Uses of Classes

Do the classes group CA by their utility?
The Following Are Gross Generalizations!

- Don’t take these ideas of utility too literally!
  - They give a general idea of which class might have a CA that might solve a particular task.
  - By no means the final word!
Utility of Class I

- These CA go to simple fixed points.
  - Like all 0.

- Ok, so not very interesting.

- Don’t expect these to be used for much.
  - Might describe systems that are heavily damped.
    - E.g., spring with a heavy weight moving through viscous fluid.
Utility of Class II: Filters

- These CA take an initial state and weed out certain features.
  - Retains other features which then repeat.

- That’s what filters do!

- Think of class II as filters.
  - Filters are used to
    - Find details of images, like edges.
    - Weed out noise (e.g., “purify” electrical signals).
    - Remove viruses from incoming email.
    - Eliminate undesirable web sites.
    - Etc.
Class II Filter

- For example, we can build a CA that eventually filters out everything except an isolated 1.
  - 00100101110111101100 would filter to
  - 001001000000000000000000 would filter to
  - 0010010000000000000000000000000000 would filter to
  - Etc.

- Rule:
  - 111 110 101 100 011 010 001 000
  - 0 0 0 0 0 0 1 0

- What rule is that? Rule 4!
  - Rule 132 (10000100) would act similarly.
Class II: Smooth Filter

- A classic image processing filter “smoothes out” noise.
  - Sometimes called “mean”, “uniform”, or “smoothing” filter.
  - Treats 2-d image as array of numbers.
    - Image may have 256 colors... i.e., 256 states.
    - Image may have 2 colors... i.e., 2 states.
  - Replace each element of the array by the average of it’s neighbors.
    - Hey, that’s a totalistic CA!
  - What is the rule?
    - Just add neighbors (and self) and divide by 9. Round to the nearest allowable state.
The filter can be visualized as an array.

\[
\text{mean filter} = \frac{1}{9} \begin{pmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{pmatrix}
\]

- This array of all 1’s is the CA neighborhood.
- It shows which neighbors are being added together.
- Divide by 9 because want the average.
- Then take the nearest integer value
Class II: Noise Filter

- Similar to “smooth” filter.
  - Replace each cell’s state with the state of the majority of cells in the neighborhood.
    - So if have a single noisy white pixel in a sea of solid blue, then this replaces that cell with the majority color of blue.

- Wait, that’s just the rule “majority wins”!

- Try it!
  - Use Cellular Automaton Explorer.
    - Create sea of one color. Add a little “salt and pepper” noise by changing a few pixels here and there. Run “majority wins” for one time step.
    - Now import a picture (recommend 2 states so black and white, but can also use many states – like 256). Change a few pixels in a uniform region. Run the filter.
Class II: Edge Filter

- Another classic image processing filter finds edges.
  - The following highlights any vertical edges in the image.

\[
\begin{pmatrix}
1 & 0 & -1 \\
1 & 0 & -1 \\
1 & 0 & -1 \\
\end{pmatrix}
\]

- Do term by term multiplication and add all the results together.
  - \[\text{neighbor[0]} \ast 1 + \text{neighbor[1]} \ast 0 + \text{neighbor[2]} \ast -1 + \text{neighbor[3]} \ast -1 + \text{neighbor[4]} \ast -1 + \text{neighbor[5]} \ast 0 + \text{neighbor[6]} \ast 1 + \text{neighbor[7]} \ast 1\]

- Take absolute value and round to nearest permissible value.

- Can you see what happens if an image is all white?
- Or if left half is black and right half is white?

- Could you write a CA rule that does this?
  - Add only those neighbors as shown above.
Class II: Another Edge Filter

- Works in all directions.
  - Cell becomes 0 if the difference between it’s value and its neighbor’s values is less than a threshold $\varepsilon$ (for all of the neighbors).
  - Otherwise, cell keeps its current value.
  - i.e.

$$cell_{new} = 0, \text{ if } |cell_{currentValue} - neighbor[i]| < \varepsilon, \forall \text{ neighbors } i$$

$$cell_{new} = cell_{currentValue}, \text{ otherwise}$$

- Can you see what happens if an image is all white?
- Or if left half is black and right half is white?
Class II: Periodic Phenomena

- Class II can also be used to model periodic systems.
  - Waves
    - Ocean
    - Light
    - Audio
  - Springs
  - Zebra stripes
  - “The wave” at a football stadium.
  - Etc.

Assumes perfect periodicity. If there are variations, then must be class III.
Your Turn: Horizontal Edge Filter

- Write a Class II CA rule that performs as a horizontal edge filter.

- Use CA Explorer.
  - Remember how we wrote rules before?
  - Copy rule102.java.
  - Change name and alter the class.

- See the parameter “int[] neighbors”?
  - In 2-d neighbor[0] is to the upper left (northwest).
  - Continues clockwise.
  - So neighbor[1] is due north.
Your Turn: Restrict the Lattice

- You may want to restrict to square 8-neighbor lattices.
  - Prevent problems with the 1-d lattice.

- Override the getCompatibleLattices() method

  ```java
  public String[] getCompatibleLattices()
  {
      String[] lattices = {SquareLattice.DISPLAY_NAME};
      return lattices;
  }
  ```

- Don’t forget to change the tooltip and display name.
Utility of Class III: Physics!

- Most physical (and social) systems involve structures more complicated than periodic.
  - Sometimes randomness is even key.
  - So physics won’t often be Class I or II.

- Describing long term behavior of Class III CA difficult because final state depends on ever increasing number of initial states.
  - So statistics often best way to describe.
  - E.g., “Most likely outcome is…”

- So class III useful for problems that are amenable to statistical mechanics, randomness, or complicated geometric structures.
Utility of Class III: Examples

- Statistical mechanics problems include
  - Magnetic behavior.
  - Fluid flow.
  - Group behavior.
  - Bird flocks.

- Problems with “randomness” include
  - Rainfall patterns.
  - Percolation.

- Problems with complicated geometries
  - Seashell designs.
  - Non-trivial zebra stripes.
  - Phase changes – fractal structures emerge.

- Stay tuned. We will focus on these applications later in the semester.
Utility of Class IV: Everything And All of the Above!

- Ultimate computers.
  - We keep saying Class IV is “universal”.
  - So capable of computing/simulating anything
    - simulates any algorithm
  - So in particular can simulate any CAs from class I, II, or III.
    - Whoa! Is that cool or what.
Utility of Class IV: On the Edge

- Also, has obvious applications to anything that is not ordered but not chaotic either.
  - E.g., many believe that “thinking” and “life-like” simulations fall into this class.
    - E.g., can use class IV CA to specify successful foraging strategies in artificial life simulations.
  - Phase changes.
    - Yeah, I know. Also listed in class III. Depends on what you are studying.

Philosophical Implications

- Halting versus not halting.
- Limits of computability.
- Nature of the universe. See next slide!
Class IV: Universe?

- An argument presented by Wolfram.
  - Class IV CA are “computers” that act on their “programs”.
  - Initial configuration is the program.
  - So suppose have infinite random initial configuration.
  - All sequences of possible values are somewhere on this string.
    - Albeit with infinitesimally small probabilities.
  - So all possible programs are on this infinite string.
  - So CA will run all possible programs in parallel (simultaneously).
  - I.e., anything and everything in the universe will be simulated by a class IV CA with infinite random initial state.
  - Including life! See next slide.
Class IV: Life

- In particular, the initial state will have programs that self-replicate and proliferate.
  - Recall first day of class?
  - Von Neumann was using CA to look for self-replicating structures.
    - He designed one, and so have many others.
  - In fact, the initial state will even have a program that simulates a genetic algorithm.

- If program self-replicates, then it will eventually dominate the CA’s state.

- So self-reproducing (life-like) structures may have infinitely small probabilities in the beginning, but they will dominate the CA in the end!
  - “Life” is inevitable.
  - Again, can you say “cool”?