Quantum Cellular Automata

Presentation for
G53NSC Non-Standard Computation

Jiří Kremser (jxk19u)
Ondřej Božek (oxb09u)

Outline

- Classical Cellular Automata
- Wolfram classes
- Conway's Game of Life
- Reversible Cellular Automata
- Quantum Cellular Automata
- QCA – History
- LQCA
- Partitioned QCA
- Quantum-Dot CA

Classical Cellular Automata

a) Intuitively

- n-dimensional grid of cells
- each cell has neighbourhood
- each cell is in one state
- local rules
- configuration = state of all cells
- example 1-d CA

Classical Cellular Automata

Example of 1-d CA

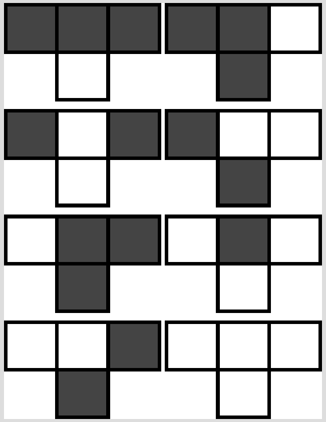
Iteration 1

↓

?

Neighbourhood:
1 cell on the left and right side

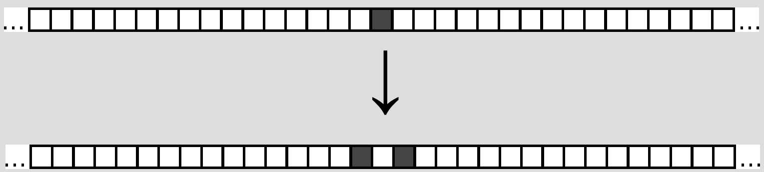
Rules:



Classical Cellular Automata

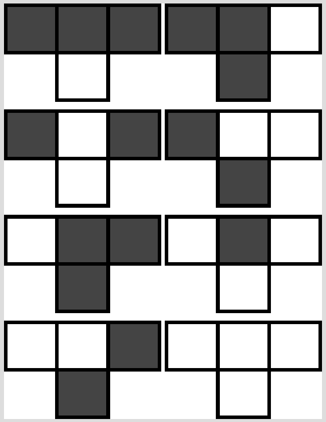
Example of 1-d CA

Iteration 1



Neighbourhood:
1 cell on the left and right side

Rules:



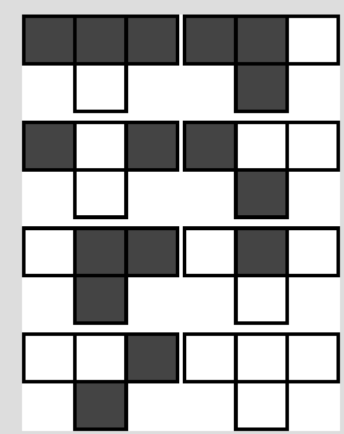
Classical Cellular Automata

Example of 1-d CA

Iteration 2

Neighbourhood:
1 cell on the left and right side

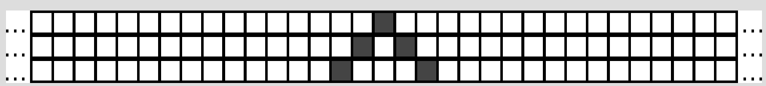
Rules:



Classical Cellular Automata

Example of 1-d CA

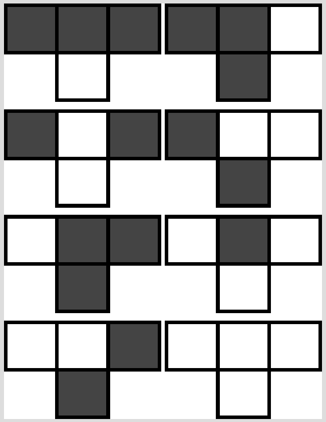
Iteration 3



Neighbourhood:

1 cell on the left and right side

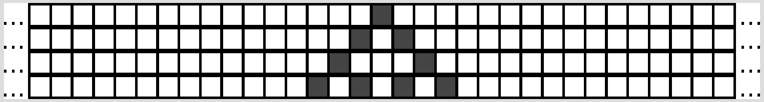
Rules:



Classical Cellular Automata

Example of 1-d CA

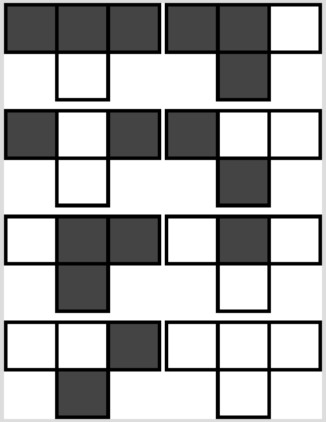
Iteration 4



Neighbourhood:

1 cell on the left and right side

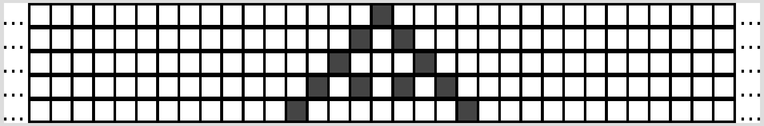
Rules:



Classical Cellular Automata

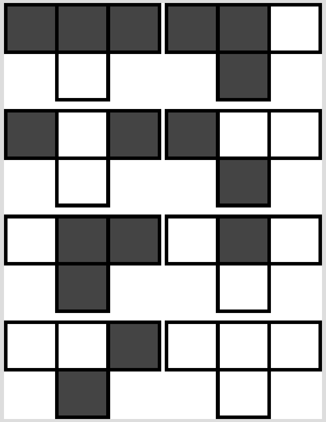
Example of 1-d CA

Iteration 5



Neighbourhood:
1 cell on the left and right side

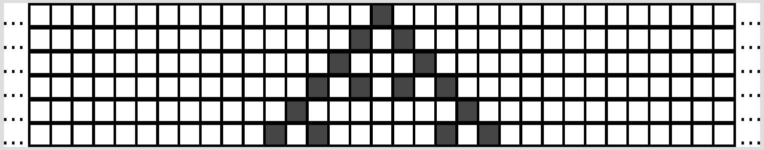
Rules:



Classical Cellular Automata

Example of 1-d CA

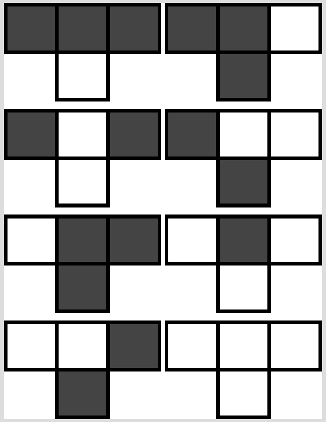
Iteration 6



Neighbourhood:

1 cell on the left and right side

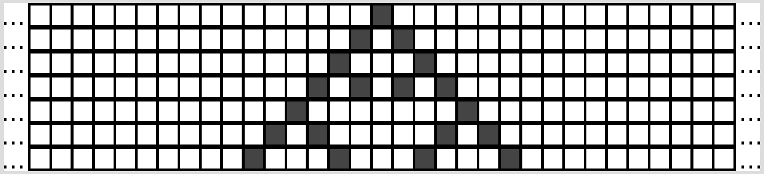
Rules:



Classical Cellular Automata

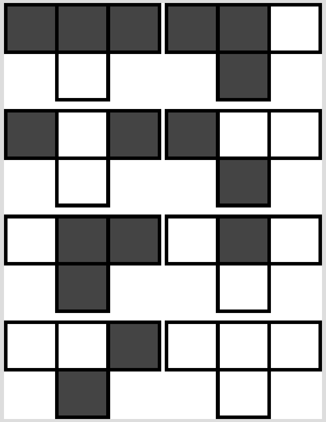
Example of 1-d CA

Iteration 7



Neighbourhood:
1 cell on the left and right side

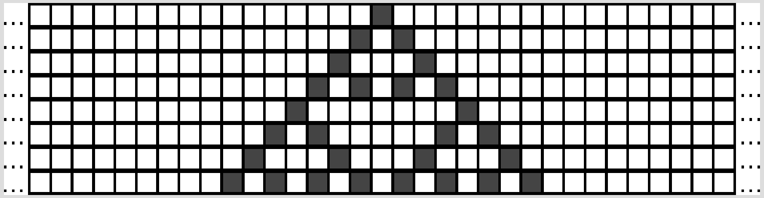
Rules:



Classical Cellular Automata

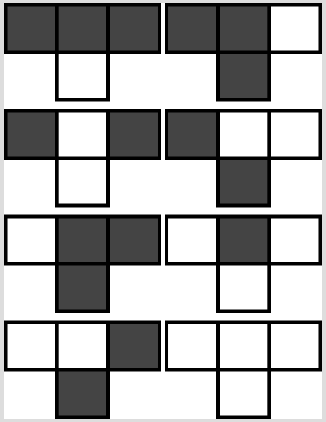
Example of 1-d CA

Iteration 8



Neighbourhood:
1 cell on the left and right side

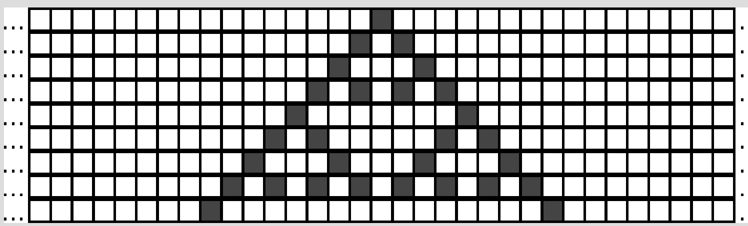
Rules:



Classical Cellular Automata

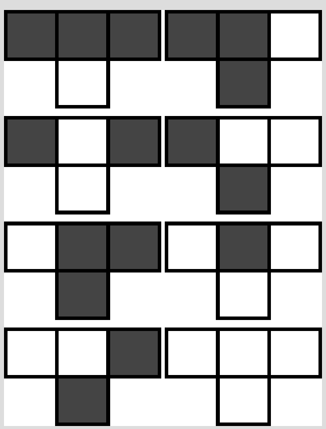
Example of 1-d CA

Iteration 9



Neighbourhood:
  1 cell on the left
  and right side

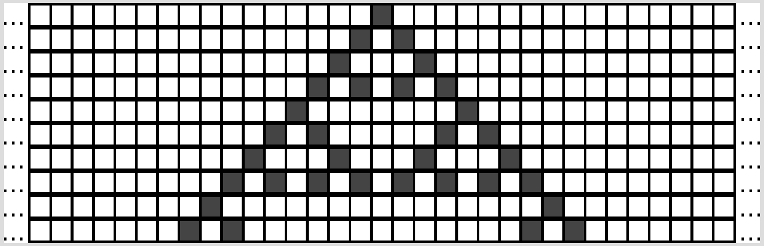
Rules:



Classical Cellular Automata

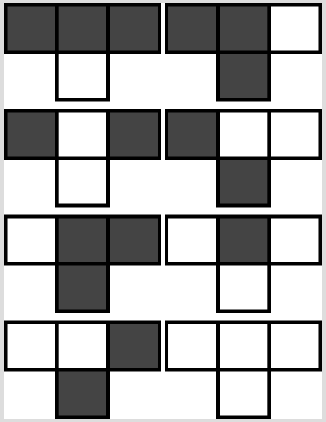
Example of 1-d CA

Iteration 10



Neighbourhood:
 1 cell on the left and right side

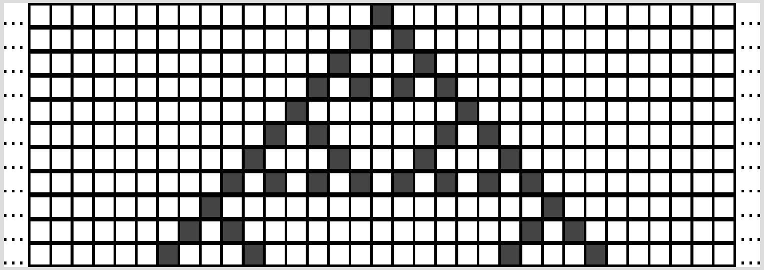
Rules:



Classical Cellular Automata

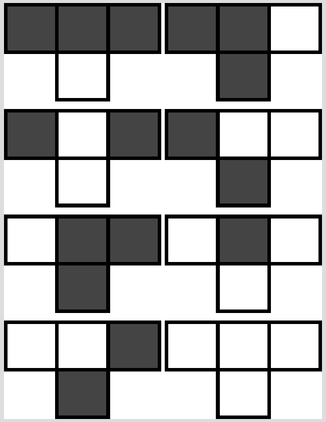
Example of 1-d CA

Iteration 11



Neighbourhood:
1 cell on the left and right side

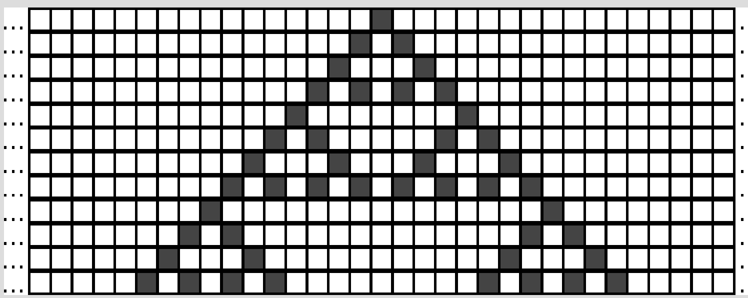
Rules:



Classical Cellular Automata

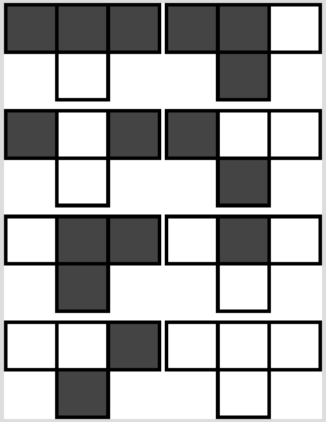
Example of 1-d CA

Iteration 12



Neighbourhood:
- 1 cell on the left and right side

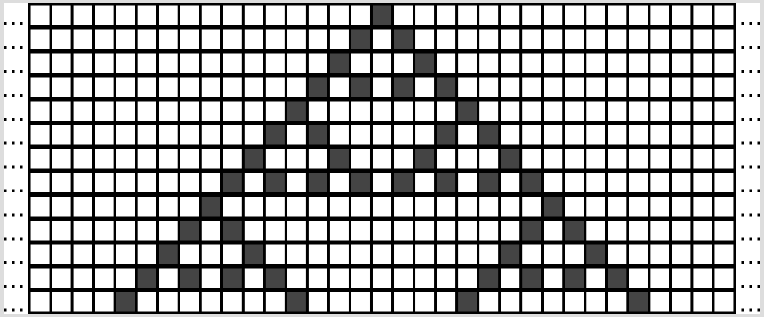
Rules:



Classical Cellular Automata

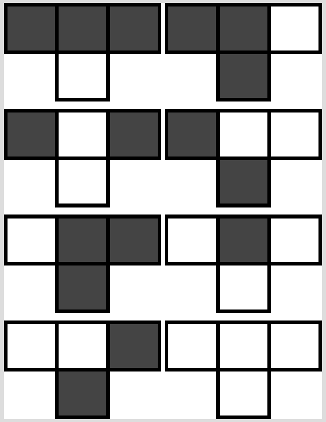
Example of 1-d CA

Iteration 13



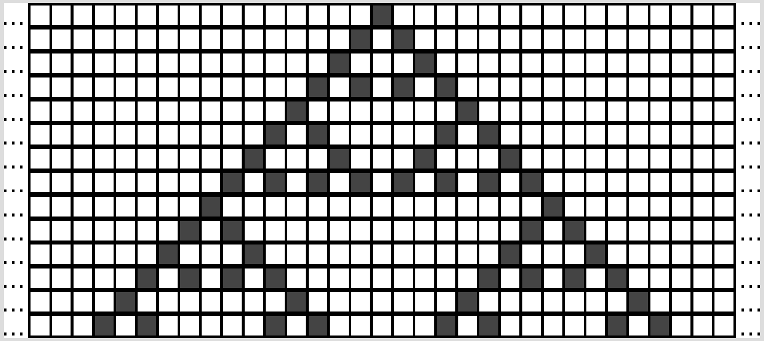
Neighbourhood:
1 cell on the left and right side

Rules:



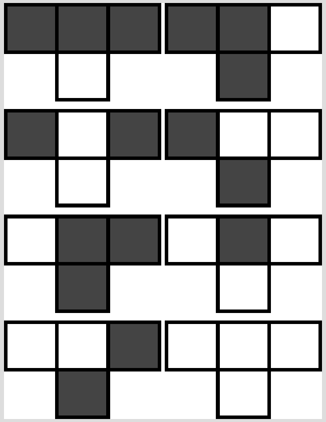
Classical Cellular Automata

Example of 1-d CA
Iteration 14



Neighbourhood:
1 cell on the left
and right side

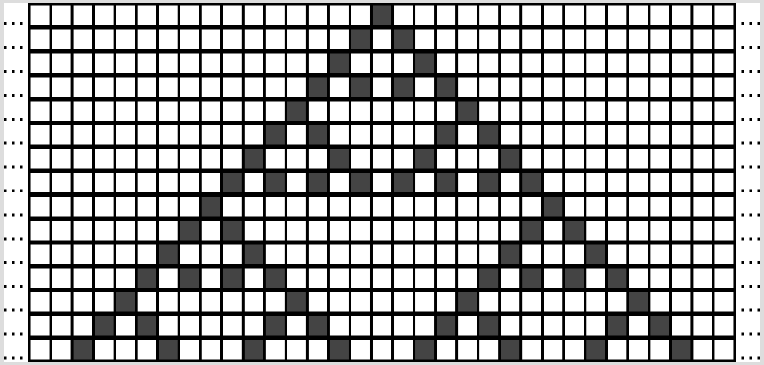
Rules:



Classical Cellular Automata

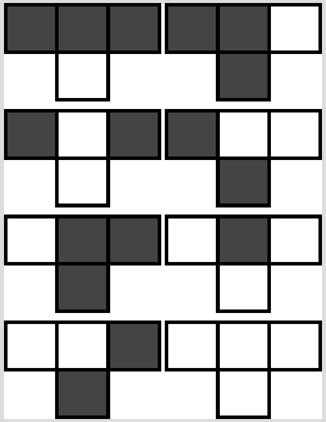
Example of 1-d CA

Iteration 15



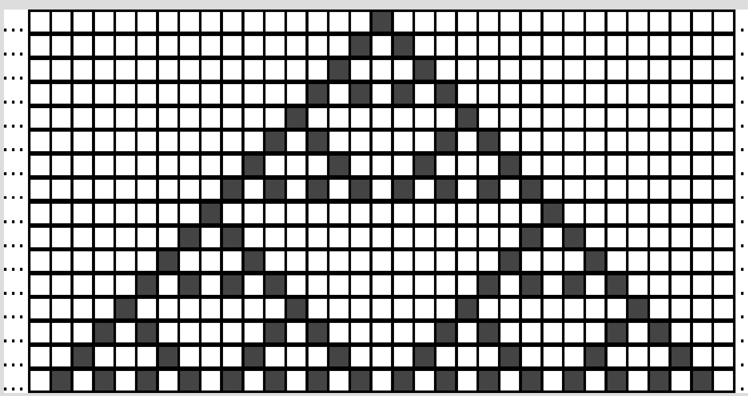
Neighbourhood:
1 cell on the left and right side

Rules:



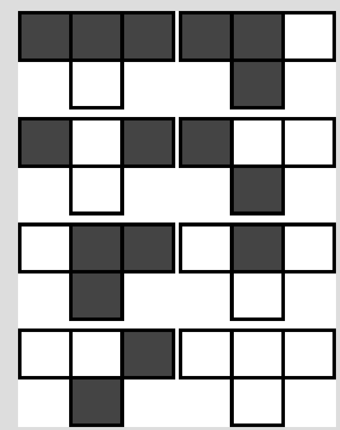
Classical Cellular Automata

Example of 1-d CA
Iteration 16



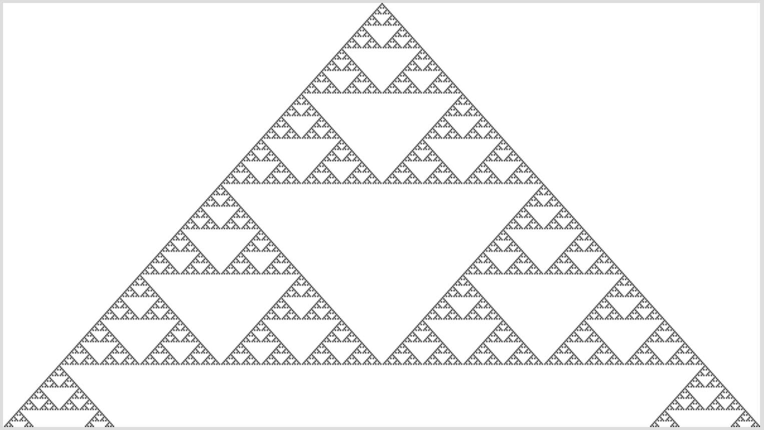
Neighbourhood:
1 cell on the left and right side

Rules:



Classical Cellular Automata

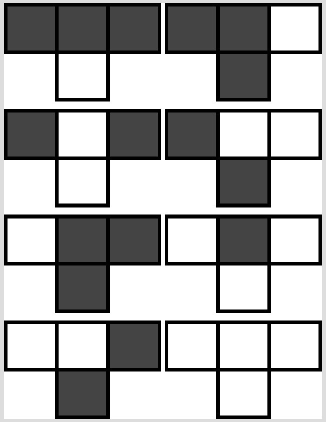
Example of 1-d CA
Iteration ~300



Sierpiński triangle (fractal)

Neighbourhood:
1 cell on the left and right side

Rules:



Classical Cellular Automata

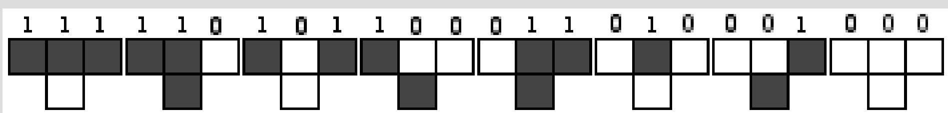
b) Formally

- 4-tuple (C, Σ, N , *f*)
- C - n-dimensional grid of cells
- Σ - set of possible states
- N - neighbourhood scheme (all possible)
- *f* - local transition function of type $\Sigma^N \rightarrow \Sigma$
- Σ^C - configuration
- *f* => F - global transition function of type $\Sigma^C \rightarrow \Sigma^C$

Classical Cellular Automata

Rules

- Rule for 1-d CA with simple neighbourhood



=01011010bin = 90dec

- 2^8 combinations → 1 byte for each set of rules
- Classification of rules to classes with respect to complexity
- Wolfram classes

Classical Cellular Automata

Wolfram classes

- Class1 – stable
- Class2 – periodic patterns
- Class3 – chaos
- Class4 – ”on the edge of chaos”
- ”Rule 110” and Conway's Game of Life

Classical Cellular Automata

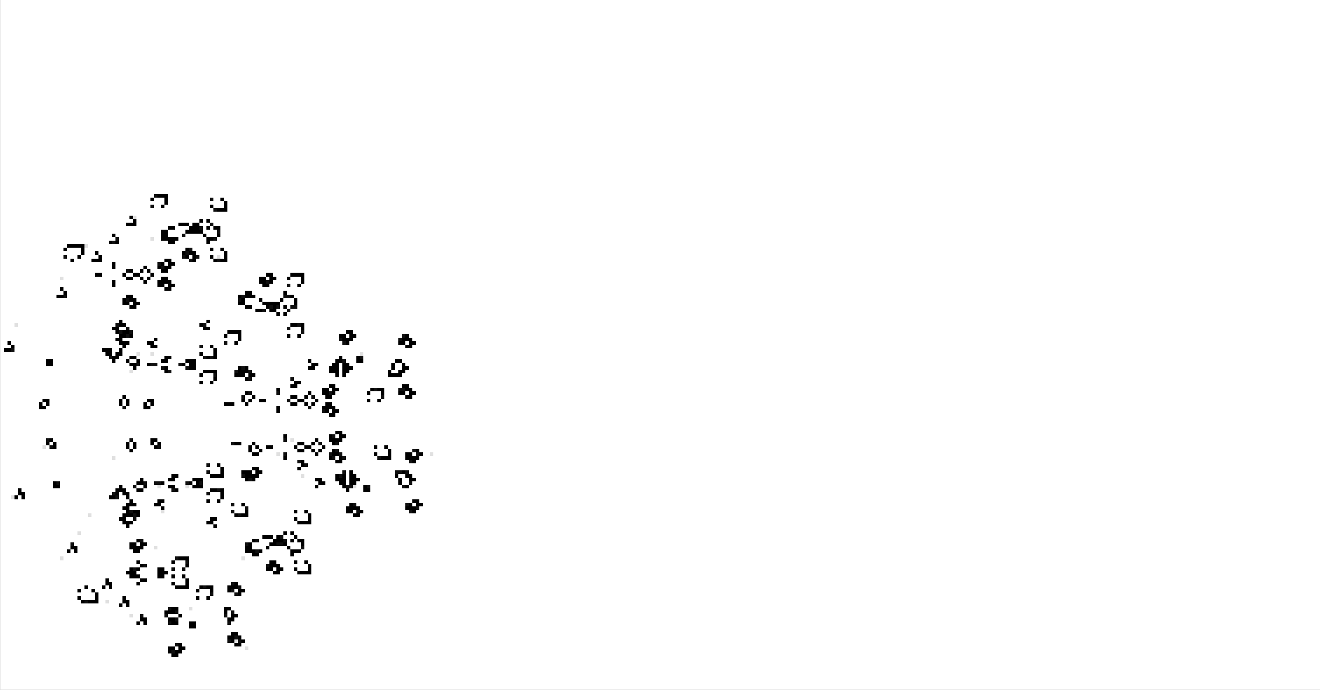
Game of Life

- The most famous CA
- 2 dimensions → configuration = 2d array
- 2 states: live or dead
- Neighbourhood made of 8 adjacent cells
- Rules:
  - A dead cell comes to life if it has exactly three living cells in its neighbourhood.
  - A cell remains living if it has two or three living cells in its neighbourhood.
  - A cell dies otherwise. (on overpopulation or loneliness)

Classical Cellular Automata

Game of Life

- GoL patterns
- demo
- Turing complete (TM as GoL)

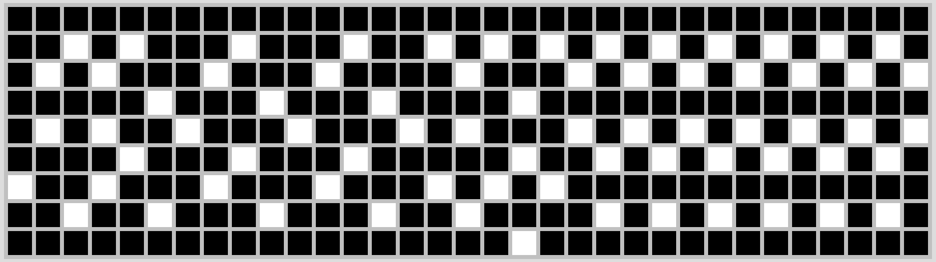


”Breeder”:
creates ”Gosper's Glider Guns”

”GGG” shoots ”gliders”

Reversible Cellular Automata

- Local & global transition function
- CA is reversible iff F is bijection
- Garden of Eden configurations (without predecessor)
- Orphan pattern



discovered by R. Banks in 1971

Quantum Cellular Automaton

- cellular automata are themselves physics-like models of computation
- seems natural to study quantum extensions of CA
- principle of computation is in quantum cells interaction
- reduced need for environment interaction, and so is possibility of decoherence
- no need to address each cell (Qbit) separately

QCA - History

- 1965 - the first example of a quantum cellular automata was Feyman's "quantum checkerboard"
- 1988 Grossing and Zeilinger attempted to introduce the concept of quantum cellular automata
- 1990 Norm Marglous described Parallel Quantum Computation.
- 1995 the first successful model of one dimensional quantum cellular automata was introduced by John Watrous.

What is QCA?

- not only generally accepted QCA model
- main common signs of various QCA definitions:
  - d-dimensional lattice of identical finite-dimensional quantum systems
  - finite set of states
  - finite neighbourhood scheme of single cell
  - set of local transition functions
  - global evolution function represents the discrete time evolution of strictly finite cell lattice

LQCA (1)

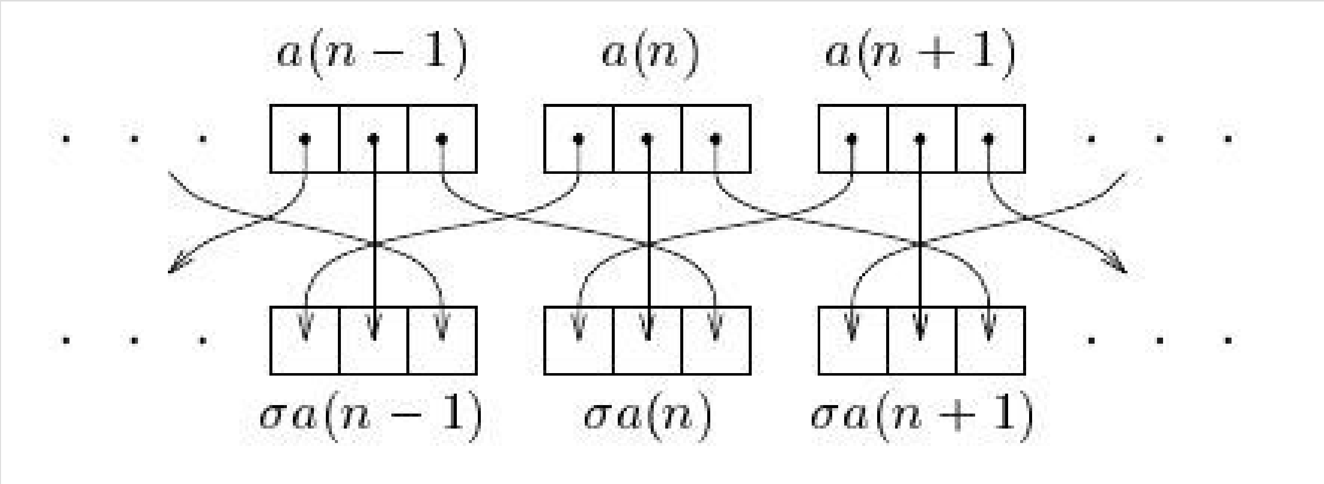
- 1-dimensional QCA defined by John Watrous
- first QCA researched in depth
- 4-tuple $A = (\Sigma, q, N, \delta)$, where (with $q\Sigma = \{q\} \cup \Sigma$):
  - Σ is a finite set of symbols ("the alphabet", possible basic states each cell may take);
  - q is a symbol such that $q \notin \Sigma$ ("the quiescent symbol", special state for empty cells);
  - N is a set of n successive signed integers ("the neighbourhood", telling which cell is next to whom);
  - δ : ΉH(qΣ)n → ΉHqΣ is a function from super-positions of n symbols words to super-positions of one symbol words ("the local transition function",the way a cell interacts with its neighbours).

LQCA (2)

- homogenous and synchronous application of transition function gives rise to global evolution Δ
- Δ must be unitary in order to have physically acceptable model of computation
- unitarity of global evolution function is really non-trivially related to the definition of the transition function δ
- LQCA is too loose formalism
- Pablo Arrighi managed to algebraically characterize unitary LQCA by adding constraints into the model

Partitioned QCA (2)

- restricted LQCA
- avoids non-unitary global evolution funciton Δ
- each cell in lattice is partitioned in to three sub-cells
- next state depends only on the states of the left sub-cell of the right neighbour, the middle sub-cell of the cell itself, and the right sub-cell of the left neighbour

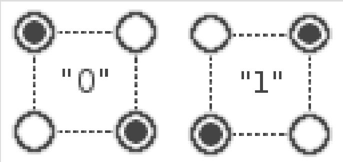


Partitioned QCA (2)

- It was shown that any Quantum Turing Machine can be simulated by PQCA with constant slowdown and every PQCA can be simulated by QTM with linear slowdown.

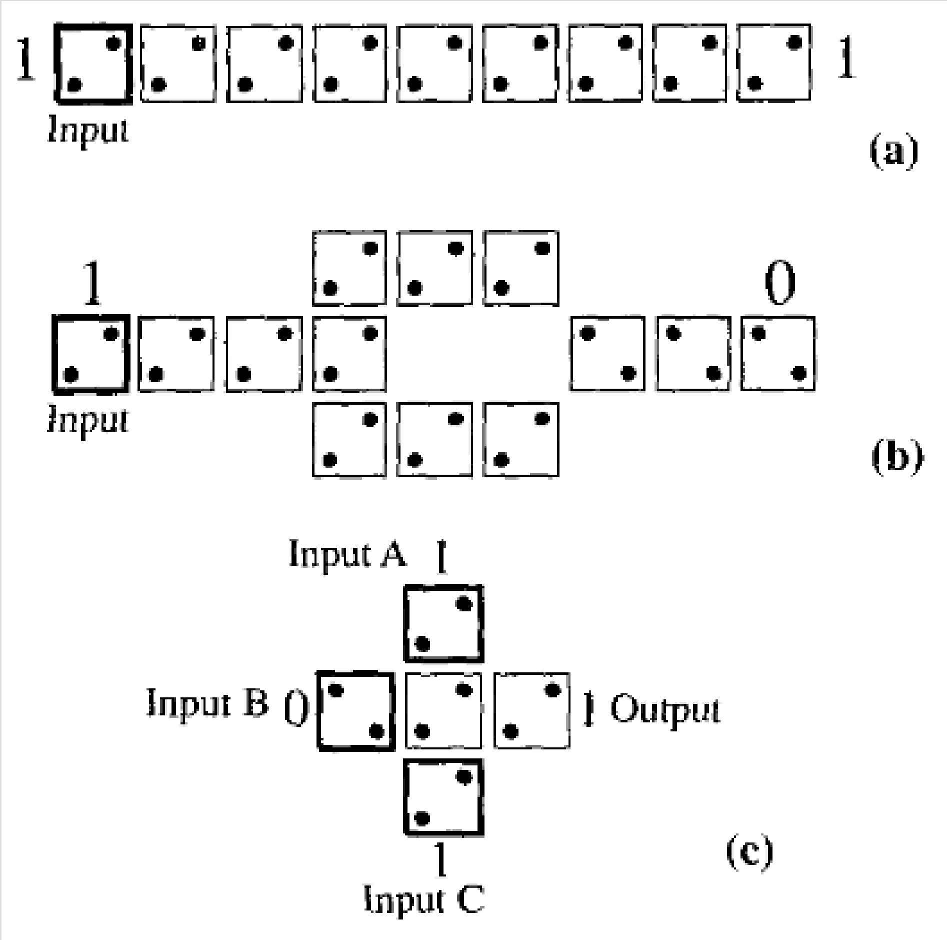
Quantum-Dot CA

- classical CA implemented in quantum mechanical structures
- cell consists of four quantum dots forming a square
- ectrons can tunnel between dots
- Coulomb repulsion will force them to dots on opposite corners
- ground states representing zero and one



Quantum-Dot CA

Basic circuits:



Wire

NOT gate

Majority gate

Majority gate

Inputs	C	A	B	Output
	0	0	0	0
	0	0	1	0
	0	1	0	0
	0	1	1	1
	1	0	0	0
	1	0	1	1
	1	1	0	1
	1	1	1	1

Output = A **AND** B if C == 0
A **OR** B if C == 1

Conclusion

- concept of Quantum Cellular Automata is currently in its early phase
- many of important definitions missing or appeared most recently
- many interesting discoveries can be expected in near future
- QCA formalism seems as promising framework for building robust and scalable quantum computers
- Quantum Dot Cellular Automata seems like suitable replacement for current CMOS technology

The end

Thank you

(Demo)

Q&A