Nature-Based Problems in Cellular Automata

Martin Kutrib

Institut für Informatik, Universität Giessen

Overview

- The Dawn of Cellular Automata
- The French Flag Problem
- Synchronization of Growing Arrays
- Oblivious Cellular Automata
- The Fault-Tolerant Early Bird Problem

Overview

- The Dawn of Cellular Automata
- The French Flag Problem
- Synchronization of Growing Arrays
- Oblivious Cellular Automata
- The Fault-Tolerant Early Bird Problem

Overview

- The Dawn of Cellular Automata
- The French Flag Problem
- Synchronization of Growing Arrays
- Oblivious Cellular Automata
- The Fault-Tolerant Early Bird Problem

Overview

- ➜ The Dawn of Cellular Automata
- ➜ The French Flag Problem
- ➜ Synchronization of Growing Arrays
- ➜ Oblivious Cellular Automata
- ➜ The Fault-Tolerant Early Bird Problem

Overview

- The Dawn of Cellular Automata
- The French Flag Problem
- Synchronization of Growing Arrays
- Oblivious Cellular Automata
- The Fault-Tolerant Early Bird Problem

The Dawn of Cellular Automata

Fundamental questions [John von Neumann 1949]:

The Dawn of Cellular Automata

Fundamental questions [John von Neumann 1949]:

(A) Logical universality. When is a class of automata logically universal? Also, with what additional attachments is a single automaton logically universal?

The Dawn of Cellular Automata

Fundamental questions [John von Neumann 1949]:

(A) Logical universality. When is a class of automata logically universal? Also, with what additional attachments is a single automaton logically universal?

(B) Constructibility. Can an automaton be constructed by another automaton? What class of automata can be constructed by one, suitably given, automaton?

The Dawn of Cellular Automata

Fundamental questions [John von Neumann 1949]:

(A) Logical universality. When is a class of automata logically universal? Also, with what additional attachments is a single automaton logically universal?

(B) Constructibility. Can an automaton be constructed by another automaton? What class of automata can be constructed by one, suitably given, automaton?

(C) Construction-universality. Can any one, suitably given, automaton be construction-universal, that is, be able to construct every other automaton?

The Dawn of Cellular Automata

Fundamental questions [John von Neumann 1949]:

(D) Self-reproduction. Can any automaton construct other automata that are exactly like it? Can it be made, in addition, to perform further tasks?

The Dawn of Cellular Automata

Fundamental questions [John von Neumann 1949]:

(D) Self-reproduction. Can any automaton construct other automata that are exactly like it? Can it be made, in addition, to perform further tasks?

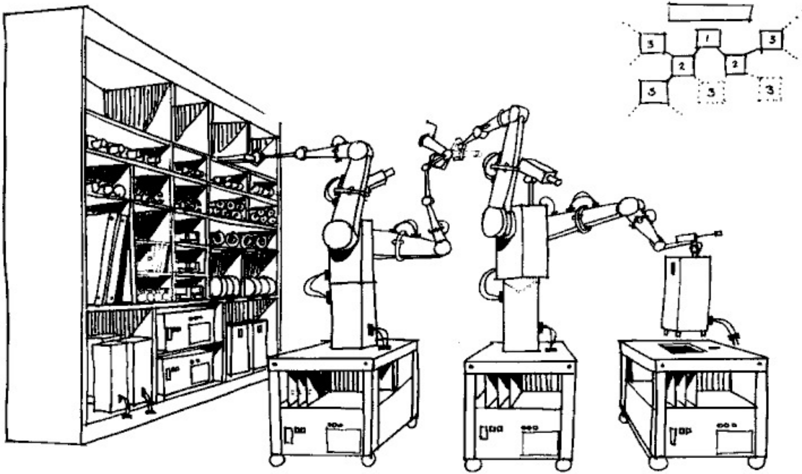
(E) Evolution. Can the construction of automata by automata progress from simpler types to increasingly complicated types? Also, can this evolution go from less efficient to more efficient automata?

A self-reproducing machine is an artificial construct that is theoretically capable of autonomously manufacturing a copy of itself using raw materials taken from its environment.

A self-reproducing machine is an artificial construct that is theoretically capable of autonomously manufacturing a copy of itself using raw materials taken from its environment.

- What is an artificial construct?
- What is raw material?
- What is an environment?

Von Neumann's Kinematic Model:



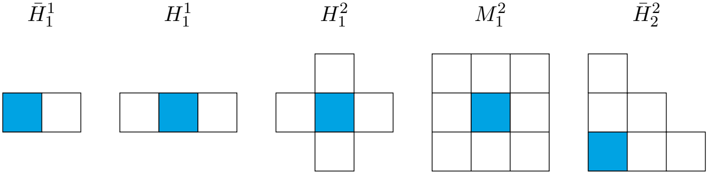
Von Neumann's Cellular Automata:

- Stanisław Ulam suggested to employ a mathematical device which is a multitude of interconnected machines operating in parallel to form a larger machine.

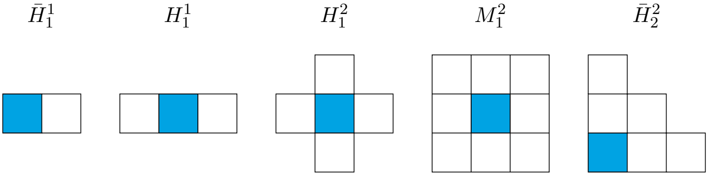
Von Neumann's Cellular Automata:

- Stanisław Ulam suggested to employ a mathematical device which is a multitude of interconnected machines operating in parallel to form a larger machine.
- d-dimensional grid of cells (machines).
- Synchronous behavior.
- Cells are deterministic finite automata (simplicity).
- All cells are identical (homogeneity).
- One interconnection scheme (homogeneous local communication structures).

Interconnection schemes:

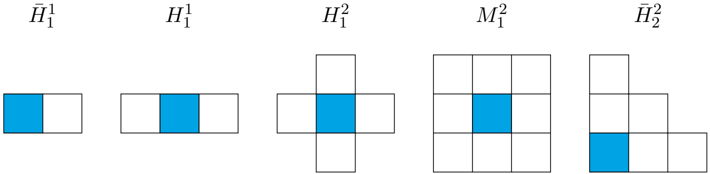


Interconnection schemes:



Quiescent state: If a cell itself and all of its neighbors are in the quiescent state, the cell remains in the quiescent state.

Interconnection schemes:



Quiescent state: If a cell itself and all of its neighbors are in the quiescent state, the cell remains in the quiescent state.

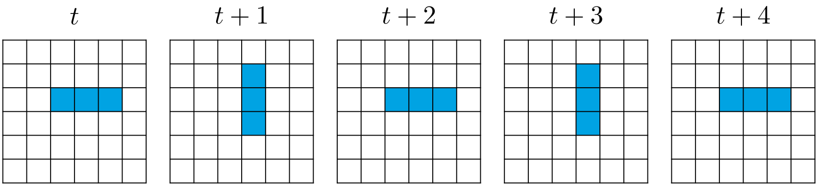
> Larger machines are patterns of cell states, embedded in space.

Example:

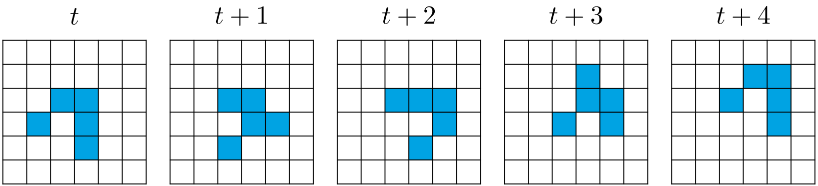
- Two dimensions.
- The state set is $\{0, 1\}$.
- Each cell is connected to its eight immediate neighbors.

Example:

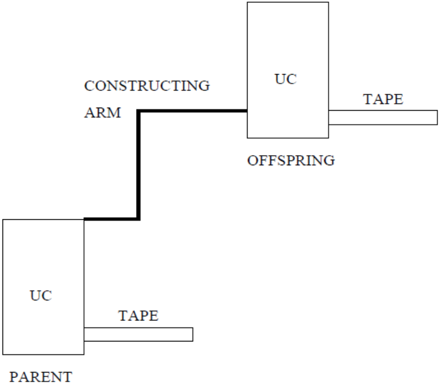
- Two dimensions.
- The state set is $\{0, 1\}$.
- Each cell is connected to its eight immediate neighbors.
- The local transition function is defined by the sum of the states of the neighbors and of the cell itself. In particular:
- A cell enters state 1, if the sum is three.
- A cell keeps its current state, if the sum is four.
- Otherwise the cell enters state 0.



t
$t+1$
$t+2$
$t+3$
$t+4$


t
$t+1$
$t+2$
$t+3$
$t+4$

John von Neumann succeeded. He constructed a 29-state cellular automaton which is contruction-universal, self-reproducing, and Turing-universal.

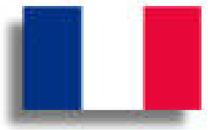


The French Flag Problem

Origin of the Problem:

- Problem of pattern formation of simple axial patterns.

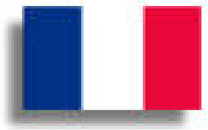
The French Flag Problem



Origin of the Problem:

- Problem of pattern formation of simple axial patterns.
- To model the determination of a pattern in a tissue having three regions of cells with discrete properties and sharp bounds between them.

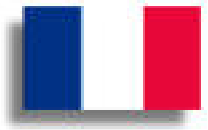
The French Flag Problem



Origin of the Problem:

- Problem of pattern formation of simple axial patterns.
- To model the determination of a pattern in a tissue having three regions of cells with discrete properties and sharp bounds between them.
- Concept of a morphogen, that is, a signaling molecule that regulates the pattern formation.

The French Flag Problem



Origin of the Problem:

- Problem of pattern formation of simple axial patterns.
- To model the determination of a pattern in a tissue having three regions of cells with discrete properties and sharp bounds between them.
- Concept of a morphogen, that is, a signaling molecule that regulates the pattern formation.
- High concentrations activate a blue gene, lower concentrations activate a white gene, where red indicates cells below the necessary concentration threshold.

The Problem in Terms of Cellular Automata:

- Construct a finite but arbitrary large one-dimensional cellular automaton with nearest neighbor connections that

The Problem in Terms of Cellular Automata:

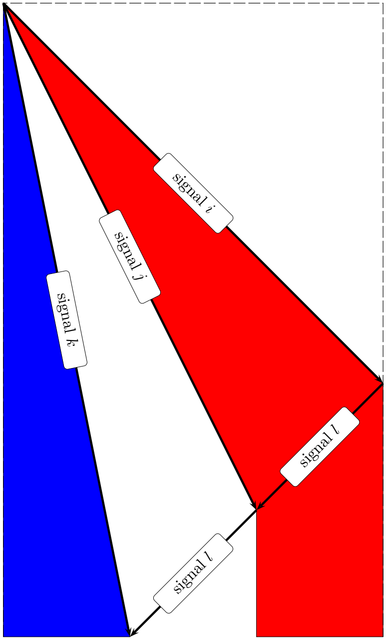
- Construct a finite but arbitrary large one-dimensional cellular automaton with nearest neighbor connections that
- when started with all cells in the quiescent state turns into a French flag upon excitation from the outside world at one of the ends.

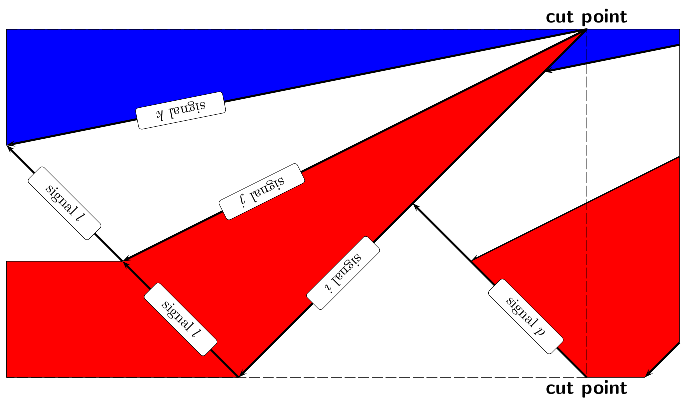
The Problem in Terms of Cellular Automata:

- Construct a finite but arbitrary large one-dimensional cellular automaton with nearest neighbor connections that
- when started with all cells in the quiescent state turns into a French flag upon excitation from the outside world at one of the ends.
- The colors are represented by states and the blue region appears at the end excited.

The Problem in Terms of Cellular Automata:

- Construct a finite but arbitrary large one-dimensional cellular automaton with nearest neighbor connections that
- when started with all cells in the quiescent state turns into a French flag upon excitation from the outside world at one of the ends.
- The colors are represented by states and the blue region appears at the end excited.
- Moreover, if the array is cut into two or more pieces, it is required that all pieces turn into French flags with the same orientation as the original.


signal i
signal j
signal k
signal l
signal l


cut point
signal k
signal j
signal l
signal i
signal l
signal p
cut point

Cellular Automata Without Local Polarity:

- Since a solution turns the entire array into an asymmetric French flag, the array exhibits a polarity.
- This global polarity can be achieved by single cells exhibiting a local polarity.
- This local polarity is an unlikely activity for cells of organisms.
- Cellular automata having no polarity on the cellular level are called symmetric.

Simulating Local Polarity:

- Immediately after emitting signal i the excited cell sends another signal which labels the cells passed through cyclically by 1,2,3,1,. . .
- Now it is easy for a cell to recognize which state received comes from right or left.

Synchronization of Growing Arrays

Origin of the Problem:

- ➜ Simulation of pigmentation patterns on the shells of sea-snails.
- ➜ It is supposed that glands stop their action synchronously at the same time,
- ➜ even though their number was growing during the synchronization process.

The Firing Squad Synchronization Problem:

- Construct finite but arbitrary large one-dimensional cellular automaton with nearest neighbor connections that grows at the fixed rates p cells in q steps at the right and r cells in t steps at the left, which

The Firing Squad Synchronization Problem:

- Construct finite but arbitrary large one-dimensional cellular automaton with nearest neighbor connections that grows at the fixed rates p cells in q steps at the right and r cells in t steps at the left, which
- when started with all but the leftmost cells in the quiescent state

The Firing Squad Synchronization Problem:

- Construct finite but arbitrary large one-dimensional cellular automaton with nearest neighbor connections that grows at the fixed rates p cells in q steps at the right and r cells in t steps at the left, which
- when started with all but the leftmost cells in the quiescent state
- synchronizes in such a way that all cells enter a distinguished state, the firing state, for the first time at the same time step.

Base of the Algorithm:

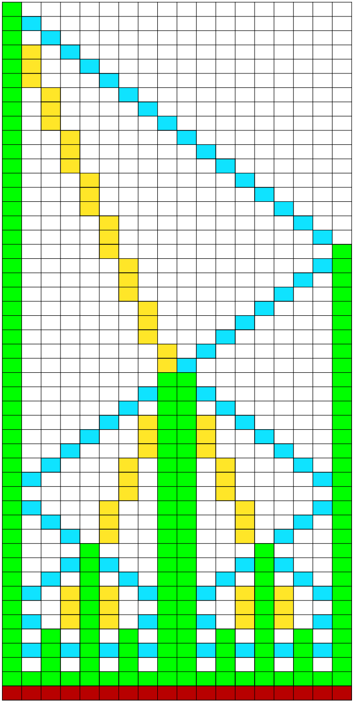
- The problem can be solved by dividing the array in two, four, eight etc. parts of (almost) the same length until all cells are cut-points.
- Exactly at this time the cells will fire synchronously.

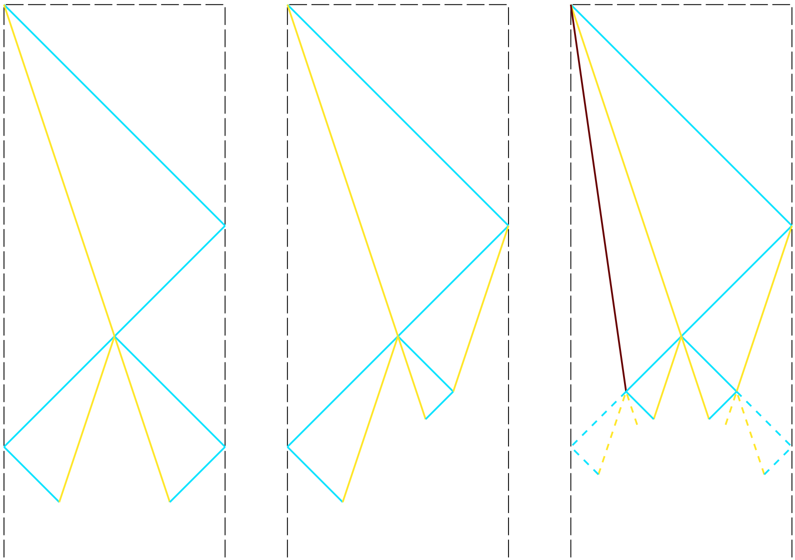
Base of the Algorithm:

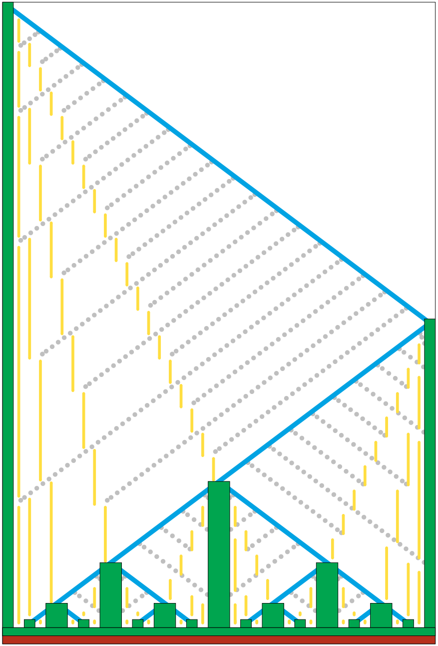
- The problem can be solved by dividing the array in two, four, eight etc. parts of (almost) the same length until all cells are cut-points.
- Exactly at this time the cells will fire synchronously.
- The divisions are performed recursively. At first the array is divided into two parts. Then the process is applied to both parts in parallel, etc.

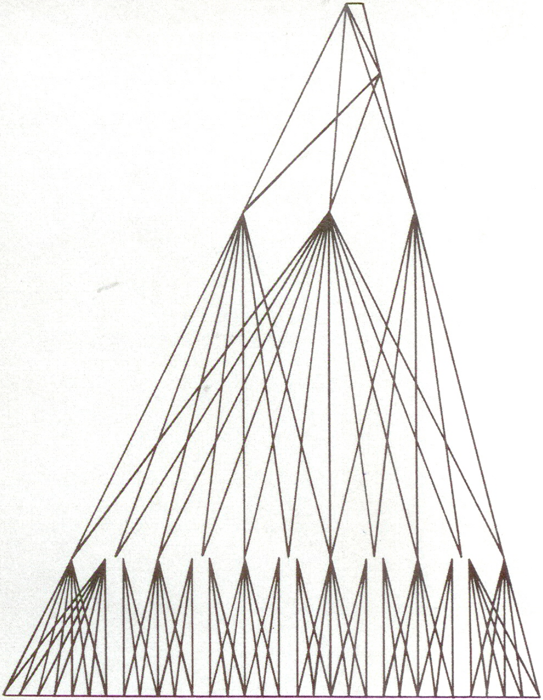
Base of the Algorithm:

- The problem can be solved by dividing the array in two, four, eight etc. parts of (almost) the same length until all cells are cut-points.
- Exactly at this time the cells will fire synchronously.
- The divisions are performed recursively. At first the array is divided into two parts. Then the process is applied to both parts in parallel, etc.
- In order to divide the array into two parts, the general sends two signals $S1$ and $S2$ to the right.
- Signal $S1$ moves with speed one, and signal $S2$ with speed $1/3$.









Oblivious Cellular Automata

Origin of the Problem:

- Another phenomenon occurring in nature is obliviousness.
- In order to be economic, information that has not been used for a certain time is supposed to be of little relevance and therefore may be forgotten.
- If it is possible to perform any computation of classical cellular automata by oblivious cellular automata, then the constructions can be seen as strategies of self-repair (with respect to the faults caused by obliviousness).

Modelling Obliviousness:

- Given a cellular automaton, let $\tau(i, p, t)$ denote the last time step between time 0 and time t at which cell i was in state p.
- If cell i was never in state p until time t then $\tau(i, p, t)$ is equal to ∞.

Modelling Obliviousness:

- Given a cellular automaton, let $\tau(i, p, t)$ denote the last time step between time 0 and time t at which cell i was in state p.
- If cell i was never in state p until time t then $\tau(i, p, t)$ is equal to ∞.
- The transition function sends a cell to its successor state p at time t provided that the cell has been in state p within the last $\varphi(t)$ time steps.
- Otherwise state p has been forgotten and the cell enters the quiescent state instead.

Theorem

Let M be a (classical) cellular automaton and φ be monotonically increasing and unbounded. Then there is an oblivious cellular automaton which R-simulates M.

The Fault-Tolerant Early Bird Problem

Origin of the Problem:

- In massively parallel computing systems each single component is subject to failure, such that the probability of misoperations and loss of function of the whole system increases with the number of its elements.
- Biological systems may serve as good examples for fault-tolerant systems.
- Due to the necessity to function normally even in case of certain failures of their components, nature developed mechanisms which invalidate the errors.

Defective Cellular Automata:

Self-diagnosis. Each cell has a self-diagnosis circuit which is run once before the actual computation. The result is stored locally in a register.

Defective Cellular Automata:

Self-diagnosis. Each cell has a self-diagnosis circuit which is run once before the actual computation. The result is stored locally in a register.

Recognition of defective neighbors. In this way intact cells can detect whether their neighbors are defective or not when they receive the first states.

Defective Cellular Automata:

Self-diagnosis. Each cell has a self-diagnosis circuit which is run once before the actual computation. The result is stored locally in a register.

Recognition of defective neighbors. In this way intact cells can detect whether their neighbors are defective or not when they receive the first states.

Static defects. Some cells are initially defect, and during the actual computation no new defects may occur.

Defective Cellular Automata:

Self-diagnosis. Each cell has a self-diagnosis circuit which is run once before the actual computation. The result is stored locally in a register.

Recognition of defective neighbors. In this way intact cells can detect whether their neighbors are defective or not when they receive the first states.

Static defects. Some cells are initially defect, and during the actual computation no new defects may occur.

Transmission. A defective cell is able to transmit information but cannot process or modify it.

The Early Bird Problem:

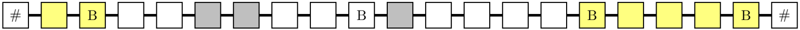
- Initially all cells are quiescent.
- Quiescent cells may be excited from the outside world (a bird has arrived at the cell).
- Distinguish between the first (early) and later birds.

Five-State Algorithm for Classical Cellular Automata:

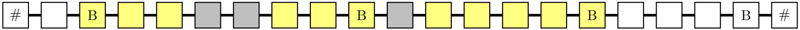
```
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . B . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . < B > . . . . . . . . . . . . . . . . .
. . . . . . . . B . . . < < B > > . . . . . . . . . . . . . . . .
. . . . . . . < B > . < < < B > > > . . . . . . . . . . . . . . .
. . . . . . < < B > * < < < B > > > > . . . . B . B . . . . . . .
. . . . . < < < B * * * < < B > > > > > . . < B * B > . . . . . .
. . . . < < < < B * * < * < B > > > > > > < < B * B > > . . . . .
. . . < < < < < B * < * < * B > > > > > * * < B * B > > > . . . .
. . < < < < < < B < * < * * B > > > > * > < * B * B > > > > . . .
. < < < < < < < < * < * * * B > > > * > * * * B * B > > > > > . .
< < < < < < < < < < * * * * B > > * > * > * * B * B > > > > > > .
* < < < < < < < < < * * * * B > * > * > * > * B * B > > > > > > >
< * < < < < < < < < * * * * B * > * > * > * > B * B > > > > > > *
* < * < < < < < < < * * * * B * * > * > * > * > * B > > > > > * >
< * < * < < < < < < * * * * B * * * > * > * > * > B > > > > * > *
* < * < * < < < < < * * * * B * * * * > * > * > * > > > > * > * >
< * < * < * < < < < * * * * B * * * * * > * > * > > > > * > * > *
```

For Defective Cellular Automata, Combination of:

- ➜ five-state algorithm (in intact regions)



- ➜ synchronization (in defective regions)



A basic principle of the solution algorithm is that signals circulate between two bird cells until the next step of the algorithm is completed.

Sketch of Synchronization in Defective Regions:

- When a bird arrives, it emits signals to the left and right at each time step until it receives a signal from another bird.
- Now it bounces arriving signals.
- The difference of the numbers of signals emitted by the birds is twice the age difference of the birds.

Sketch of Synchronization in Defective Regions:

- The birds convert the number of emitted signals into binary.
- The binary number is sent as signals which circulate, too.
- The difference of binary numbers of neighboring birds is computed and divided by two, in order to determine the elder bird.
- Again, the difference is sent as signals which circulate.

Sketch of Synchronization in Defective Regions:

- All differences in between two farther birds are summed up in a specific way in order to compare their ages.
- The birds in between them are deleted and behave like defective cells.
- The last step of the algorithm is repeated until all birds have the same age and, thus, are the global early birds.