A cellular automaton is a discrete model studied in computability theory, mathematics, physics, complexity science, theoretical biology and microstructure modeling. Cellular automata are also called cellular spaces, tessellation automata, homogeneous structures, cellular structures, tessellation structures, and iterative arrays.\[2\]

A cellular automaton consists of a regular grid of cells, each in one of a finite number of states, such as on and off (in contrast to a coupled map lattice). The grid can be in any finite number of dimensions. For each cell, a set of cells called its neighborhood is defined relative to the specified cell. An initial state (time $t = 0$) is selected by assigning a state for each cell. A new generation is created (advancing $t$ by 1), according to some fixed rule (generally, a mathematical function) that determines the new state of each cell in terms of the current state of the cell and the states of the cells in its neighborhood. Typically, the rule for updating the state of cells is the same for each cell and does not change over time, and is applied to the whole grid simultaneously, though exceptions are known, such as the stochastic cellular automaton and asynchronous cellular automaton.

The concept was originally discovered in the 1940s by Stanislaw Ulam and John von Neumann while they were contemporaries at Los Alamos National Laboratory. While studied by some throughout the 1950s and 1960s, it was not until the 1970s and Conway's Game of Life, a two-dimensional cellular automaton, that interest in the subject expanded beyond academia. In the 1980s, Stephen Wolfram engaged in a systematic study of one-dimensional cellular automata, or what he calls elementary cellular automata; his research assistant Matthew Cook showed that one of these rules is Turing-complete. Wolfram published A New Kind of Science in 2002, claiming that cellular automata have applications in many fields of science. These include computer processors and cryptography.

The primary classifications of cellular automata, as outlined by Wolfram, are numbered one to four. They are, in order, automata in which patterns generally stabilize into homogeneity, automata in which patterns evolve into mostly stable or oscillating structures, automata in which patterns evolve in a seemingly chaotic fashion, and automata in which patterns become extremely complex and may last for a long time, with stable local structures. This last class are thought to be computationally universal, or capable of simulating a Turing machine. Special types of cellular automata are reversible, where only a single configuration leads directly to a subsequent one, and totalistic, in which the future value of individual cells depend on the total value of a group of neighboring cells. Cellular automata can simulate a variety of real-world systems, including biological and chemical ones.
The red cells are the Moore neighborhood for the blue cell.

The red cells are the von Neumann neighborhood for the blue cell. The extended neighborhood includes the pink cells as well.

One way to simulate a two-dimensional cellular automaton is with an infinite sheet of graph paper along with a set of rules for the cells to follow. Each square is called a "cell" and each cell has two possible states, black and white. The neighborhood of a cell is the nearby, usually adjacent, cells. The two most common types of neighborhoods are the von Neumann neighborhood and the Moore neighborhood. The former, named after the founding cellular automaton theorist, consists of the four orthogonally adjacent cells. The latter includes the von Neumann neighborhood as well as the four remaining cells surrounding the cell whose state is to be calculated. For such a cell and its Moore neighborhood, there are 512 (= 2^9) possible patterns. For each of the 512 possible patterns, the rule table would state whether the center cell will be black or white on the next time interval. Conway's Game of Life is a popular version of this model. Another common neighborhood type is the extended von Neumann neighborhood, which includes the two closest cells in each orthogonal direction, for a total of eight. The general equation for such a system of rules is $k^s$, where $k$ is the number of possible states for a cell, and $s$ is the number of neighboring cells (including the cell to be calculated itself) used to determine the cell's next state. Thus, in the two dimensional system with a Moore neighborhood, the total number of automata possible would be $2^{2^9}$, or $1.34 \times 10^{154}$.

It is usually assumed that every cell in the universe starts in the same state, except for a finite number of cells in other states; the assignment of state values is called a configuration. More generally, it is sometimes assumed that the universe starts out covered with a periodic pattern, and only a finite number of cells violate that pattern. The latter assumption is common in one-dimensional cellular automata.
Cellular automata are often simulated on a finite grid rather than an infinite one. In two dimensions, the universe would be a rectangle instead of an infinite plane. The obvious problem with finite grids is how to handle the cells on the edges. How they are handled will affect the values of all the cells in the grid. One possible method is to allow the values in those cells to remain constant. Another method is to define neighborhoods differently for these cells. One could say that they have fewer neighbors, but then one would also have to define new rules for the cells located on the edges. These cells are usually handled with a toroidal arrangement: when one goes off the top, one comes in at the corresponding position on the bottom, and when one goes off the left, one comes in on the right. (This essentially simulates an infinite periodic tiling, and in the field of partial differential equations is sometimes referred to as periodic boundary conditions.) This can be visualized as taping the left and right edges of the rectangle to form a tube, then taping the top and bottom edges of the tube to form a torus (doughnut shape). Universes of other dimensions are handled similarly. This solves boundary problems with neighborhoods, but another advantage is that it is easily programmable using modular arithmetic functions. For example, in a 1-dimensional cellular automaton like the examples below, the neighborhood of a cell $x^i$ is \{(x_{i-1}^{t-1}, x^i, x_{i+1}^{t-1})\}, where $t$ is the time step (vertical), and $i$ is the index (horizontal) in one generation.