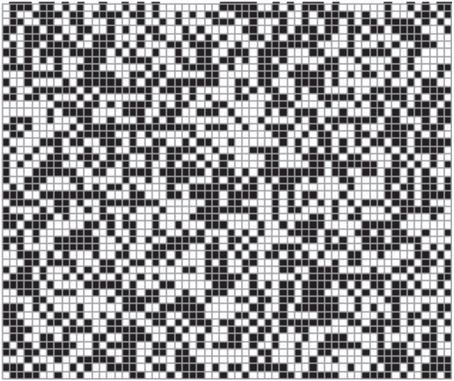
Cellular Automata

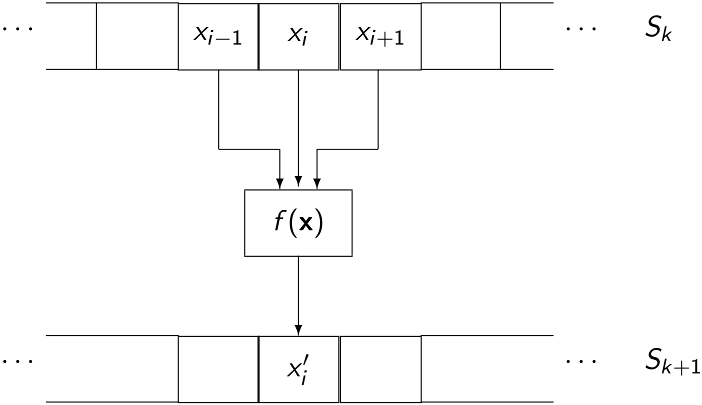


Cellular Automata

- A one-dimensional cellular automaton consists of a linearly connected array of n cells, each of which takes the value of 0 or 1, and a boolean function $f(\mathbf{x})$ with q variables
- The value of the cell x_i is updated in parallel (synchronously) using this function in discrete time steps as $x_i' = f(\mathbf{x})$ for $i = 1, 2, \ldots, n$
- The boundary conditions are usually handled by taking the index values modulo n, i.e., the linearly connected array is actually a circular register
- The parameter q is usually an odd integer, i.e., $q = 2r + 1$, where r is often named the radius of the function $f(\mathbf{x})$; the new value of the ith cell is calculated using the value of the ith cell itself and the values of r neighboring cells to the right and left of the ith cell

Cellular Automata

The one-dimensional cellular automaton with $q = 3$.



Cellular Automata

- Since there are n cells, each of which takes the values of 0 or 1, there are 2^n possible state vectors
- Let S_k denote the state vector at the automaton moves to the states S_1, S_2, S_3, etc., at time steps $k = 1, 2, 3$, etc
- The state vector S_k takes values from the set of n-bit binary vectors as k advances, and the state machine will eventually cycle, i.e., it will reach a state S_{k+w} which was visited earlier $S_k = S_{k+w}$
- The period w is a function of the initial state, the updating function, and the number of cells

CA30 – A Random Updating Function

- Cellular automata are generally considered as discrete dynamical systems, or discrete approximations to partial differential equations modeling a variety of natural systems
- Wolfram proposed a random sequence generator based on the one-dimensional cellular automaton with $q = 3$ and the so-called CA30 updating function

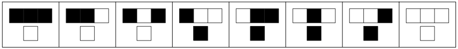
  $$f(x_{i-1}, x_i, x_{i+1}) = x_{i-1} \oplus (x_i + x_{i+1})$$

  where $+$ is the boolean OR and $\oplus$ is the exclusive OR function
- The state vectors produced by this cellular automaton seem to have randomness properties, e.g., the time sequence values of the central cell shows no statistical regularities under the usual randomness tests

CA30 – A Random Updating Function

- The truth table for the CA30 function $x_{i-1} \oplus (x_i + x_{i+1})$

x_{i-1}	x_i	x_{i+1}	x_i'
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

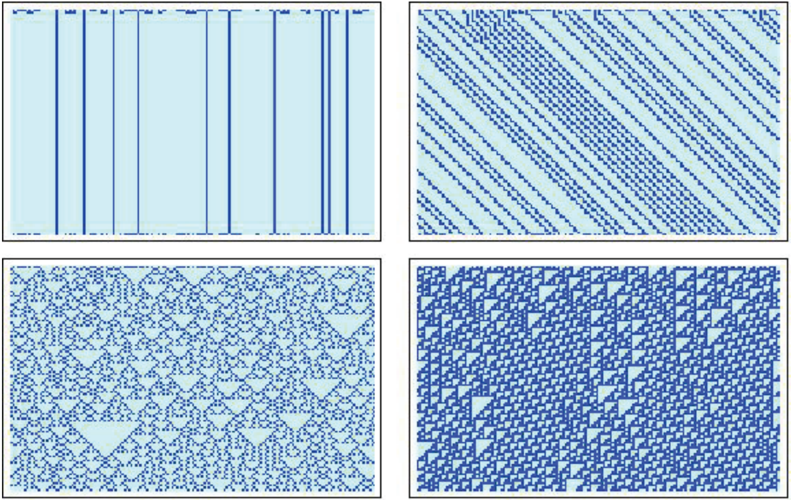


- The binary expansion of the integer $30 = (0001\ 1110)$
- For $q = 3$, we have $2^{2^3} = 256$ different CA functions: CA0 .. CA255

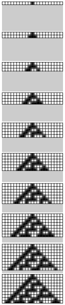
1-Dimensional CA Functions

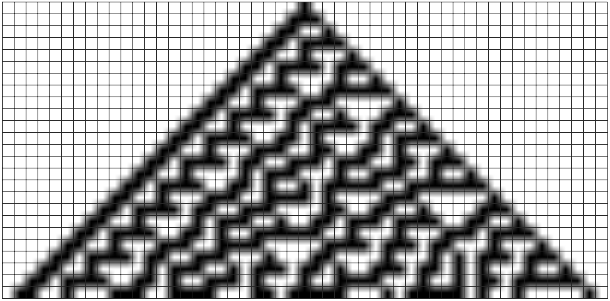
- Wolfram analyzed the behaviors of all them, starting from different initial conditions
- There seems to be 4 classes of behaviors
  - Class I Homogeneous: Everything eventually dies (or eventually lives forever). Some initial transient behavior usually precedes this final state
  - Class II Periodic: Perhaps after some initial transients, the pattern repeats itself exactly, in space (horizontally), in time (vertically), or both
  - Class III Chaotic: Patterns grow in a chaotic fashion: short-lived islands of order and sensitivity to initial conditions
  - Class IV Complex: Patterns grow in a complicated way, with both local stable behavior (acting as memory) and long-range correlations (acting to transmit data)

1-Dimensional CA Functions

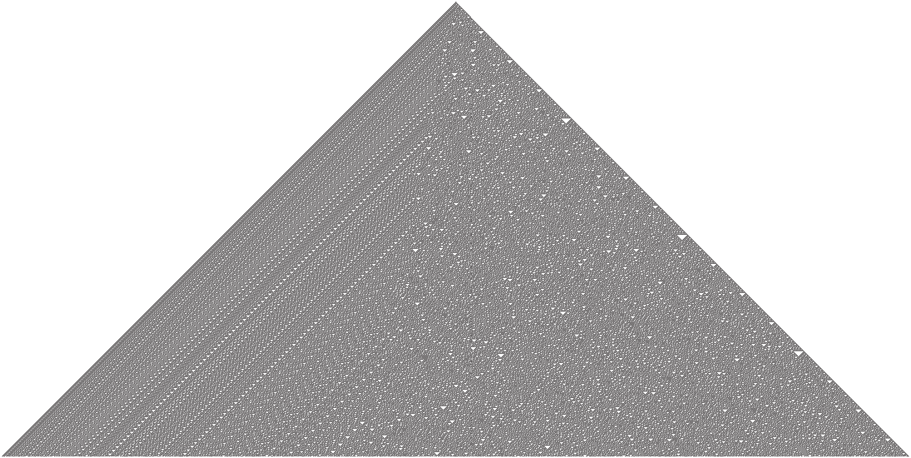


CA30 Behavior





CA30 Behavior



Security of CA-based RNG

- CA30 and many other CA functions satisfy R1
- Question: Do they satisfy R2?
- In order to use such a generator for cryptographic purposes, we must also ensure that the seed value (the initial state vector S_0) is difficult to construct given a sequence of state vectors
- It was claimed that this problem is in the class NP – no systematic algorithm for its solution for an "unbounded" n
- However, in order to use CA as a RNG in a stream cipher, we need select a suitable updating function and a large n (several hundred bits)
- We may not have cryptographic strength for some updating functions and for small values of n