1 Introduction of the Ulam-Warburton Automaton

This project will allow you to explore the fun world of automatons and prove some interesting things. Suggested tasks are in the gray boxes. Since the assignment is an open-ended exploration completed by working closely with your instructors, you may end up going down other avenues than just the gray box concepts. It is okay if you don't complete many of these parts. This is a hard task!

> **Task: Read and inform**
> Please read this entire document before starting. Let your instructor know if you are interested in completing this project instead of the regular assignment. I, Marcus, am happy to give guidance along the way, so please email me.

Cellular automata are discrete models (there are some continuous varieties too, but we'll disregard them today) used to mimic processes and computation. They relate to a diverse set of things, including disease/information spread, seashell patterns, and the difficulty of computation. A cellular automaton is made up of a grid of cells and a fixed rule. Each cell has a set of states it can be in, today we will start by considering the states as *on* and *off*. While the grid can be in any finite number of dimensions, we will stick to two dimensions so we can show it like an image. We start with all the cells in some initial state and then advance to the next generation by applying the rule to each cell simultaneously. (There are many ways to make an automaton. Some varieties allow for asynchronous or stochastic rule application.)

> **Task: Background inquiry**
> I would suggest searching and reading a few articles on cellular automata. There is quite a variety, so it will help you to understand the landscape and how the model we will discuss compares. It will also help you generate ideas for later parts of this exercise.

The rule defines the behavior of the automaton as generations progress. We will start by considering the rule where a cell turns on only if exactly one of its neighbors, the cells directly adjacent to it on the north, south, east, or west, is on. Cells will never turn off in our first rule set. This rule defines what is called the Ulam-Warburton automaton (UWA) shown in Figure 1. UWA is named after the discoverers, Polish-American mathematician Stanislaw Ulam and Scottish engineer and amateur mathematician Mike Warburton. Ulam was influential in much of the early automata research (as well as nuclear physics and the atomic bomb). He actually was a professor and chair of the Mathematics Department here in CU Boulder from 1967 to 1975. (I would be curious to know if there is anything in honor of him on campus.)

> **Task: Implement UWA**
> Write code in your favorite programming language to visualize the UWA. You should be able to input any arbitrary (but reasonable) generation number and get a plot. It might be helpful to also be able to generate a movie of a sequence of grids. Use good code design so that you can modify the rule set later to create new automata.

2 Counting active cells

One natural question arises: how many cells are active in generation k? We wish to explore this question and prove some related ideas.

> **Task: Counting**
> Using your implementation, generate many frames, at least the first 50 generations. What patterns do you observe? You'll notice that UWA repeats specific structures after so many generations How many cells are active in those generations? *(Hint: Every so often the grid looks complete. How many cells are active then? Write a formula for that.)* Make a plot of the number of active cells per generation for a few hundred generations.

In class, we have talked about recurrence relations to define sequences. For example, the Fibonacci sequence is defined $F_0 = 1, F_1 = 1, \forall n \geq 2\, F_n = F_{n-1} + F_{n-2}$. It will be difficult to come up with a

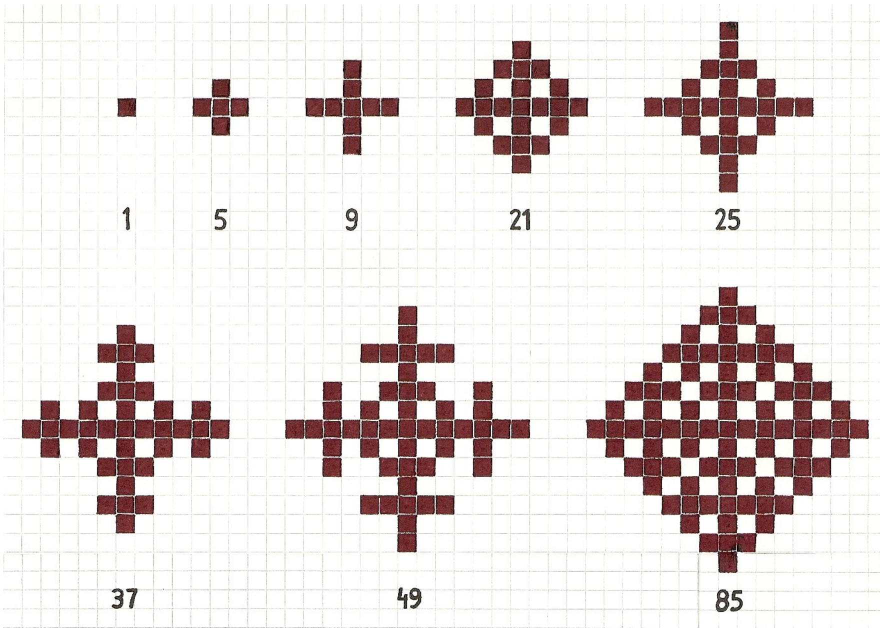

Figure 1: Illustration of the first several generations of the Ulam-Warburton Automaton with a count of the number of active cells. This figure is borrowed from OEIS entry A147562.

recurrence relation for the number of active cells, but let's try! First, let's think about an upper bound for the number of cells. This won't be an equation for the number of cells active at a given iteration but instead will tell us that at most that many cells are active. It is a function that lies above the curve you just generated. For example, $f(n) = n! + 50$ is an upper bound for the curve since it always lies above it. However, it is not a tight upper bound since the factorial function grows much more quickly. We instead want a tight upper bound, one that hugs right next to the curve. In fact, our bound is tight in the sense that for any generation k, I can find a later generation where the upper bound is equal to the number of active cells.

Task: Upper bound
Using the pattern you observed in the previous portion, propose an upper bound function (of a closed/non-recurrent form) for the experimentally derived curve. You can visually check it by plotting it on the same graph as your curve. Is your bound tight? While this plot is strong evidence you have an upper bound, we have to prove it. Use induction (or any other proof method) to prove that you have an upper bound of the equation for the repeating pattern in the last part.

Now that we have some intuition about the number of active cells, we are ready to try and write a formula to describe the sequence. Since cells never turn off, we could express the number of active cells at generation k by just summing up the number of cells turned on at any given generation. Let $u(k)$ be the number of cells turned on at generation k. Before we create the automaton, we turn on no cells so $u(0) = 0$. We then activate 1 cell to get things going so $u(1) = 1$. For the next generation, we turn on 4 cells so $u(2) = 4$.

Task: Number of cells turned on
Graph the number of cells turned on in each generation. You can start by doing this experimentally using your implementation. Then, you can propose various formulas and see if they are identical to your experimental results. It is likely not obvious what the correct formula is so here's a hint: it is of the form $c \times b^{f(k-g)}$ where c is a rational number, b and g are integers, and $f(n)$ is a function that describes the number of 1s in the binary expansion of n (or in this particular case $f(k-g)$ is the number of ones in the expansion of $k-g$. k is the generation number. Look up the Hamming weight function for why $f(n) = n - \sum_{k=1}^{\infty} \lfloor \frac{n}{2^k} \rfloor$.

Now that we have a formula for the number of cells turn on, we can return to writing the number of cells active.

Task: Number of cells active
The number of cells active at generation k is the sum of the number of cells activated in all the previous generations. Write this symbolically. Can you show that this is indeed the correct answer? Can you prove that your upper bound is an upper bound? Can you prove it is a tight upper bound?

3 A variant

Great job! You have done great work so far in exploring UWA. We might use UWA to naively model disease spread. If you get sick your neighbor will get sick the next time. However, I truly hope that you get better after a while. Our model currently would force you to be sick always. Let's fix that. We can change the rule so that an on-cell deactivates after being on for so long.

Task: Implement UWA variant
Implement this new variant of UWA in a program. Write down the new rule symbolically and in words. Make a movie and bask in the fractally glory that you just created. It looks awesome!

In this new variant, you have a parameter for the model: how long you wait before an on-cell deactivates. Let's call this number d for delay. We might be curious what that does to the sequence of on-cells we saw previously. Let's investigate.

Task: Counting cells again!
Consider many values of d. $d = \infty$ is identical to the standard UWA. Plot the number of active cells for a variety of d values such as 1, 2, 5, 15, 50, 100. Do you notice anything interesting? Is the standard UWA an upper bound? Can you prove that for a specific d? What about for a general d?

4 Creative time

Just like before, our model is still not really comprehensive enough. For example, if you get well after some delay d, we currently have you getting instantly sick as soon as exactly one neighbor gets sick. Maybe we should have a wellness period where your medicine is still in effect and you cannot get sick. Maybe your neighbors should include not just the four adjacent cells but also the four diagonal cells. Maybe there should be some randomness in the model. Maybe some parts of the grid are more vulnerable and should have a different d than other regions. Maybe you should start with not just one on cell at the start but many on cells.

Task: Investigate
Come up with a new variant. Change your implementation so that you can plot it and make the same movies and graphs of active cells as we have before. See what you can prove. This is your time to just have fun and potentially make something that no one else has done before. You might even prove something brand new!

5 Evaluation

This project is difficult and by choice. Even attempting it is admirable. You will be graded individually depending on your relative effort and progress. Learning is about pushing your boundaries, not just meeting some standard, so it's okay if you do less than someone else. It's also okay if you do a lot more. Your evaluation will come from a written report that you submit, an interview to ask any follow-up questions, as well as a qualitative judgment of your effort level determined by the conversations we had during your work and how long you spent on the exercise. **You should work on this nearly every day throughout the week instead of just waiting until the last second!** Your grade will include a bonus and be scored out of 120% of the regular homework since this is much more difficult. You are welcome to consult the internet and other resources to learn ideas. This is one of the rare cases that if you are stuck and after working for a reasonable time, it is actually okay to Google for the answer (although for some parts you likely will not find an answer through Google). **You are not allowed to immediately Google though. You should think on your own or by talking to someone first. You must cite any resource you used.** Treat this like a real research experience. It is okay to learn and borrow ideas from others but you cannot take credit for them. If you use someone else's proof you should explain in words what it means to illustrate you understand it. Try and come up with a clearer version of the proof.

Task: Reflection
What did you learn from this exercise? What parts did you enjoy most and least? What were the major challenges you had to overcome? What do you wish you had done differently? How long did you spend on this project?