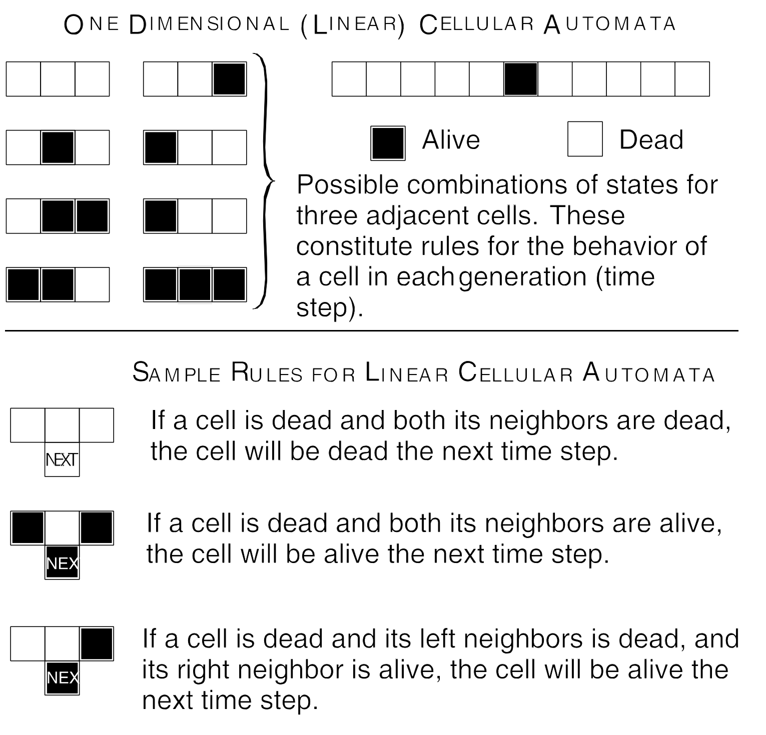
CELLULAR AUTOMATA[1]
How They Are Created and Work

WHAT ARE CELLULAR AUTOMATA?

Cellular Automata (CA) are simply grids of cells, where the individual cells change states according to a set of rules. The CA may be one dimensional, or linear, like a string of cells in a row (below), or two dimensional, like a checkerboard (next page).

Cells in a CA can exist in any states you assign them. For example, each cell may have two states: "alive" (shaded, or lit up on a computer) or "dead" (unshaded, or unlit). Or each cell may have five states: red, green, yellow, blue, or white, or any other states you wish.

A ONE DIMENSIONAL (LINEAR) CA



Each cell is also able to change state from one generation (one time step, or one iteration) to the next generation (time step, or iteration) following a set of rules. For a one dimensional system there are 8 possible rules of change, 2 states for each cell, for each of three adjacent cells ($2^3 = 8$). Only a few selected rules are operating at any one time in a CA.

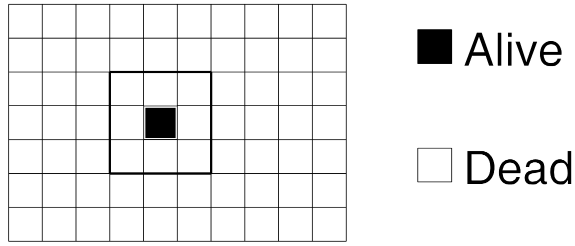
In linear CA the first generation is the first row, and each following generat-ion adds another row to show the sequence of changing states. Examples of three of the eight rules (above) show the original state of the center cell (top row) and what happens to that cell in the next generation (second row, with only the changed central cell shown) based on the given rule.

CELLULAR AUTOMATA are the working out of these fixed rules over a series of generations.

---

[1] Extensive explorations of cellular automata (including forms more sophisticated than we are studying) can be found in Stephen Levy, 1992, Artificial Life: Pantheon Books, and Stephen Prata, 1993, Artificial Life Playhouse: Waite Group Press (complete with computer disk with artificial life programs, including Life3000). *[WP\brain\alife\cellauto*.97]

TWO DIMENSIONAL CELLULAR AUTOMATA



To the right is a two dimensional CA. Note that there is a live cell in the center, surrounded by 8 other cells. Because in this case there are a total of nine cells, each of which can assume one of two states (alive or dead) there are $2^9 = 512$ possible rules. These rules are usually divided into rules necessary for ***survival*** of a cell, and rules necessary for ***birth*** of a cell. As with linear CA, only a few rules are operating at any one time. Two common rules are:

- ☞ **A Survival Rule:** *A living cell with two or three neighboring cells survives during the next time step, those with fewer neighbors die of loneliness, and those with more die of overcrowding.*
- ☞ **A Birth Rule:** *The only way for a dead cell to come alive the next time step is for it to have exactly three living neighbors.*

Of course, the rules apply to every cell every generation, so every cell must be compared with every surrounding cell, and its state adjusted accordingly. Unlike a linear CA, to show the next generation of a two dimensional CA a new grid is necessary. With all these rules, and all these cells, calculating a CA can be a task of tedious detail. An simple example of how to calculate a two dimensional CA is shown on the last page.

Although there are 512 rules, some rules are very unproductive. For example, if all rules produce a dead cell, then no live cells are left after the first generation and the CA becomes a closed system with no information flowing. Alternatively, if a cell become a live cell for all possible states of its neighbors, then after one generation the entire system becomes a field of all live cells stimulated to be alive every additional generation. The system is open, with information flowing without constraint.

Also, it is typical for rules to be combined by being less specific about which cells must be alive or dead to cause birth or allow survival. This significantly reduces the total number of rules from 512 to only a few. For example, we could specify that a cell must have only the upper three cells alive before it becomes alive, or only the three right side cells alive, etc. Or, we could just say, like the rules above, ***any*** three adjacent cells of the 8 need be alive for a cell to come alive. In this case this one rule combines 56 rules, since there are 56 different ways you can arrange three live neighbors around a central cell.

In addition to altering the birth/death neighbor rules in CA, there are other variables that can be used to alter how the CA's develop. For example, a ***mutation*** control can be set to randomly bring cells to life, and at varying rates.

Also possible is ***worldwrap***. Normally the edge of the CA world is the edge of the grid of cells, and any activity coming up against that edge gets frozen in its last state. With worldwrap on, however, the left side "wraps around" to the right side, and the top "wraps around” to the bottom, so that any activity “falling off the left”, reappears on the right, and vice versa, and just keeps on going.

CLASSIFYING CELLULAR AUTOMATA RULES

Researchers have explored the ramifications of the CA and discovered some interesting properties about their behavior. Stephen Wolfram, studying one dimensional CA’s found he could group rule sets into four classes based on the patterns of living, surviving, and dying cells spreading across the grid (next page).

Chris Langton developed a quantitative scheme that assigns a numerical measure, called lambda (λ), to the behavior of CA under various rule sets. λ is calculated from the behavior of the CA, and varies smoothly through a range of values, as shown in the diagram on the next page.

λ measures the freedom with which information flows in the CA. If the CA rules make it very difficult for information to flow (Class One-Fixed; i.e. no change in cells), then cells become frozen in their present state and cannot change. If the rules allow information to flow without constraint (Class Three-Chaotic) the result is a CA which is so fluid that it shows no recognizable patterns. Rules in between (Two-Periodic, and Four-Complexity) allow information to flow, but with varying degrees of constraint. These produce interesting patterns.

THE EMERGENT PROPERTIES OF CELLULAR AUTOMATA

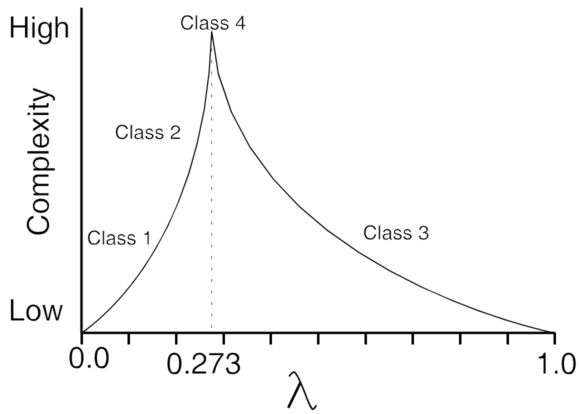
Intuitively, it might seem that nothing very interesting would happen in these relatively simple systems. Because the rules are fixed, the system is "deterministic." That is, the outcome of each run of iterations, for each set of rules, is fully determinable from the rules. These systems are creatures of pure logic.

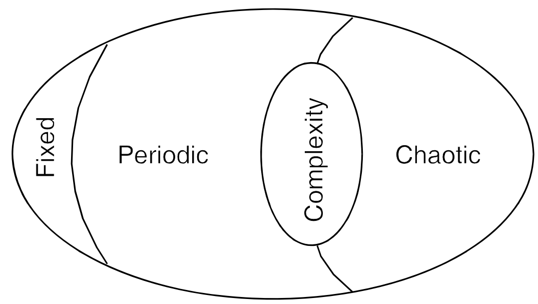
But it turns out to be not that simple. CA’s can produce a rich world of unexpected behavior with waves and patterns of blinking cells sweeping across the grid. With the right rules, cellular automata enter into the realm of complexity where the emergent properties of sensitive dependence and local rules/global behavior are abundant. That is, what emerges from the CA’s is, even though deterministic, unpredictable, unexpected, and rich with information and meaning.

CA’s not only exhibit the principles of chaos/complexity theory, they also fall within the realm of the ***Computational Viewpoint***; that is, to know a mathematical truth you must be able to compute it (for CA’s, a great chore before computers). Or, the outcome of an algorithm can only be known by calculating the algorithm (because of sensitive dependence). Or, you have to see it to believe it, since what emerges from these CA’s defies intuition.

Classifying Cellular Automata Behaviors

Chris Langton's Lambda λ Values





Stephen Wolfram's Classes

Class One - Fixed or Static:

- ☞ Rules that produce dull universes, such as all dead cells, or all living cells, or mixed living and dead cells which do not change; e.g. a solid.
- ☞ Information stops flowing producing a closed system.
- ☞ Corresponds to a fixed attractor in chaos/complexity theory.

Class Two - Periodic or Oscillatory:

- ☞ Rules that produce stable, repetitive configurations; e.g. a pendulum.
- ☞ Information flows weakly.
- ☞ Corresponds to a periodic attractor in chaos/complexity theory.

Class Three - Chaotic:

- ☞ Rules that produce chaotic (random, non repeating) patterns; e.g. molecules in a gas.
- ☞ Information flows without constraint.
- ☞ Corresponds to a strange attractor in chaos/complexity theory.

Class Four - Complexity:

- ☞ Rules that produce complex, locally organized patterns; e.g. like turbulent liquid.
- ☞ Information flows fluidly, but not unconstrained, easily producing complex patterns.
- ☞ Corresponds to a strange attractor in chaos/complexity theory.

Observationally, Information Flow Can Be Crudely Determined by the Following Behaviors

(Intermediate classifications are possible; for example 3 evolving to (⇒) 2, or a class between 1 and 2)

Class 3 - Information Flows Freely If:	Intermediate Information Flow ⇦ ⇦ Classes 2 & 4 ⇨ ⇨	Class 1 - Information Is Retained If:
1. Cell behavior is chaotic 2. Patterns never settle down. 3. Cells expand and contract rapidly.	⇦ ⇦ ⇨ ⇨	1. Cell behavior is static 2. Patterns easily settle down. 3. Cells expand and contract slowly, or are static.