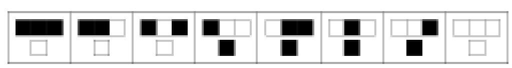
Researcher: Taylor Dupuy
Advisor: Aaron Wootton
Semester: Fall 2004

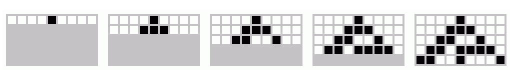
Investigating Cellular Automata

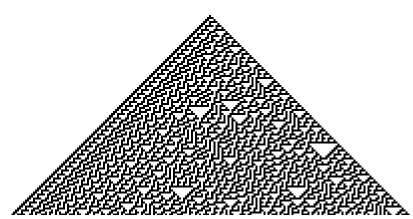
An Overview of Cellular Automata:

Cellular Automata are simple computer programs that generate rows of black and white cells based on rules which "update" each cell based on its neighboring cells. An example of this would be a cellular automata having a row containing one black cell surrounded by two white cells where in this specific automata it has a rule that says for each cell of this type, that is, for each black cell surrounded by two white cells it will be updated to a black cell in the next row. Rules like this for all possible conditions which are defined exhaustively. An example of a cellular automata rule 30[1] is shown below. The box directly below this text depicts the rules by which the cellular automata is updated.



The image below and to the left depicts the cells being updated step by step for the first five steps. The image to the right depicts the evolution of cellular automata rule30 after 100 steps.[2]





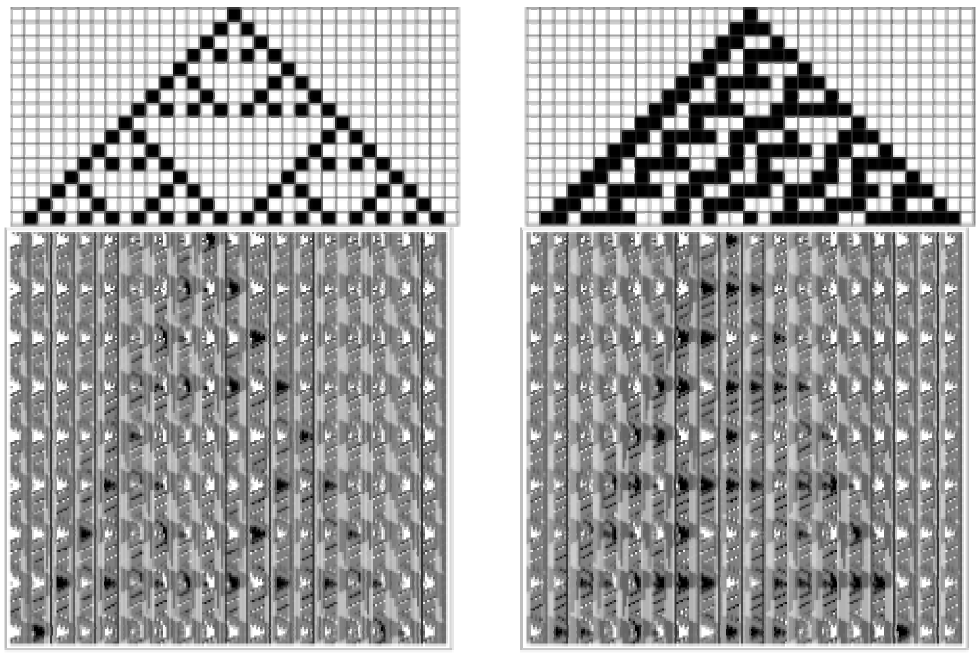
An interesting fact about rule 30 is the that the center column of rule 30 has passed every randomness test it has gone through and that it is in fact is used in the random number generator for Stephen Wolfram's *Mathematica* program ever since the program's inception.

Universality is the property of being able to perform different tasks with the same underlying construction by being programmed in a different way.[3] In plain English this is to say that a computer program, upon giving it different initial conditions, will perform tasks that mimic another computer program in the same way the program which it is

---

[1] Naming scheme: reading the bottom row from right to left we have in binary 2 + 4 + 8 +16 which is equal to 30. Also note that from right to left the initial conditions for which the given rule are applied also go from 0 to 7 in binary.
[2] These and more images can be found at http://wolframscience.com/downloads/basicimages.html.
[3] Weinstein, E http://mathworld.wolfram.com/Universality.html

mimicking would construct the said construction. I believe it would be helpful to see a picture of a universal cellular automata which will illustrate the notion of universality better than any confusing sentence :-).



adapted from Wolfram, S. *A New Kind of Science*. Wolfram Media, pp. 646--647, 2002. [4]

The above picture shows a 19-color cellular automation with significantly more rules than the elementary cellular automata discussed previously. Nevertheless I think the illustration of the notion of universality is clear from these images.

I must note that this property of universality is not limited to cellular automata. The property of universality can be found in Turing machines, tag-cyclic systems, and register machines, which are other idealized computational systems.

Background:

In 2002 Stephan Wolfram of Wolfram Research published *A New Kind of Science* a history and investigation of cellular automata and other computational systems. The results presented in *A New Kind of Science* are atypical of a major work because it does not contain a formal mathematical structure by which the ideas presented in the book are discussed.

After having read *A New Kind of Science* I developed an interest in the mathematical structure that lies behind these idealized computational systems. I wish to explain why computation behaves in a way it does. I wish to explain formally why certain automata

[4] Images taken from http://mathworld.wolfram.com/UniversalCellularAutomaton.html

have the property of universality or complexity, what properties (rules) determine the behavior of the automata (complexity, and universality).

In order to do so, I believe I need to understand and/or develop a mathematical structure in which I may examine automata as an operation on a set.

Because I'm most familiar with elementary cellular automata, I wish first understand a particular universal elementary automata, elementary cellular automata 110. Rule 110 is the smallest universal system which has been shown to be universal. It was shown to be universal by both Cook in 2004 and Wolfram in 2002.[5] At the end of the semester I wish to explain why rule 110 is universal and be able to prove it formally. I've commented on the mathematical structure below and I must note that my research will include finding/understanding a rigorous representation these operations performed by idealized computational systems.

Structure:

The structure in which one can represent an automata seems not to be unique. I'm considering representing cellular automata as a set with the operation on the set of the updating rule for the particular cellular automata. I've shown below a way of representing an elementary cellular automata in a finite field. This method below is not necessarily the one I will choose in representing the automata in midterm of final reports.

[5] Cook, M. "Universality in Elementary Cellular Automata." *Complex Systems* **15**, 1-40, 2004., Wolfram, S. *A New Kind of Science*. Champaign, IL: Wolfram Media, pp. 642-644, 2002.

1. Defining Elementary Automata over a Finite Field:

We shall call the black and white states elements of a set $B = \{1, 0\}$ which we shall define in conjunction with operations which will allows us to classify this set as a finite field.

If we have a cellular automata row with n components, or n nontrivial rows we shall define this row r_0 as follows.

$$r_0 = (b_1, b_2, \ldots, b_n) \in B^n \text{ where } b_1, b_2, \ldots, b_n \in B.$$

This allows us to use a vector space structure on the cellular automata and we make write r_0 as a linear combination of basis elements

$$e_1 = (1, 0, \ldots, 0), e_2 = (0, 1, \ldots, 0), \ldots, e_n = (0, 0, \ldots, 1).$$

And we write,

$$r_0 = \sum_{i=1}^{n} b_i e_i .$$

2. Defining Automata Function:

Our goal is to define the generalized automata function $A : B^n \to B^{n+2}$ for which we may take any rules for a given cellular automata and use them to operate on the row r_0 as to define the recurrence relation $A(r_0) = r_1$ for $r_0 \in B^n$ and $r_1 \in B^{n+2}$. It is convenient at this point to define the functions for updating each cell based on the adjacent cells.

2.1. Updating Cells

We desire the function $\alpha : B^3 \to B^1$. α is defined exhaustively, giving all the cases for each possible element in B^3.

$$\begin{aligned} \alpha(0,0,0) &= \alpha_0 \\ \alpha(1,0,0) &= \alpha_1 \\ \vdots \quad & \quad \vdots \\ \alpha(1,1,1) &= \alpha_7 \end{aligned}$$

for $\alpha_1, \alpha_2, \ldots, \alpha_7 \in B$. Now in order to get these single bit values into a form which we can use we need to do perform a transformation $\phi_{jk} : B \to B^k$ which maps the single bit to a subspace of B^k with the same dimension (which in this specific case is 1, but which could possible be generalized for in other computational systems). Now all that is left is do it deconstruction of some element of B^n into a form where we may apply the α function and then reconstruct the components under the image into a B^{n+2} vector space.

2.2. Application of alpha-function

Allow us to deconstruct r_0 into components which will be more useful in our venture. Let $\Psi_i : B^n \to B^3$ defined by

$$\Psi_i(r_0) = \beta_i = (b_i, b_{i+1}, b_{i+2})$$

for $i \in \mathbb{N}_{n-2}$ and let us create some $\beta_{-1} = (0,0,b_1), \beta_0 = (0,b_1,b_2), \beta_{n-1} = (b_{n-1}, b_n, 0)$, and $\beta_n = (b_n, 0, 0)$ which all can be used to apply α to part of b in B^n. Taking all β_p under the image of α for $p = -1, \ldots, n$ we get that

$$\begin{array}{ccc} \alpha(\beta_{-1}) & = & \alpha_1 \\ \alpha(\beta_0) & = & \alpha_2 \\ \vdots & & \vdots \\ \alpha(\beta_n) & = & \alpha_{n+2} \end{array}.$$[6]

Now applying $\phi_{j(n+2)} : B \to B^{n+2}$ to each $j = 1, \ldots, n+2$ where $\phi_{j(n+2)}$ is defined by

$$\phi_{j(n+2)} = \underbrace{(0, \ldots, 0, \underbrace{\alpha_j}_{jthspot}, 0, \ldots, 0)}_{n+2\ terms}$$

we get components of $A(r_0)$ which when we take the sum of all of them we get $A(r_0)$.[7] So we have

$$\begin{aligned} A(r_0) &= r_1 \\ &= \sum_{j=1}^{n+2} \phi_{j(n+2)}(\alpha_j) \\ &= (\alpha_1, \alpha_2, \ldots, \alpha_{n+2}) \end{aligned}$$

where α_j is given by

$$\alpha(\Psi_i(r_0)) = \alpha_j .$$

[6] The sloppiness of this argument is to be corrected

[7] note that α_j can also be as coefficients for a linear combination of the basis elements for a basis of B^{n+2}.