### Introductory Lecture on Cellular Automata

Modified and upgraded slides of

Martijn Schut schut@cs.vu.nl Vrij Universiteit Amsterdam

Lubomir Ivanov Department of Computer Science Iona College

and anonymous from Internet

• Conway's Game of Life

Cellular Automata

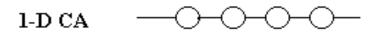
Self Reproduction

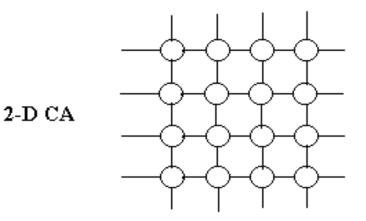
• Universal Machines



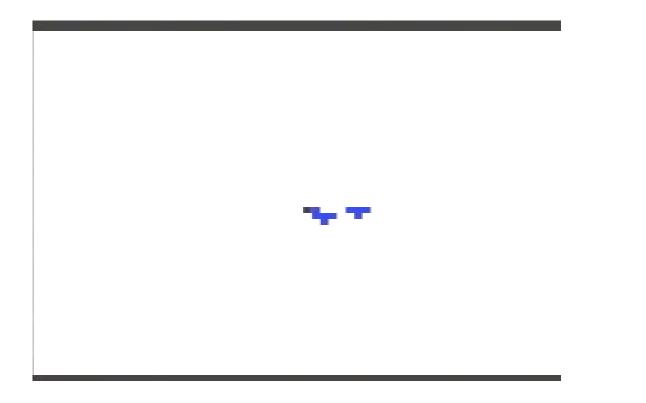
# **Cellular Automata**

- A Cellular Automaton is a model of a parallel computer
- A CA consists of processors (cells), connected usually in an n-dimensional grid





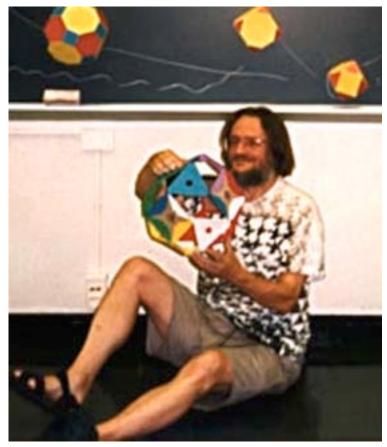
### **EXAMPLE: Life - The Game**



Movement of black patterns on grid matrix

### **History of Cellular Automata**

- Original experiment created to see if simple rule system could create "universal computer"
- Universal Computer (Turing): a machine capable of emulating any kind of information processing through simple rule system
- late 1960's: John Conway invents "Game of Life"

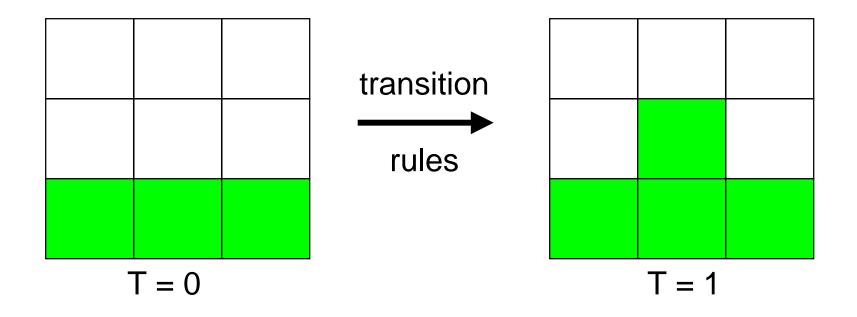


John H. Conway



- Simplest possible universe capable of computation
- Basic design: rectangular grid of "living" (on) and "dead" (off) cells
- Complex patterns result from simple structures
- In each generation, cells are governed by three simple rules
- Which patterns lead to stability? To chaos?

• A cell dies or lives according to some *transition rule* 



- As in Starlogo, the world is round (flips over edges)
- How many rules for Life? 20, 40, 100, 1000?

Three simple rules

- dies if number of alive neighbor cells =< 2 (*loneliness*)
- dies if number of alive neighbor cells >= 5 (*overcrowding*)
- lives is number of alive neighbor cells = 3 (*procreation*)

This means that in original "Game of Life" when the cell has 4 alive neighbors, then its state remains as it was.

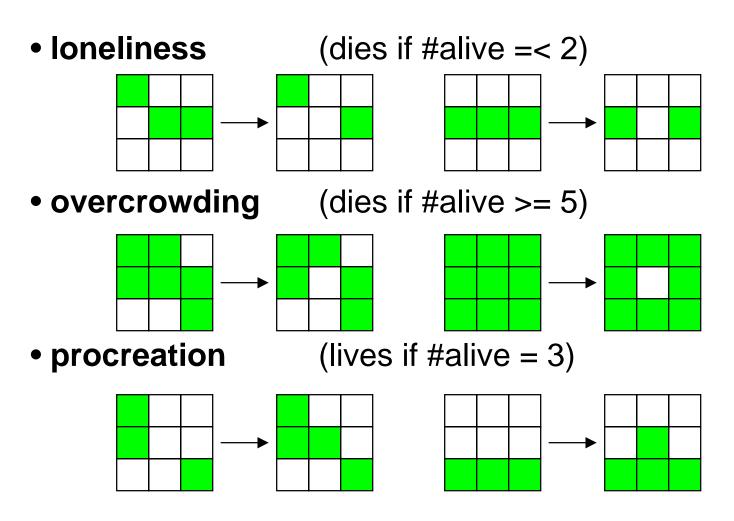
### Another variant of Conway's Rules

- Death: if the number of surrounding cells is less than 2 or greater than 3, the current cell dies
- Survival: if the number of living cells is exactly 2, or if the number of living cells is 3 (including the current cell), maintain status quo
- Birth: if the current cell is dead, but has three living cells surrounding it, it will come to life

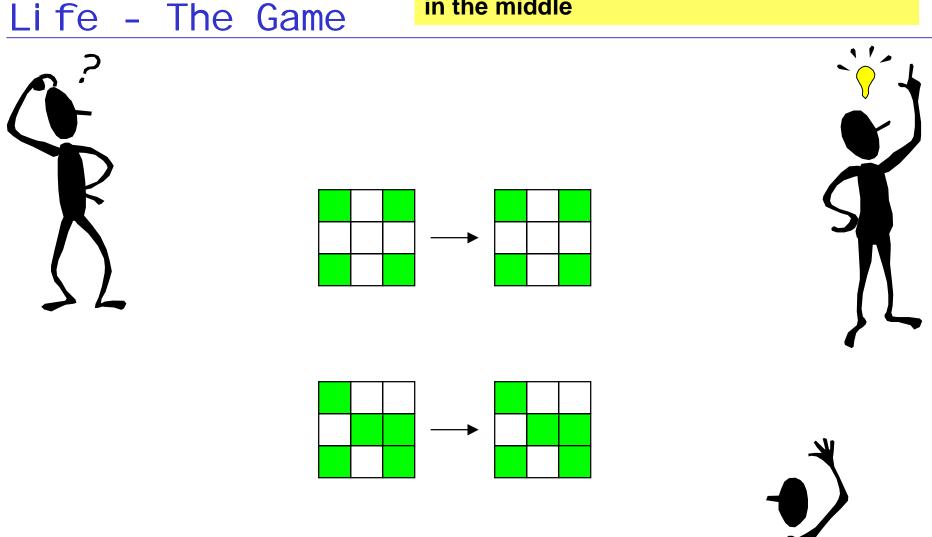
#### Life - The Game

# Here the rules are applied only to the cell in the middle

Examples of the rules



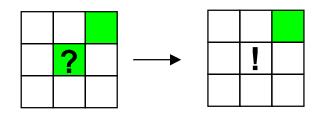
### Here the rules are applied only to the cell in the middle

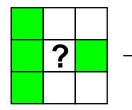


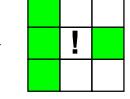
Cell has four alive neighbors so its state is preserved

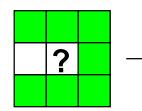


#### Life - The Game

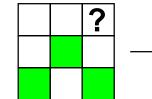


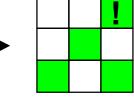


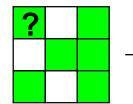


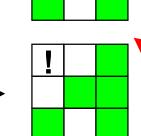






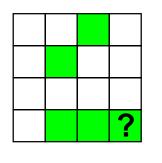


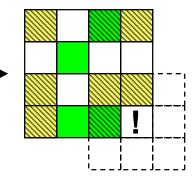


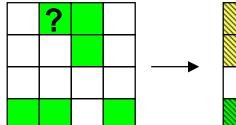


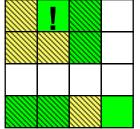
Rap-around the east and west

and the north and south



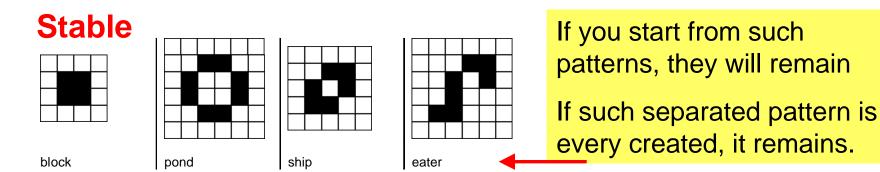




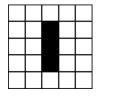


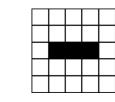
What happens at the frontiers?

#### Life - Patterns



**Periodic** 

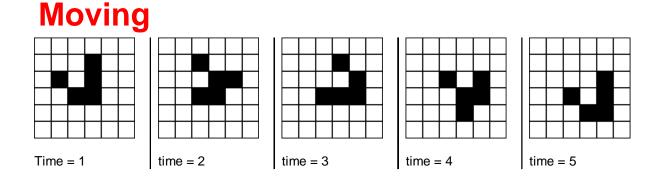




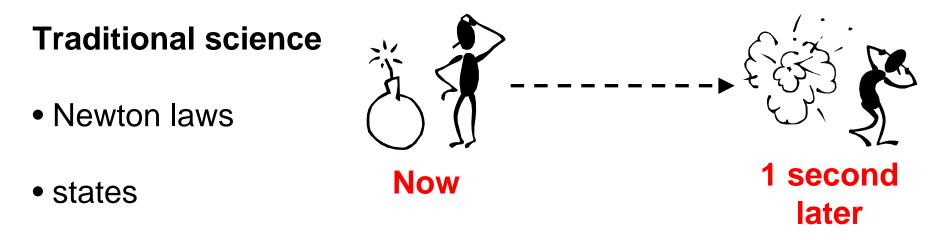
These patterns oscillate with certain periods, here the period is two, please analyse

time = 1

time = 2



#### Cellular Automata - Introduction

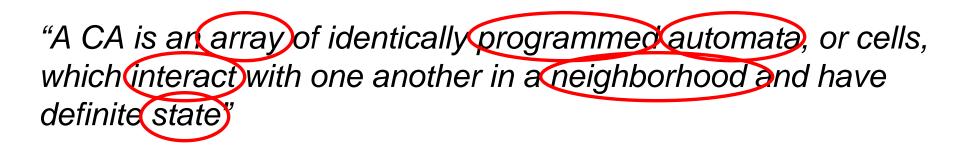


• problem: detailed description of states impossible etc etc

#### Heisenberg principle

- states that it is impossible to precisely know the speed and the location of a particle
- basis of quantum theory

#### Beyond Life - Cellular Automata



### In essence, what are Cellular Automata?

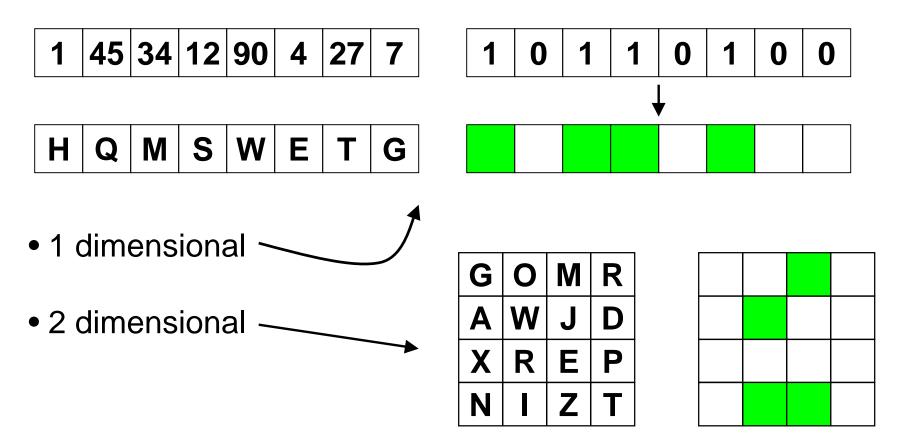
- Computer simulations which emulate the laws of nature
- Discrete time/space logical universes
- **Complexity from simple rule set:** reductionist approach
- Deterministic local physical model
- Rough estimation of nature: no precision
- **Does not reflect 'closed sphere' life:** can achieve same end results given rules and initial conditions

# **Simulation Goals**

- Avoid extremes: patterns that grow too quickly (unlimited) or patterns that die quickly
- Desirous behavior:
  - No initial patterns where unlimited growth is obvious through simple proof
  - Should discover initial patterns for which this occurs
  - Simple initial patterns should grow and change before ending by:
    - fading away completely
    - stabilizing the configuration
    - oscillating between 2 or more stable configurations
  - Behavior of population should be relatively unpredictable

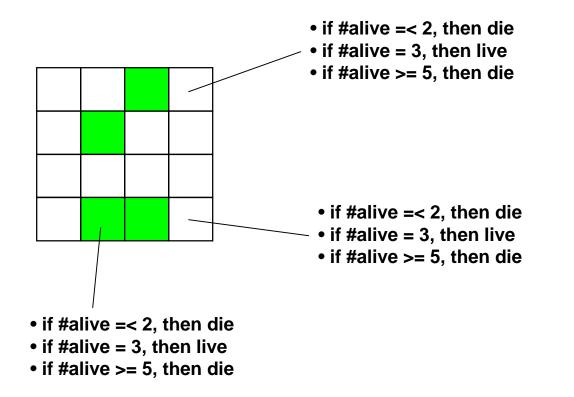
#### Cellular Automata - Array

"A CA is an array of identically programmed automata, or cells, which interact with one another in a neighborhood and have definite state"



#### Cellular Automata - Cells

"A CA is an array of dentically programmed automate, or cells, which interact with one another in a neighborhood and have definite state"



#### Cellular Automata - Interaction

"A CA is an array of identically programmed automata, or cells, which interact with one another in a neighborhood and have definite state"

#### the rules

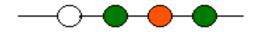
if #alive =< 2, then die if #alive = 3, then live if #alive >= 5, then die Discuss the role of local interaction in modern VLSI and future (nano) technologies

#### Cellular Automata - Neighbourhood

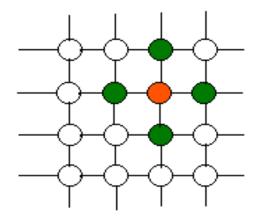
• Classic examples of cell neighborhoods:

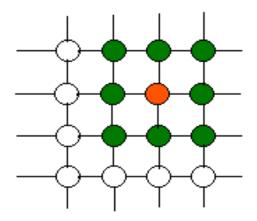


Moore Neighborhood



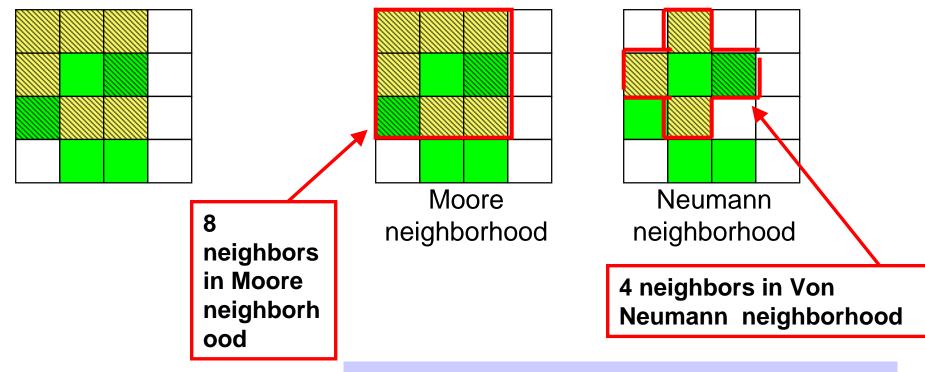






#### Cellular Automata - Neighbourhood

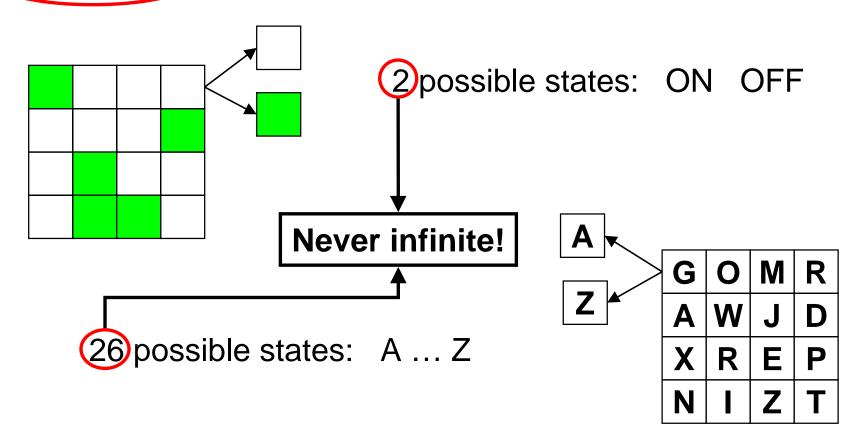
"A CA is an array of identically programmed automata, or cells, which interact with one another in a neighborhood and have definite state"



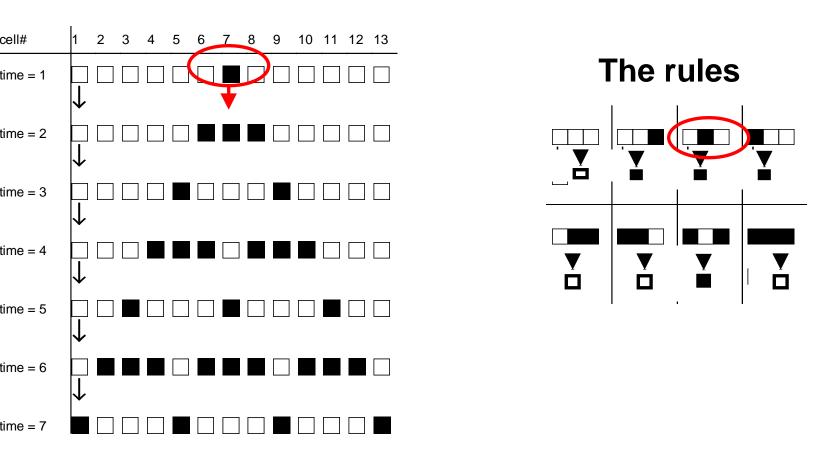
Margolus, Wolfram and other neighborhoods

#### Cellular Automata - States

"A CA is an array of identically programmed automata, or cells, which interact with one another in a neighborhood and have definite state"

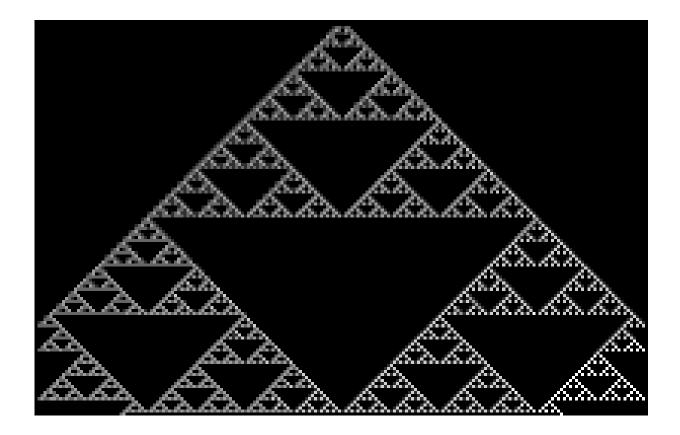


#### Cellular Automata - Simple 1D Example



Observe the recursive property of the pattern

#### Cellular Automata - Pascal's Triangle

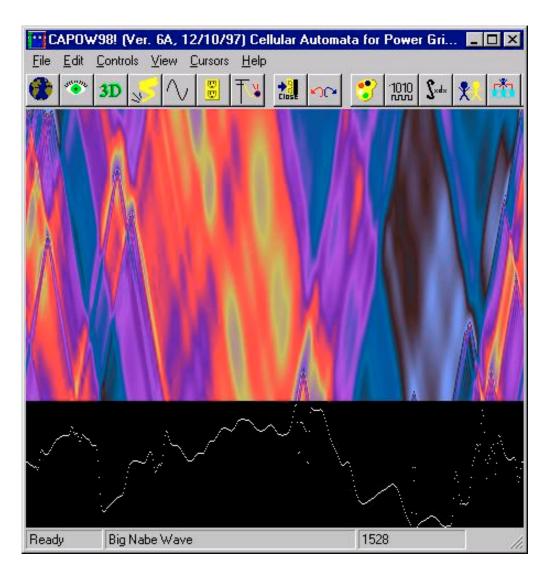


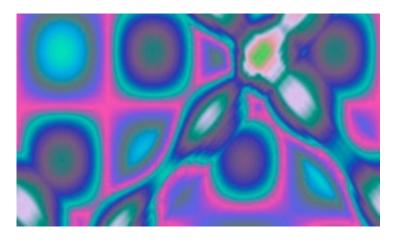
• *dimension* 1D, 2D ... nD

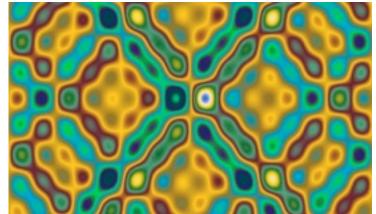
neighborhood Neumann, Moore for 1D
 (2D => r is used to denote the radius)

• number of states 1,2,..., n

#### Cellular Automata - Wow! examples







... a little bit of formalism....

### **Automata Theory**

Automata Theory is a branch of Computer Science which:

a) Attempts to answer questions like "What can computers do, and what is beyond their capabilities?"

b) Helps create and study new models of computation in a clear, unambiguous way.

c) Contrary to popular belief, has very practical implications and is the basis for many real-world applications

### **Cellular Automata Formalism**

- An important component of a Cellular Automaton is its interconnection graph, Γ, which is, typically, an ndimensional grid.
- Each cell of the CA can be in one of several possible states. The state set, Q, of a Cellular Automaton is the set of all possible states that a cell can be in.
- The pair (Γ, Q) is usually referred to as a Cell Space of the CA.

### **Cellular Automata Formalism**

• A configuration, x, of a CA is a mapping from the graph to the state set, which assigns a state from the state set Q to each node in the graph  $\Gamma$ , i.e.

x: 
$$\Gamma \rightarrow Q$$
  
x(i) = q, where i∈ Γ and q∈ Q

• A configuration of a CA describes the overall state of the Cellular Automaton on a global scale

### **Cellular Automata Formalism**

- The computation of CAs, though, is a local process. The next state of each cell depends on its current state, and the states of its closest neighbors only.
- Thus, we need to define the concept of a cell neighborhood.
- A neighborhood of a cell in a cellular automaton, is the collection of cells situated at a "distance" r or less from the cell in question.

# Cellular Automata Formalism: local dynamics

- Each cell of a CA is a simple Finite State Machine
- The local dynamics (transition function) of a cell, denoted  $\delta$ , is a function, which receives as inputs the state of a cell and its "neighbors", and computes the next state of the cell.
- For example, the local dynamics of a 1-D CA can be defined as follows:

$$\delta(\mathbf{x}_{i-1},\mathbf{x}_i,\mathbf{x}_{i+1}) = \mathbf{x}_i'$$

• The local dynamics is often expressed as a table:

## Cellular Automata Formalism: definition

- Formally, a Cellular Automaton is a quadruple  $M = (\Gamma, Q, N, \delta)$ , where:
  - $\Gamma$  interconnection graph,
  - Q set of states
  - N neighborhood (e.g. von Neumann, etc.)
  - $\delta$  local dynamics

# Cellular Automata Formalism: global dynamics

- The local dynamics,  $\delta$ , of a CA describes the computation occurring locally at each cell.
- The global computation of the CA as a system is captured by the notion of global dynamics.
- The global dynamics, T, of a CA is a mapping from the set of configurations C to itself, i.e.

 $T: C \to C$ 

• Thus, the global dynamics describes how the overall state of the CA changes from one instance to the next

# Cellular Automata: link to dynamical systems

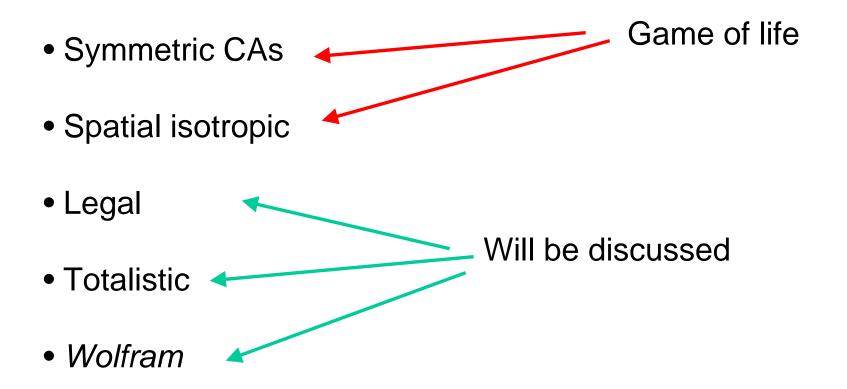
• Since the global computation is determined by the computation of each individual cell, the global dynamics, T, is defined in terms of the local dynamics,  $\delta$ :

$$T(x)_i := \delta(x_{i-1}, x_i, x_{i+1})$$

• Starting with some initial configuration, x, the Cellular Automaton evolves in time by computing the successive iterations of the global dynamics:

• Thus, we can view the evolution of a CA with time as a computation of the forward orbit of a discrete dynamical system.

# Cellular Automata - Types



# **Cellular Automata - Wolfram**

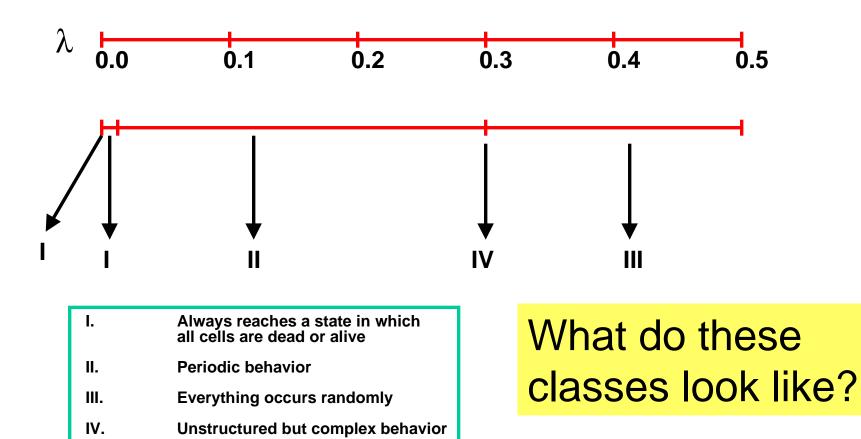
What are the possible "behaviors" of "black patterns"?

- Always reaches a state in which all cells are dead or alive
- II. Periodic behavior
- III. Everything occurs randomly
- IV. Unstructured but complex behavior

#### Cellular Automata - Wolfram

Wolfram introduced a parameter called lambda

 $\lambda$  = chance that a cell is alive in the next state



- What is the total number of possibilities with CAs?
- Let's look at total number of possible rules
- For 1D CA:
  - $2^3 = 8$  possible "neighborhoods" (for 3 cells)  $2^8 = 256$  possible rules
- For 2D CA:

 $2^9 = 512$  possible "neighborhoods"  $2^{512}$  possible rules (!!)

#### <u>Cellular Automata - Alive or not?</u>

• Can CA or Game of Life represent life as we know it?

• A computer can be simulated in Life

• Building blocks of computer (wires, gates, registers) can be simulated in Life as patterns (gliders, eaters etcetera)

• Possible to build a computer, possible to build life?

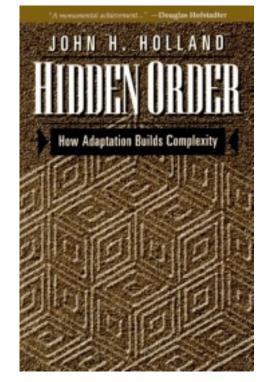
#### Universal Machines - Cellular Automata



Stanislaw Ulam (1909 - 1984)

### Universal Machines - Cellular Automata

- conceived in the 1940s
- Stanislaw Ulam evolution of graphic constructions generated by simple rules
- Ulam asked two questions:
  - can recursive mechanisms explain the complexity of the real?
  - Is complexity then only appearant, the rules themselves being simple?



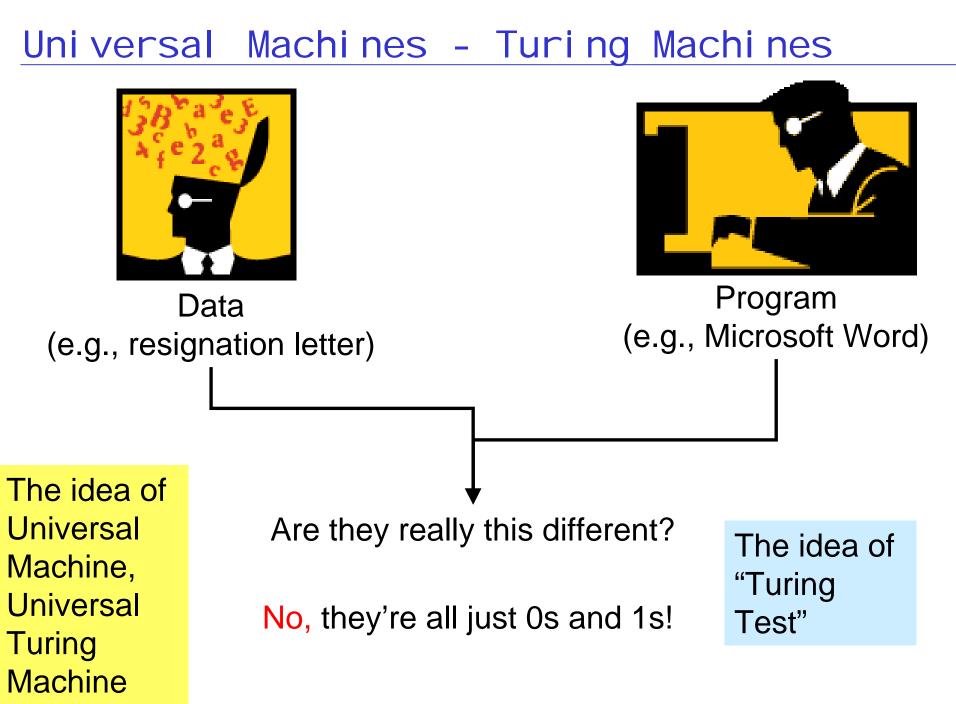
Stanislaw Ulam Memorial Lectures

#### Universal Machines - Turing Machines





#### Alan Turing (1912-1954)



#### Universal Machines - Neumann Machines

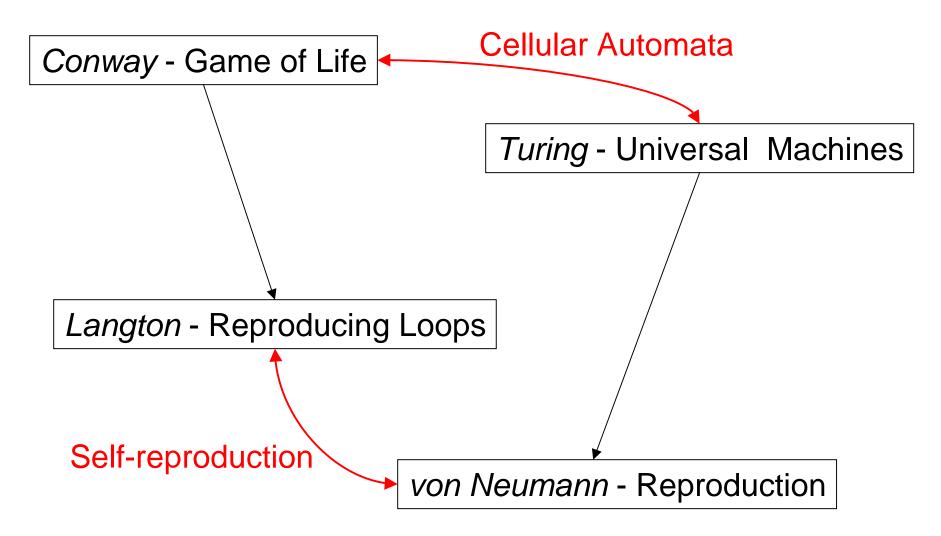


John von Neumann (1903- 1957)

- John von Neumann interests himself on theory of self-reproductive automata
- worked on a self-reproducing "kinematon" (like the monolith in "2001 Space Odyssey")
- Ulam suggested von Neumann to use "cellular spaces"
- extremely simplified universe



#### Self Reproduction

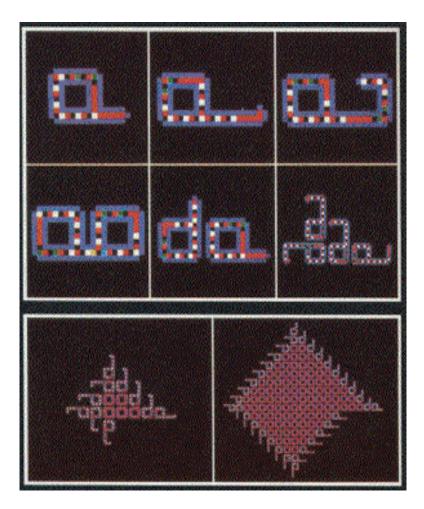


### Self Reproduction

	2	2	2	2	2	2	2	2						
2	1	7	0	t	4	0	1	4	2					
2	0	2	2	2	2	<b>2</b>	2	0	2					
2	7	2					2	1	2					
2	1	2					2	1	2					
2	0	2					2	1	2					
2	7	2					2	I	<b>2</b>					
2	1	2	<b>2</b>	$\cdot 2$	<b>2</b>	<b>2</b>	<b>2</b>	1	<b>2</b>	2	2	<b>2</b>	2	
2	0	7	1	0	7	1	0	7		1	1	1	1	
	2	2	2	<b>2</b>	2	2	2	2	<b>2</b>	2	2	2	2	

### Langton Loop's

- 8 states
- 29 rules



# Cellular Automata as Dynamical Systems

### **Chaos Theory**

# Chaotic Behavior of Dynamical Systems

• A Discrete Dynamical System is an iterated function over some domain, i.e.

 $F: D \rightarrow D$ 

• Example 1: F(x) = x

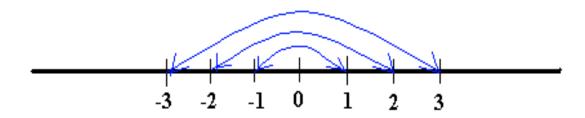
x=0, F(0) = 0,  $F(F(0)) = F^2(0) = 0$ , ...,  $F^n(0) = 0$ , ...

x=3, F(3) = 3,  $F(F(3)) = F^2(3) = 3$ , ...,  $F^n(3) = 3$ , ...

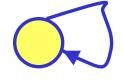
x=-5, F(-5) = -5,  $F(F(-5)) = F^2(-5) = -5$ , ...,  $F^n(-5) = -5$ , ...

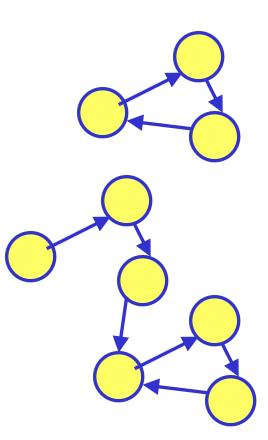
• Example 2: F(x) = -x

x=0, F(0) = 0,  $F(F(0)) = F^2(0) = 0$ , ...,  $F^n(0) = 0$ , ... x=3, F(3) = -3,  $F(F(3)) = F^2(3) = 3$ , ...,  $F^n(3) = 3$ ,  $F^{n+1}(3) = -3$ , ... x=-5, F(-5) = 5,  $F(F(-5)) = F^2(-5) = -5$ , ...,  $F^n(-5) = -5$ ,  $F^{n+1}(-5) = 5$ , ...

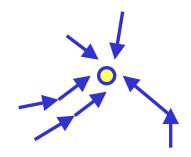


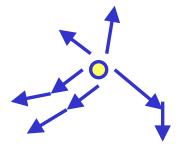
- A point, x, in the domain of a dynamical system, F, is a fixed point iff F(x) = x
- A point, x, in the domain of a dynamical system, F, is a periodic point iff F<sup>n</sup>(x) = x
- A point, x, in the domain of a dynamical system, F, is eventually periodic if F<sup>m+n</sup>(x)=F<sup>m</sup>(x)





- Sometimes certain points in the domain of some dynamical systems exhibit very <u>interesting properties</u>:
  - A point, x, in the domain of F is called an attractor iff there is a neighborhood of x such that any point in that neighborhood, under iteration of F, tends to approach x
  - A point, x, in the domain of F is called a repeller iff there is a neighborhood of x such that any point in that neighborhood, under iteration of F, tends to diverge from x





• Our goals, when studying a dynamical system are:

a) To predict the long-term, asymptotic behavior of the system given some initial point, x, and

b) To **identify interesting points** in the domain of the system, such as attractors, repellers, periodic points, etc.

- For some simple dynamical systems, predicting the long-term, asymptotic behavior is fairly simple (recall examples 1 and 2)
- For other systems, one cannot predict more than just a few iterations into the future.
  - Such unpredictable systems are usually called chaotic.

### **Chaotic Dynamics**

• A chaotic dynamical system has 3 distinguishing characteristics:

a) **Topological Transitivity** - this implies that the system cannot be decomposed and studied piece-by-piece

b) **Sensitive Dependence on Initial Conditions** - this implies that numerical simulations are useless, since small errors get magnified under iteration, and soon the orbit we are computing looks nothing like the real orbit of the system

c) The set of periodic points is dense in the domain of the system - amidst unpredictability, there is an element of regularity

# Cellular Automata as Dynamical Systems

- As we saw earlier, the behavior of a Cellular Automaton in terms of iterating its **global dynamics**, **T**, can be considered a dynamical system.
- Depending on the initial configuration and the choice of **local dynamics**,  $\delta$ , the CA can exhibit any kind of behavior typical for a dynamical system fixed, periodic, or even chaotic
- Since CAs can accurately model numerous real-world phenomena and systems, understanding the behavior of Cellular Automata will lead to a better understanding of the world around us!

# Summary

- Ulam
- Turing
- von Neumann
- Conway
- Langton

- Universal Machines
- Turing Machines
- von Neumann Machines
- Game of Life
- Self Reproduction

### New Research and Interesting Examples

- Image Processing shifter example from Friday's Meetings
- GAPP Geometric Array
  Processor of Martin
  Marietta used in
  (in)famous Patriot Missiles
- Cube Calculus Machine a controlled one dimensional Cellular Automaton to operate on Multiple-Valued Functions

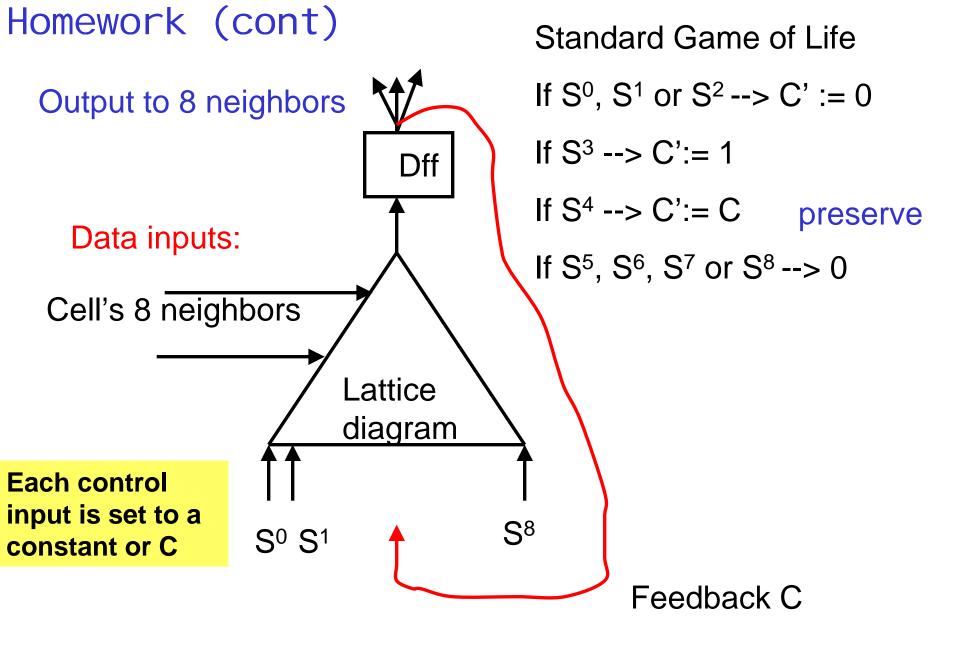
- Applications in Physics
- •CAM 8 Machine of Margolus
- CBM machine of Korkin and Hugo De Garis
- Applications in biology, psychology, models of societies, religions, species domination, World Models.
- Self Reproduction for future Nano-technologies

# Homework

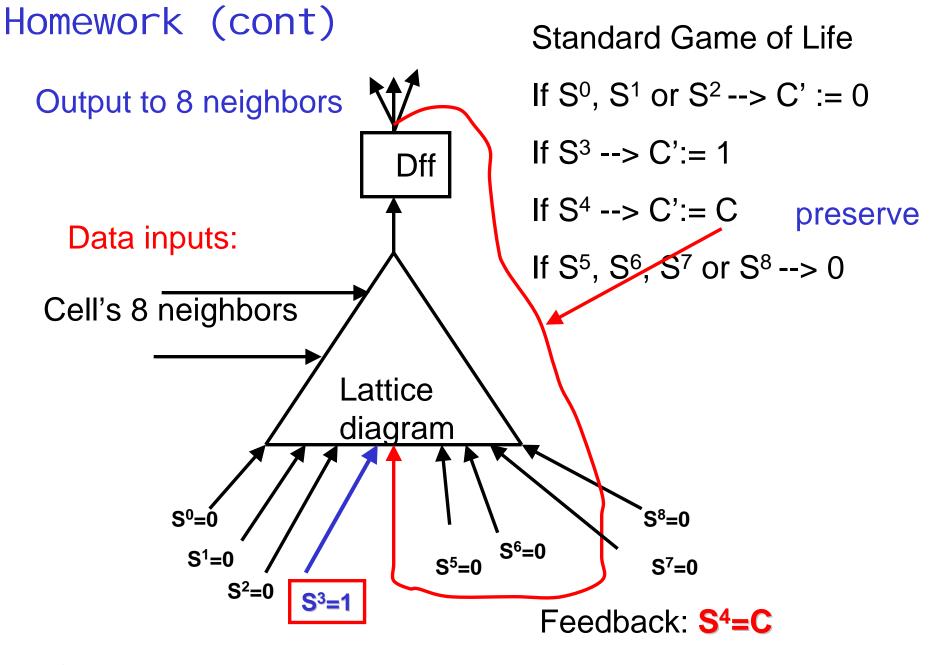
- This is a programming and presentation homework, at least two weeks are given.
- Your task is to simulate and visualize an emergent "generalized game of life".
- How many "interesting Games of Life" exists? Try to find the best ones.
- Use a generalized symmetric function in which every symmetry coefficient is 0, 1 or output from flip-flop of Cell' State C. Thus we have 8 positions, each in 3 states, and there is 3<sup>8</sup> possible ways to program the Game of Life.
- The standard Game of Life is just one of that many Games of Life. Most of these all "universes" are perhaps boring. But at least one of them is an universal model of computation? What about the others?
- Your task is to create a programming and visualization environment in which you will investigate various games of life. First set the parameters to standard values and observe gliders, ships, ponds, eaters and all other known forms of life.
- Next change randomly parameters, set different initial states and see what happens.
- Define some function on several generations of life which you will call "Interestingness of Life" and which will reflect how interesting is given life model for you, of course, much action is more interesting than no action, but what else?
- Finally create some meta-mechanism (like God of this Universe) which will create new forms of life by selecting new values of all the parameters. You can use neural net, genetic algorithm, depth first search, A\* search, whatever you want.

# Homework(cont)

- Finally create some meta-mechanism (like God of this Universe) which will create new forms of life by selecting new values of all the parameters. You can use neural net, genetic algorithm, depth first search, A\* search, whatever you want.
- Use this mechanism in feedback to select the most interesting Game of life. Record the results, discuss your findings in writing.
- Present a Power Point Presentation in class and show demo of your program.
- You should have some mechanism to record interesting events. Store also the most interesting parameters and initial states of your universes.
- Possibly we will write a paper about this, and we will be doing further modifications to the Evolutionary Cellular Automaton Model of Game of Life.



Control (program) inputs (register)



Control (program) inputs (register)