Instructor Version

Overview

Complex patterns often arise from repeatedly applying a simpler set of rules. In this activity, we explore a decentralized and spatially-extended computational model known as a cellular automaton. Then we explore the output of its computation when given a particular rule set and initial condition.

Work together as a group to complete each step of the activity. Write yours answers to the questions on the large sheet of paper, and be sure to write large enough so that other groups can see your answers.

Outline

Instructor introduces cellular automata as an nonstandard computational model useful for understanding the foundations of computation. One reason CAs are important is because their computational model is atypical—it is decentralized and spatially-extended. So CA are useful for developing an an understanding of more general notions of computation. Cellular automata rule 110 has been proven to be Turing complete. So in principle, it is as expressive and powerful as your laptop (but it clearly isn't as efficient).

A cellular automata is defined by a grid and a rule set. From a given initial condition, local rules are applied in parallel to each cell in current row. Applying the rules is tantamount to performing a computation. In this case, the output of that computation is a pattern.

Demonstrate to the whole class *how* to apply the rules using Rule 50. The output of this computation should be a triangle filled with a checkerboard pattern.

Now, let the students do a computation with Rule 90. The output of this computation is the Sierpenski triangle. This let's us talk about fractals and how/why they are important/appear in Nature. It is suggestive that many complex patterns we see might have been generated by recursively applying simple rules just as they did with the cellular automata. Show some pictures of Romanesco broccoli and other fun fractals. Google search for "mollusk shell cellular automata" and show them an example of how a stochastic variant of CAs can model the patterns on mollusk shells.

Activity

1. Grab a single sheet labeled "Rule 90". Tape it to a window or pin it to a board, so that all members in your group can see and draw on it.

2. Beginning with the first row, fill out the first 6 rows by applying the cellular automaton's rules. Make sure everyone understands how to apply the rules.

3. Using a "divide and conquer" approach, quickly fill out the rest of the grid. This means that every one in the group should pick up a marker and try to apply the rules at the same time.

   Q1: What feature of the rules makes it possible to cooperatively fill out the rest of the grid?

   A1: The rules are applied locally, and only require knowledge of the 3 cells directly above it. This means, that students can apply the rules in a parallel fashion. . . it is embarrassingly parallel.

   Q2: Could one person work on filling out the top half of the grid, while another person worked on filling out the bottom half? Why or why not?

   A2: No. The rules require the previous three cells. So you must always work down from the top.

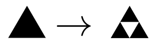
   Q3: Why is color of the middle square in each triplet not important? Find a simpler description of the rules that does not rely on the middle square. Describe it in words.

   A3: The rule set is symmetric. For example, 010 and 000 both map to the same output. The simpler rule set is just to look at the L and R cells above. An ideal description might be something like: The next square is black if just one of the L or R squares is black. It is an exclusive or (xor) function.

Bonus

1. Where does the "90" in "Rule 90" come from? Hint: $90 = 2^6 + 2^4 + 2^3 + 2^1$.

   Answer:
   90 is the decimal equivalent of the rule written as the binary number 01011010_2.
   Recall, the number 425 has a decimal expansion of $425 = 4 \cdot 10^2 + 2 \cdot 10^1 + 5 \cdot 10^0$.
   Similarly, $01011010_2 = 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$.

2. Consider the following rule: 

   You begin with an equilateral triangle, having side length equal to 1. Then, apply the rule recursively to each subsequent triangle. What is the output of this computation? What are the formulas for the perimeter and area of the pattern as a function of the number of iterations? What is the perimeter and area as the number of iterations tends to infinity?

   Answer: The output of the computation is also an approximation of the Sierpenski triangle.

$$P(n) = 3\left(\frac{3}{2}\right)^n \qquad \lim_{n\to\infty} P(n) = \infty$$

$$A(n) = \frac{\sqrt{3}}{4}\left(\frac{3}{4}\right)^n \qquad \lim_{n\to\infty} A(n) = 0$$

Resources

If you want to learn more, try searching for some of the following keywords:

> fractals, cellular automata, iterated function systems, chaos game,
> Sierpinski triangle, Turing complete, game of life, Mandelbrot set

Here are some fun webpages and a nice YouTube video:

> http://ecademy.agnesscott.edu/~lriddle/ifs/ifs.htm
> http://www.kevs3d.co.uk/dev/lsystems
> https://www.youtube.com/watch?v=5plLxMnbtAw