

Lab 10: Cellular Automata

Part 1: Wolfram's elementary CAs

In Stephen Wolfram's rule numbering scheme for elementary CAs, each CA rule is defined by 8 bits corresponding to each of the 8 possible cell neighborhood configurations. In Wolfram's numbering scheme, the 8 bits are interpreted as a binary integer. Assume the following ordering of the 8 possible neighborhood configurations:

111 110 101 100 011 010 001 000

As an example, let the following be the corresponding output states for each of the above neighborhoods:

1 0 1 0 0 0 1 1

Interpreted as a binary integer, the string 10100011 is equal to 163 in decimal. So this is CA rule #163. Its λ value is the fraction of 1s in the update rule, which in this case is $4/8 = 0.5$. Note that in general there are only 9 possible λ values for elementary CAs: 0.0 (0/8), 0.125 (1/8), 0.25 (2/8), ..., 0.875 (7/8), and 1.0 (8/8).

Download the NetLogo file **elementaryCA.nlogo** (available on the class web page under Labs) to your Desktop, and then open it in NetLogo. For each λ value other than 0.0 or 1.0, choose two different elementary CAs with that λ value and write down its binary rule string and its rule number, as shown in the table below:

λ	CA rule	Rule number	Wolfram class
0.125	00100000	32	
0.125			
0.25			
0.25			
0.375			
0.375			
0.5			
0.5			
0.625			
0.625			
0.75			
0.75			
0.875			
0.875			

Then try out each CA on several random initial patterns and observe the resulting behavior, and decide which "Wolfram class" it is in (I, II, III, or IV). See pages 155-156 of *Complexity: A Guided Tour* for a more detailed description of these classes. Fill in the rest of the table for the other λ values. In your sample of elementary CAs, how well does the λ value of a CA correlate with its Wolfram class?

Part 2: One-dimensional CAs with more than two states

To try out some one-dimensional CAs that are more complex than Wolfram's ECAs, go to our class web page under Links and click on [Edge of Chaos CA](#). By default, this CA utilizes 4 states per cell, with a neighborhood size of 5 cells (in contrast, Wolfram's elementary CAs use only 2 states, with a neighborhood size of 3). Read through the instructions and familiarize yourself with the features of the program.

- You can generate a new CA rule table at random with the specified number of cell states and neighborhood size by clicking on the button *Create New Rule Set Using*. If you change the number of states or the neighborhood size, make sure to click this button again so the changes will take effect.
- The button *Create New World Using* generates a random initial configuration of cells, but does not change the CA rule itself. The *Fill* menu can be used to specify the type of initial configuration that is generated. After changing the *Fill* selection, click the button again for the change to take effect.
- Another way to generate a new CA rule is by changing the lambda (λ) slider control. This controls the fraction of non-dead states in the CA rule table.

Experiment with different lambda settings, numbers of states, neighborhood sizes, and initial configurations. Can you find a specific value of λ at which the CA goes from “fixed” to “periodic” behavior? How about from “periodic” to “chaotic”, or “predictable” to “unpredictable”? For each value of λ , try running the CA on several different random starting configurations in order to get a sense of the typical behavior generated by the CA. Can you locate a precise “edge of chaos” for your CA?

With some experimentation, it's possible to produce a truly amazing variety of images. A few of mine are included below for inspiration. Create some of your own, and screen-share them with the class!





