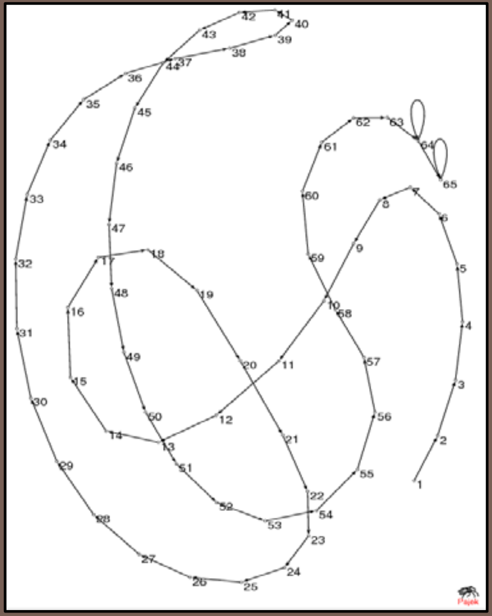
TRAJECTORIES AND ATTRACTORS AS SPECIFICATION FOR THE EVOLUTION OF BEHAVIOUR IN CELLULAR AUTOMATA



2010, 20th July
IEEE CEC 2010
Barcelona

Stefano Nichele and Gunnar Tufte
Norwegian University of Science and Technology – Trondheim

Question

- Is the evolutionary process and the **evolvability** of uniform and non uniform cellular automata influenced by the level of detail in the behavioural description?

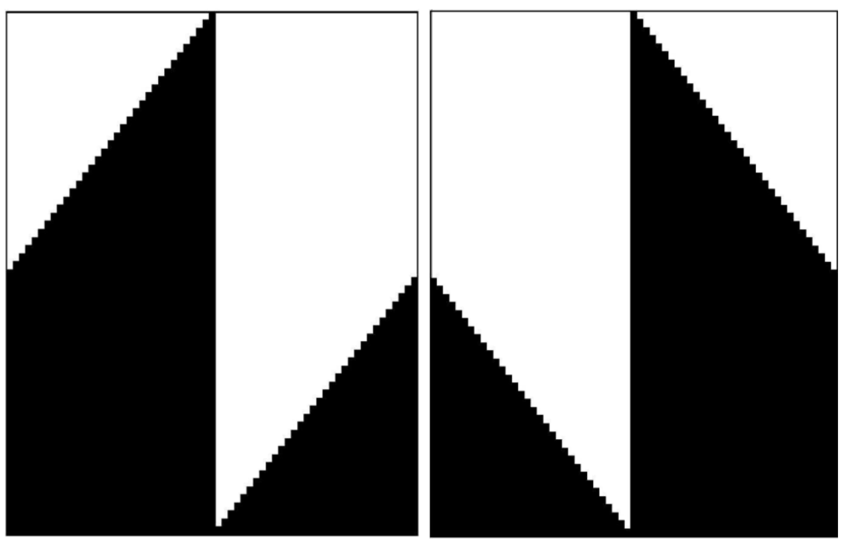
  Experimental approach: the ***behaviour*** *of the machine is specified by including information regarding the* ***trajectory*** *(initial state, final state, intermediate states in specific intervals)*

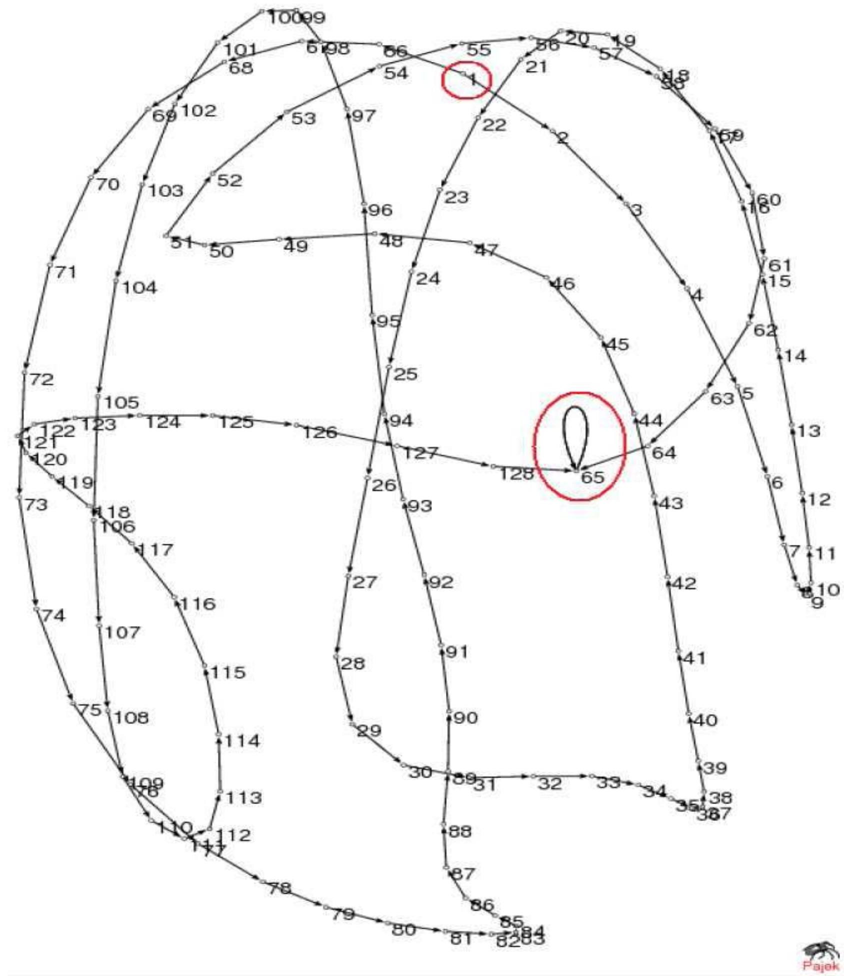
Index

- Introduction on Cellular Automata and behaviour description by trajectories in the basin of attraction
- Trajectories and artificial development
- Description of the applied Evolutionary Algorithm
- Experimental setup and results
  - 1 - Introduction of an intermediate state
  - 2 - Loosen up constrains
  - 3 - Non uniform CA
- Conclusion and future work

Introduction on Cellular Automata and behaviour description by trajectories in the basin of attraction

- Cellular Automata: vast amount of cells, local interconnections, parallel discrete time updates
- CA can be modelled and analysed by methods applied to Boolean Networks and RBN





(a) Behaviour of Rule 238. (b) Behaviour of Rule 252. (c) Trajectory of rule 238 and 252.

Fig. 1. Print out of two CA rules with shared initial state reaching a common attractor by different trajectories.

Introduction on Cellular Automata and behaviour description by trajectories in the basin of attraction

- If an evolutionary approach based on the final state of the CA was applied to find the behaviour of one of these two rules the **solution space** would include both rules (and all other rules sharing the attractor).

  As such, the amount of information must be extended to be able to **discriminate** the sought behaviour from rules with divergent behaviour with the same attractor.

- To be able to discriminate the behaviour of the desired trajectory from other trajectories, **information** of known states or hypothesis of states that should emerge can be included in the specification of behaviour, i.e. the fitness function.

- Evolving behaviour for known CAs is herein used as examples and partly as a test case to investigate how an increased amount of information on the behaviour influences on evolvability. However, the larger goal is to be able to evolve cellular machines that can obtain a desired behaviour based on **incomplete** information. The incompleteness of such behaviour may be a result of trying to model a system where only parts or fragments of the behaviour are known, e.g. a regulatory network for cell cycles.

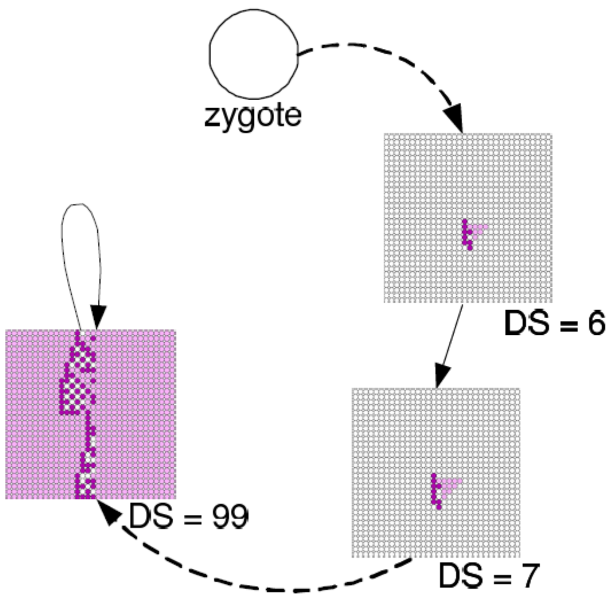
Trajectories and the artificial development

- Moreover, the cellular approach can be related to the process of how **multicellular** systems develops from a single cell

- In biology, development is a process starting with a **zygote** which develops to a multicellular organism.

- An artificial developmental process may also reflect this principle.

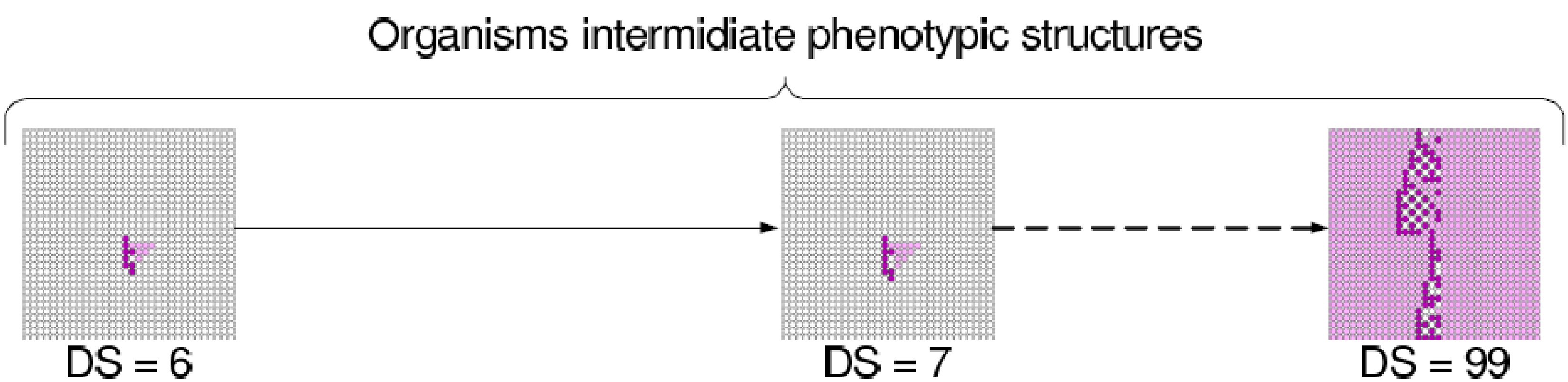
- An initial cell or a given initial condition can be the starting condition, i.e. the zygote, which the iterative developmental process grows from, change and/or reshape to reach a multicellular articial organism.



The developmental path shown as a trajectory.

Trajectories and the artificial development

- If the parallel nature and limited local communication of a cellular developmental process is considered in relation to the discrete time update of the system, a developmental system can be approached as a network of sparsely connected units, i.e. cells.
- Each **phenotype** structure is a node and the transition from node to node in time represents the developmental path from zygote to the fully developed organism.



Description of the applied Evolutionary Algorithm

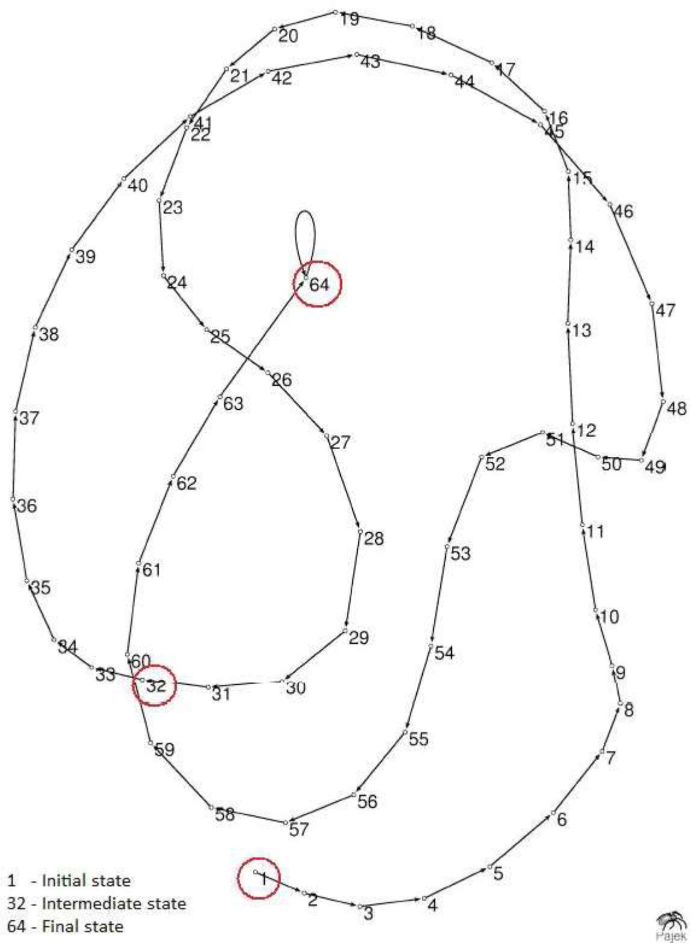
- Genetic Algorithm (GA)
- Fitness Proportionate Selection with crossover (rate 0,7) for uniform CAs and whitout crossover for non-uniform CAs
- Mutation rate 1/L, where L represents the size of the population which is subject to mutation
- Generate initial population of 10 random rules and execute the following steps:
    1. Run the automata for each of the rules
    2. Calculate the best and worst two rules according to the fitness function (hamming distance number of matching bits between the reached final state and the desired final state)
    3. Roulette-Wheel technique to select the rules that will generate the next generation elements
    4. Perform uniform crossover between the two best ranked rules
    5. Replace the two worst ranked rules with the new generated rules
    6. Perform random mutation in the new generated rules with small probability
    7. Repeat all the steps (except the first) until a fitness of 100 per cent is reached

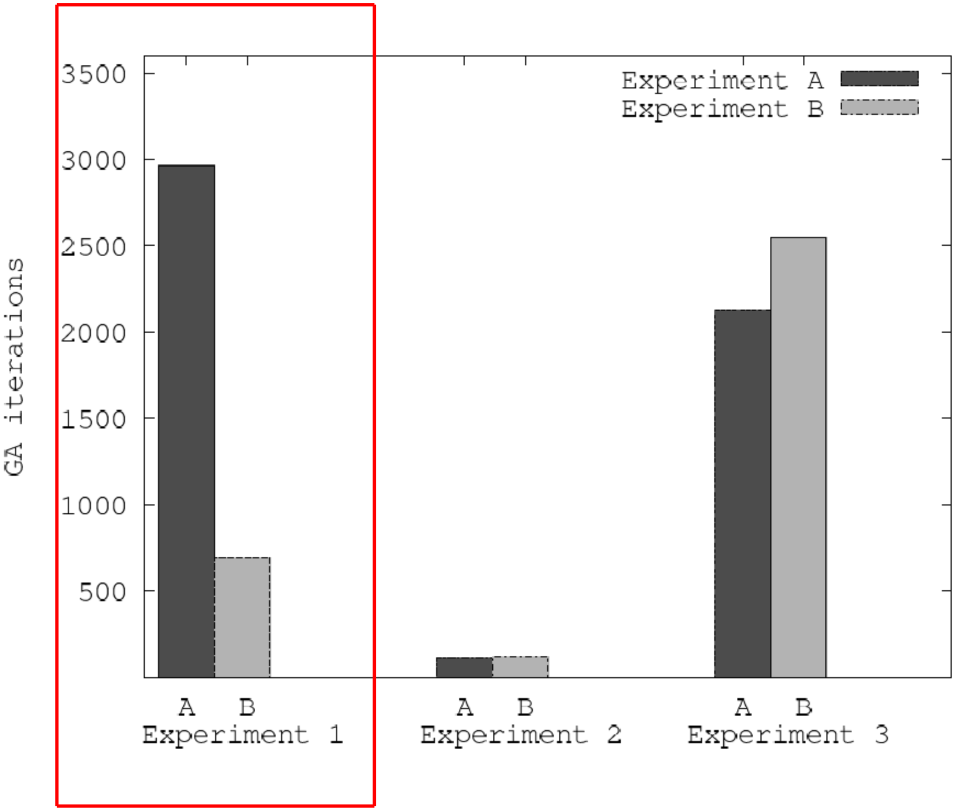
Experimental setup and results
1 - Introduction of an intermediate state

(The experiments show how increased inclusion of trajectory information influence on the evolution of sought behaviours)

- Goal: test how an introduction of a more specific trajectory influenced on the evolvability of the system
- A known trajectory for a given initial condition to a specified end condition was used. The sought behaviour was the trajectory of CA rule 30 (a rule that generates a unique trajectory from the given initial condition of a single cell expressing a logical "1" to its configuration, here defined as the pattern at iteration 64
- Test A: evolving the CA rule that could reach a specific attractor at a specific iteration
- Test B: to test the influence of including more details of the desired trajectory an **intermediate state** was introduced. The intermediate state was the state output of rule 30 at CA iteration 32.

Experimental setup and results
1 - Introduction of an intermediate state



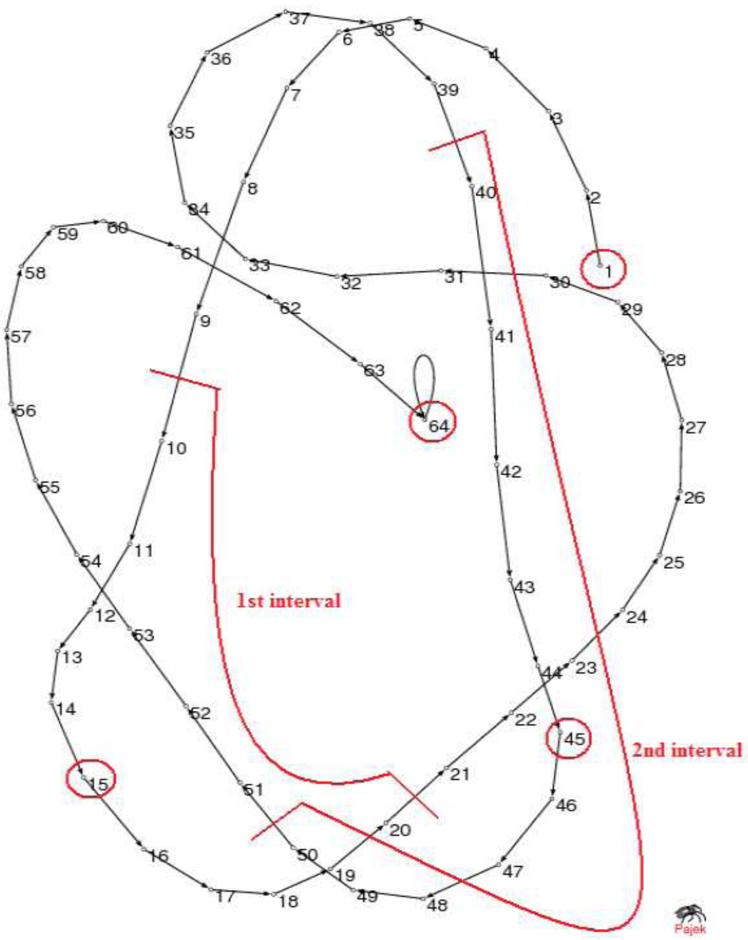


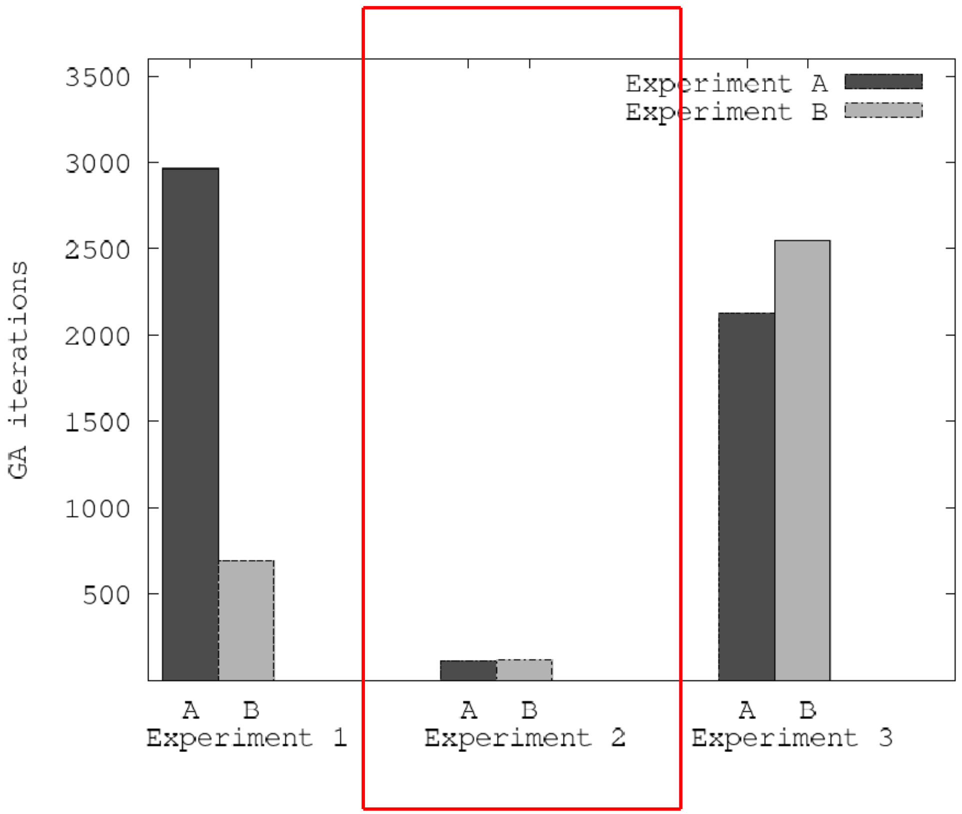
Experimental setup and results: 2 - Loosen up constraints

- Expansion of experiment 1
- Rule 30 is as stated unique. Here we desired a setup where several trajectories could reach the desired final state, i.e. more than one rule
- Rule 206 is a rule with such a characteristic. Rule 252 share the state pattern at the desired CA iteration 64 with rule 206, but follows a different trajectory from the common initial state to the targeted state pattern at CA iteration 64
- Test A: evolving the CA rule that could reach a specific attractor at a specific iteration, crossing an intermediate state in the middle of the trajectory (initial, intermediate and final state using rule 206)
- Test B: we may **not** have the exact timing for the emergence of the desired state, i.e. in regulatory biological networks

  (See following picture for details)

Experimental setup and results:
2 - Loosen up constraints





Experimental setup and results: 3 – Non uniform CA

- Increased state space (each cell may contain a unique rule)
- Set of rules **reduced** to 12: →
- Size of CA reduced to 17
- 12^17 possible rule sets
  (instead of 256^17)

Rule	Name	Function
0	IDENTITY C	$no\ change$
1	IDENTITY L	$\leftarrow LeftPropagation$
2	IDENTITY R	$\rightarrow RightPropagation$
3	OR LC	$OR\ L + C$
4	OR CR	$OR\ C + R$
5	OR LR	$OR\ L + R$
6	XOR LC	$XOR_a\ L \oplus C$
7	XOR CR	$XOR_a\ C \oplus R$
8	XOR LR	$XOR_a\ L \oplus R$
9	NAND LC	$NAND\ \overline{L \bullet C}$
10	NAND CR	$NAND\ \overline{C \bullet R}$
11	NAND LR	$NAND\ \overline{L \bullet R}$

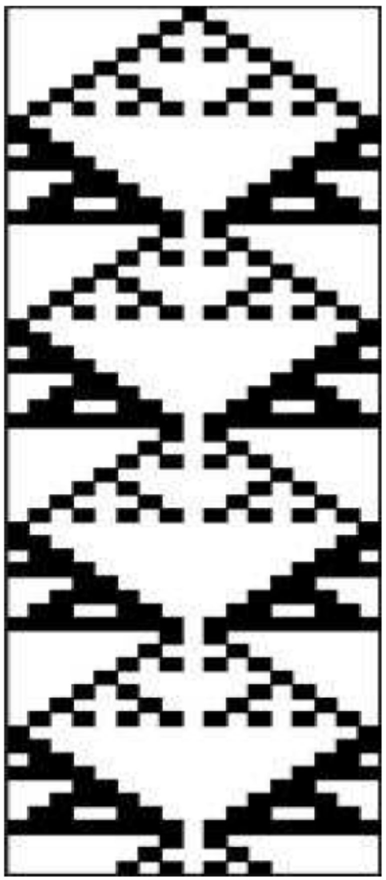
- As the behaviour of the non-uniform CA is not well known, it was decided to use the trajectory of rule 90 in a uniform CA as a desired trajectory

- Test A: 1 intermediate state at CA iteration 32

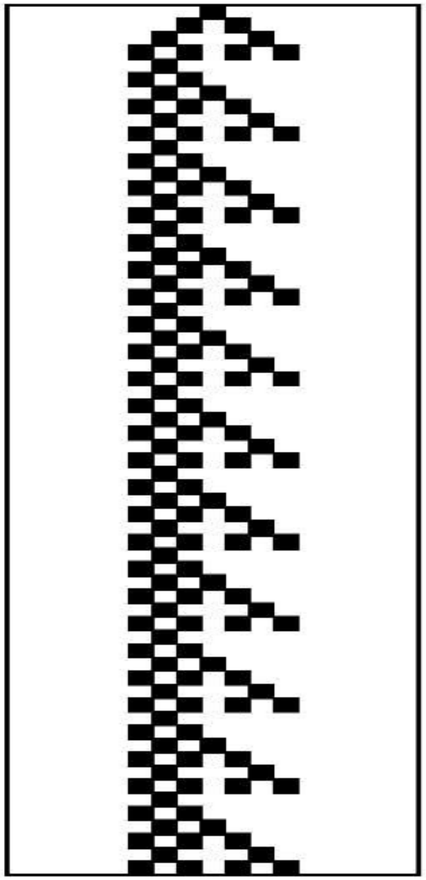
- Test B: 2 intermediate states at CA iteration 15 and 45

M. Sipper, Evolution of Parallel Cellular Machines. The Cellular Programming Approach. Springer-Verlag, 1997.

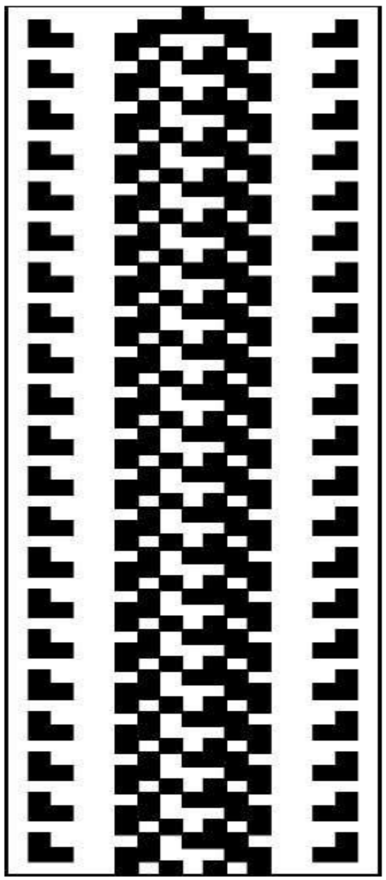
Experimental setup and results: 3 – Non uniform CA



(a) Behaviour of rule 90 over 64 CA iteration.

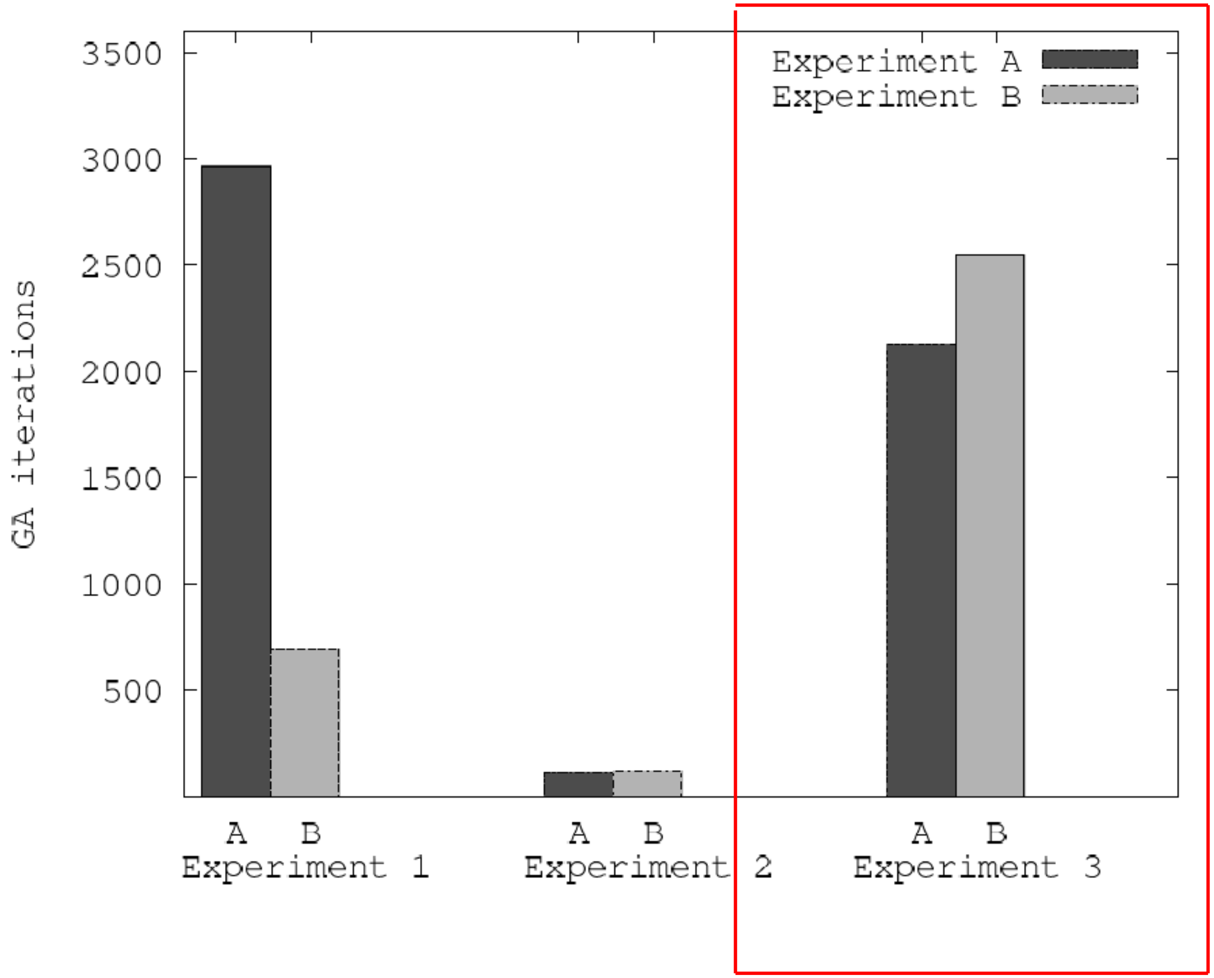


(b) Resulting behaviour of evolved non-uniform CA with one intermediate state from rule 90.



(c) Resulting behaviour of evolved non-uniform CA with two intermediate states from rule 90.

Experimental setup and results:
3 – Non uniform CA



Conclusion & future work

- Exp. 1: extending the information of the sought trajectory actually seems to **increase** evolvability
- Exp. 2: loosen up constrains hardly influenced on the result. However, the introduction of states in a specific interval was **not annihilating** considering evolvability
- Exp. 3: with non-uniform CA, the increased information in the specification slightly increase the required GA iterations but the target behaviour is **still evolvable**
- In general: results are promising for further work. Results show that an interval specification seems to be evolvable. This fits well with the problems we want to target, e.g. **incomplete** information of the trajectory

- Future work:
  - Analysis of 2-dimensional cellular automata
  - Scaling the problem, investigating CAs of bigger size

THANK YOU

All trajectory plots presented were made by PAJEK
V. Batagelj and A. Mrvar, Pajek program for large network analysis. 1991