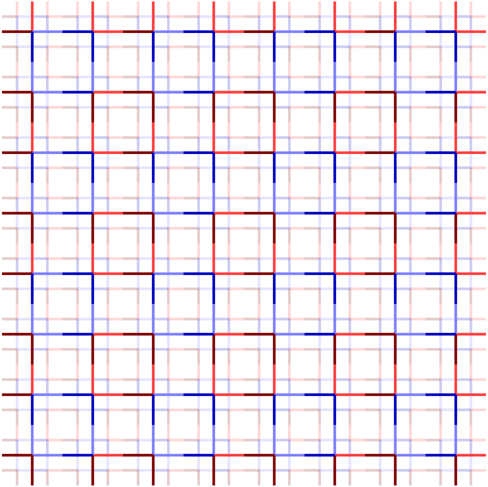
Question: Does there exist synchronizing CAs on periodic configurations which are not nilpotent ?

2. Kari’s wonderful tiling

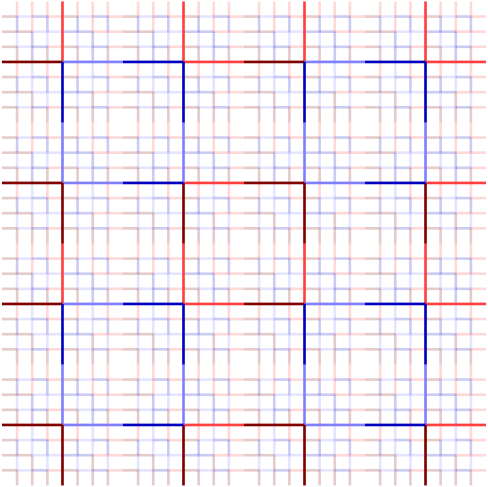
Robinson



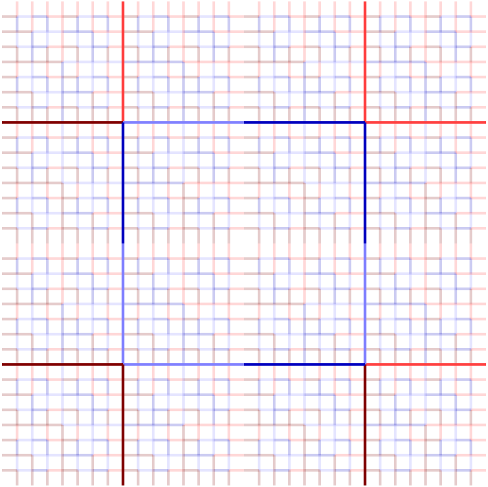
Robinson



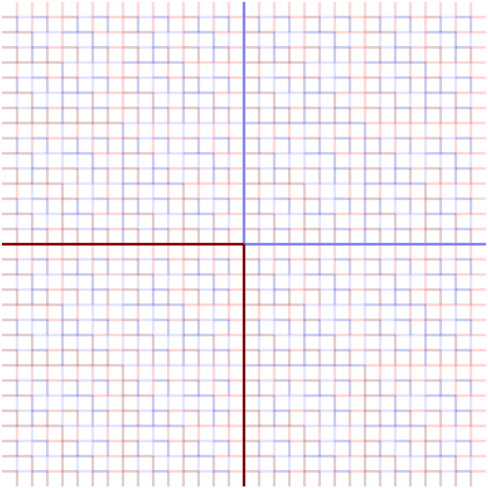
Robinson



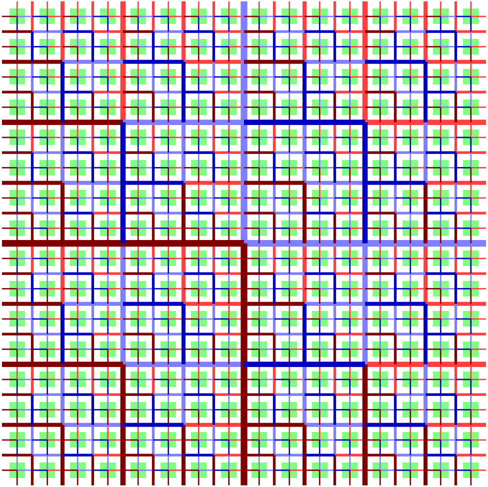
Robinson



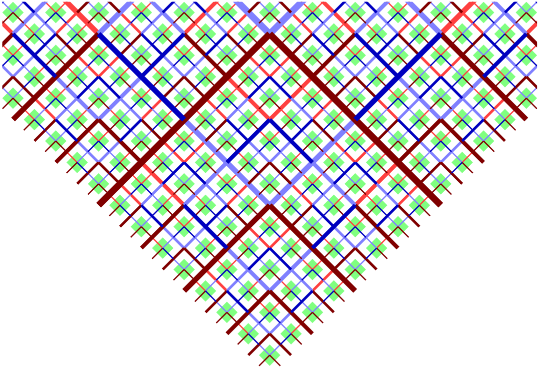
Robinson



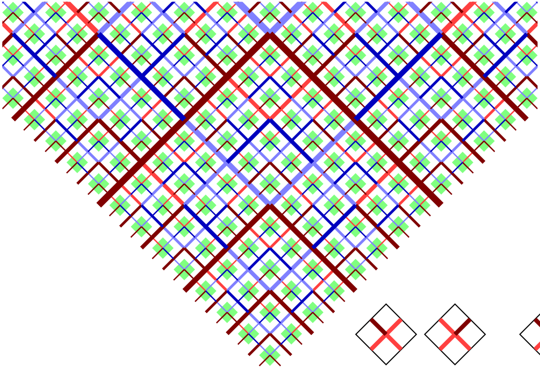
Robinson

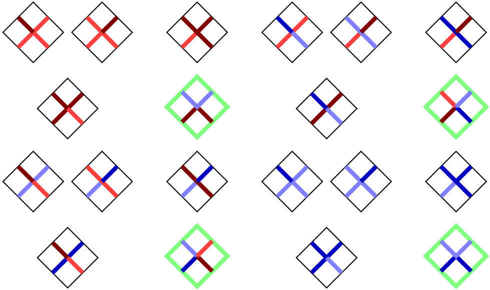


Kabinson

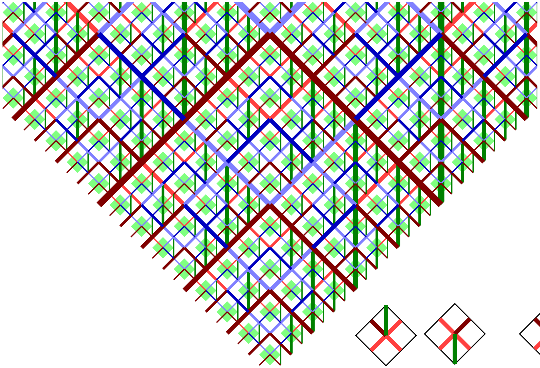


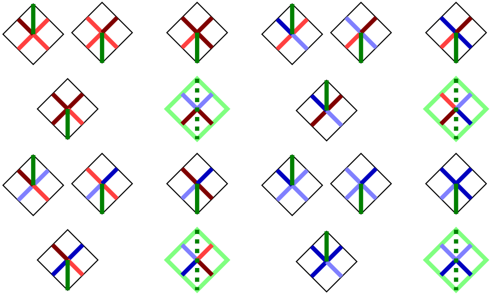
Kabinson





Kari

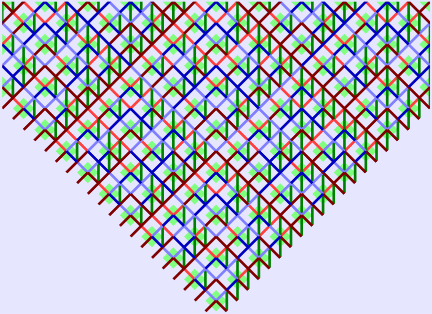




Kari is regular

Regularity:

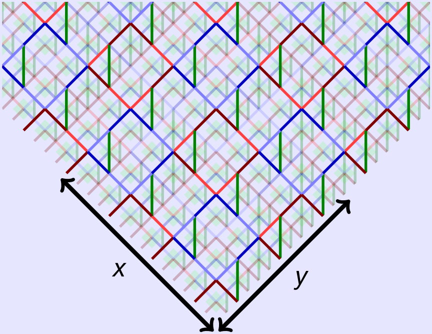
- The central diagonals are regular;



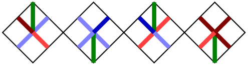
Kari is regular

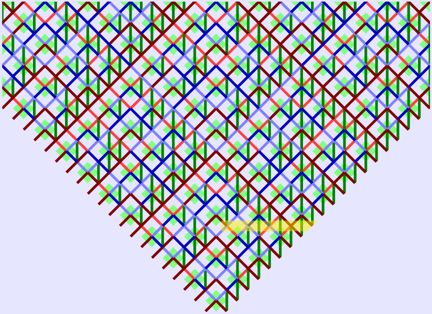
Regularity:

- The central diagonals are regular;
- For a point at position *x*, *y*, the right going signal is determined by the *p*-th and $p + 1$-th bits of *y* where *p* is the rightmost bit of *x* with value 1.

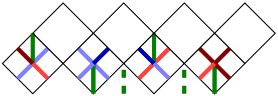


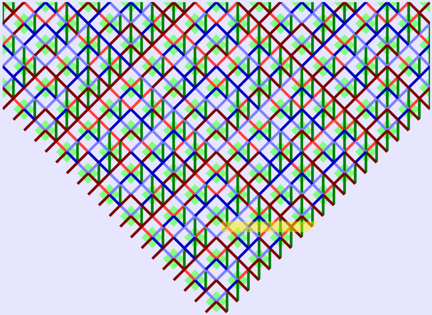
Kari is extensible



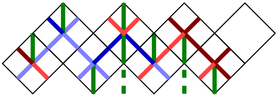


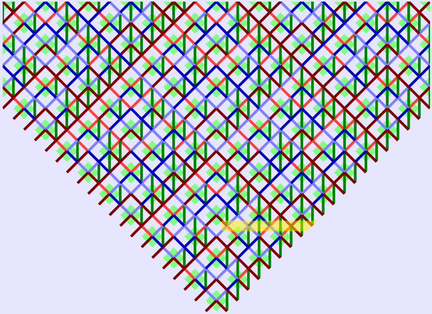
Kari is extensible



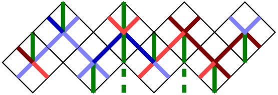


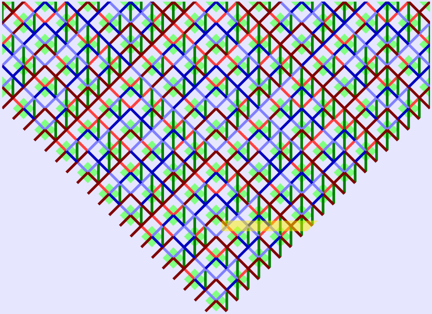
Kari is extensible



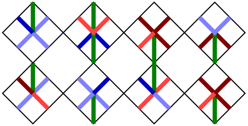


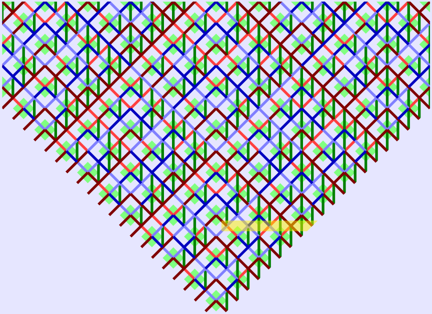
Kari is extensible



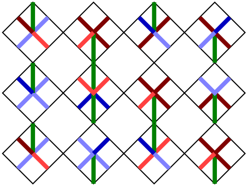


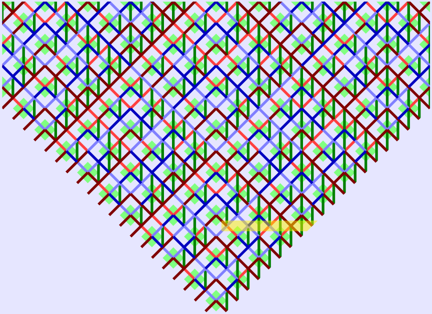
Kari is extensible



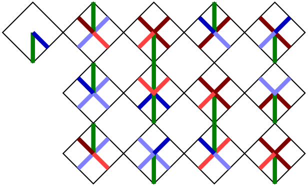


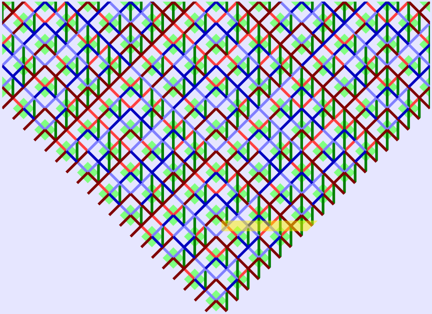
Kari is extensible



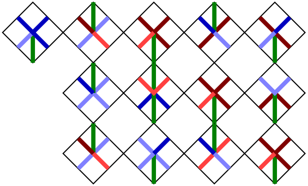


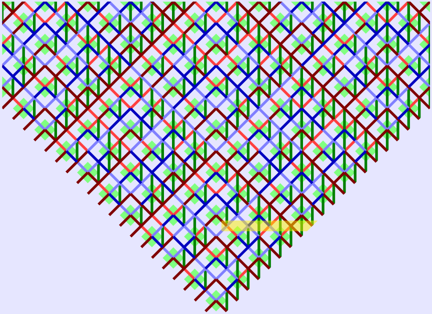
Kari is extensible



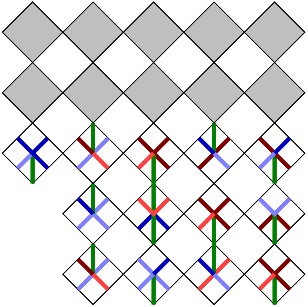


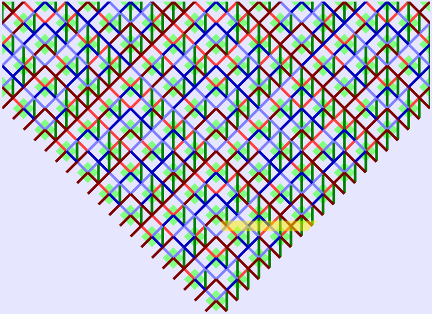
Kari is extensible



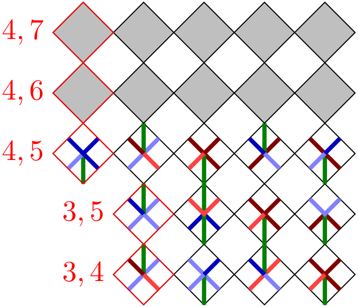


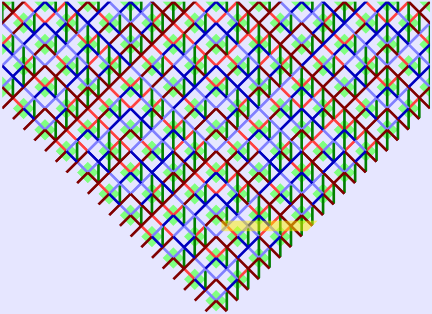
Kari is extensible



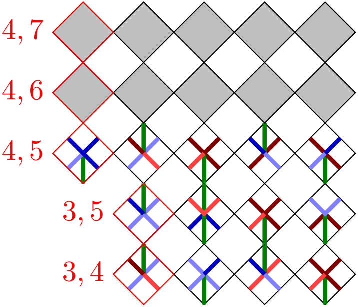


Kari is extensible

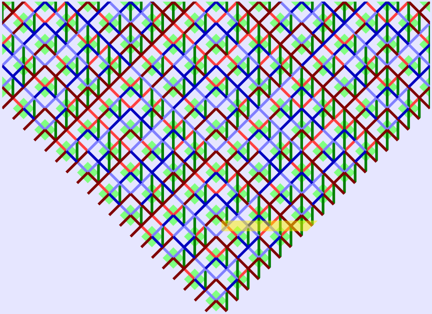




Kari is extensible



Binary: numbers are encoded in binary resulting the need of *log*(n) delays.



3. Interval merging

The automaton

Set of states:

$$S = \{0, 1\} \cup (K \times \{-, \vdash, \dashv, \vdash\!\dashv\} \times W)$$

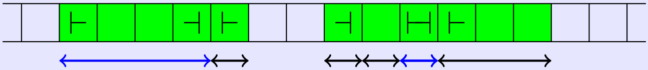
Ideas:

- $f(0, 0, 0) = 1$ and $f(1, 1, 1) = 0$;
- $\vdash$ and $\dashv$ are oriented borders which can progress only on 0 or 1 at speed at most $1/2$;
- except for those borders, 0 and 1 are spreading;
- in case 01 or 10 appears, one create an new empty interval with dual border $\vdash\!\dashv$;
- any error or undefined case enter state 0.

Interval

Definition: An interval is a consecutive sequence of states between $\vdash$ and $\dashv$ not containing any 0 or 1.

Definition: An interval is condemned where is border are not $\vdash$ on the left and $\dashv$ on the right.



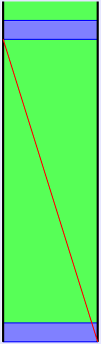
Lemma: Condemned intervals will disappear in linear time.

Timer

TTL: each free interval in endowed with a signal going right to left at speed $1/4$

Once it reaches the left border, the interval is checked in starting form and have a new signal ready on the right.

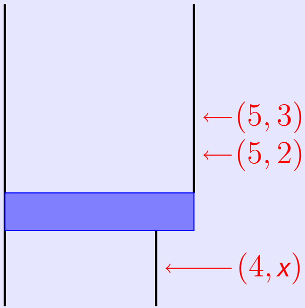
This ensure every free interval is doing correct computation.



Order on interval

Order: Free intervals are sorted by increasing size n then time t since they have reached this size.

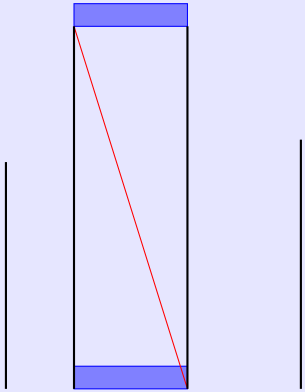
Graded order: by construction, we will ensure later than size can only be on the form (n, t) with $t < 12n$.



Interval behavior: observation

Objective: Compare itself with our neighbours without interacting.

Method:

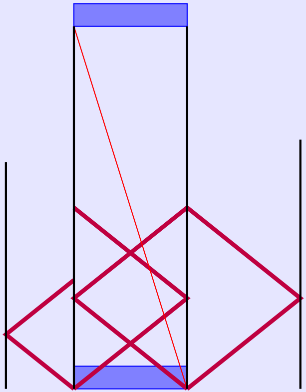


Interval behavior: observation

Objective: Compare itself with our neighbours without interacting.

Method:

- Compare its size with both neighbours;

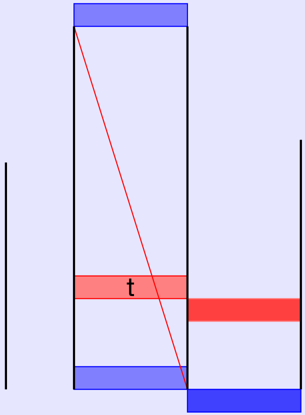


Interval behavior: observation

Objective: Compare itself with our neighbours without interacting.

Method:

- Compare its size with both neighbours;
- launch a first Firing-squad and compare time with similar of neighbours;

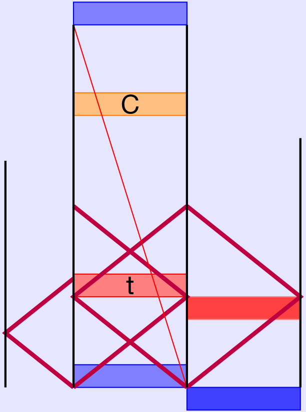


Interval behavior: observation

Objective: Compare itself with our neighbours without interacting.

Method:

- Compare its size with both neighbours;
- launch a first Firing-squad and compare time with similar of neighbours;
- check results.

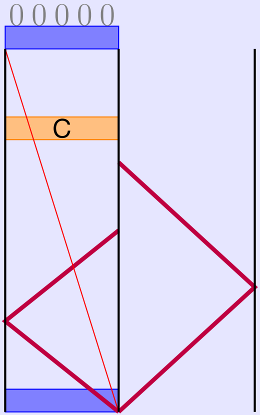


Interval behavior: actions

Time: Choice is made at time **C**.

Possible actions:

- If smaller than one of the interval, disapear;

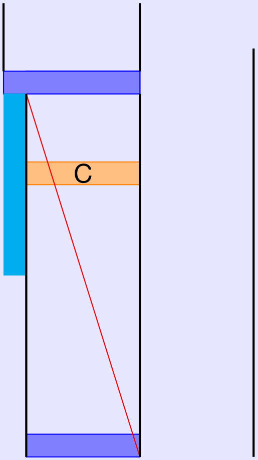


Interval behavior: actions

Time: Choice is made at time **C**.

Possible actions:

- If smaller than one of the interval, disapear;
- If possible, expand;



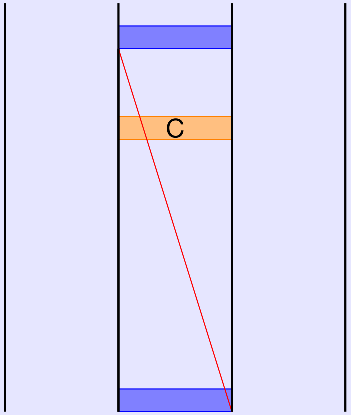
Interval behavior: actions

Time: Choice is made at time **C**.

Possible actions:

- If smaller than one of the interval, disapear;
- If possible, expand;
- Otherwise, repeat (limited to twice a size).

Life limit: One interval is limited to 3 iterations of the behavior *per size*.



Limit set

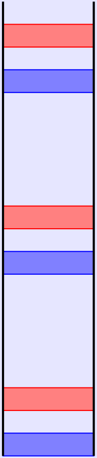
Greatest interval: once there are only artificial intervals, the maximum of interval is strictly increasing unless disappearance occurs.

Key point: In the latter case, only possible choice is go into uniform 0 configuration

Proof by picture

Facts:

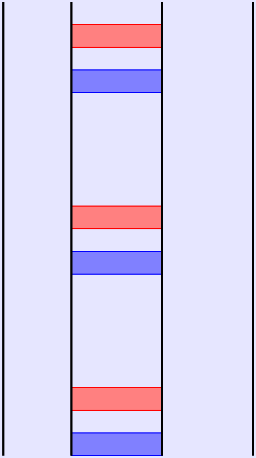
- The maximal interval has spend $3 \times 4n$ at the maximal size being the greatest;



Proof by picture

Facts:

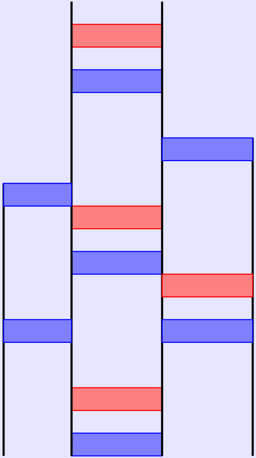
- The maximal interval has spend $3 \times 4n$ at the maximal size being the greatest;
- It had always two neighbours;



Proof by picture

Facts:

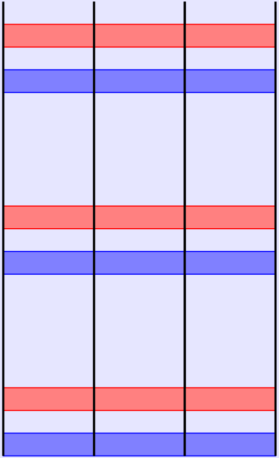
- The maximal interval has spend $3 \times 4n$ at the maximal size being the greatest;
- It had always two neighbours;
- Those neighbours are equal with it;



Proof by picture

Facts:

- The maximal interval has spend $3 \times 4n$ at the maximal size being the greatest;
- It had always two neighbours;
- Those neighbours are equal with it;
- All intervals are identical.



4. Conclusion

Combining results

Proposition: The described CA is synchronizing into a length two uniform cycle for periodic configurations.

Extensions: The algorithm can be extended into any desired length cycle, several cycle, “regular” language, …

Open problems:

- Extension to higher dimension ?
- Use ?

Combining results

Proposition: The described CA is synchronizing into a length two uniform cycle for periodic configurations.

Extensions: The algorithm can be extended into any desired length cycle, several cycle, “regular” language, …

Open problems:

- Extension to higher dimension ?
- Use ? *(see J. Mairesse, I hope)*