Life As An Algorithm

Kathryn Lindsey

March 4, 2011
This presentation is inspired by a Williams College course “Life As An Algorithm” by Prof. Duane Bailey
Self-replicating machines

- A *self-replicating machine* is an artificial construct that is theoretically capable of autonomously manufacturing a copy of itself using raw materials taken from its environment.
Can you think of some uses for SRMs?
• Early Thinkers: Rene Descartes, William Paley, John Bernal, Stephen Kleene

• John von Neumann, 1948 proposed a kinematic model. Used a “sea” of parts, a memory “tape,” 8 types of components. Problems: how to actually make it? sea of parts expensive; hard to analyze mathematically

• Stanislaw Ulam: suggested von Neumann use a mathematical model
  -> cellular automata
John Conway’s “Game of Life”

The Rules

For a space that is 'populated':
  Each cell with one or no neighbors dies, as if by loneliness.
  Each cell with four or more neighbors dies, as if by overpopulation.
  Each cell with two or three neighbors survives.

For a space that is 'empty' or 'unpopulated':
  Each cell with three neighbors becomes populated.

Show GOL
Definition of CAs

- A *cellular automaton* is a quadruple \( \{L,Q,N,f\} \)
  where
  - \( L \) is a lattice (the tiles),
  - \( N = \{n_1,\ldots,n_k\} \) a finite subset of \( L \) (the neighborhood)
    (i.e. the neighborhood of a tile \( x \) in \( L \) is \( x+N \))
  - \( Q = \{q_0,\ldots,q_m\} \) is a finite set (the set of states)
  - \( f: Q^N \rightarrow Q \) (local evolution function)

An element of \( Q^L \) is a *configuration*.
For \( c \) in \( Q^L \), denote the value of \( c \) at \( x \) by \( c(x) \).

\( f \) defines a *global evolution function* \( F: Q^L \rightarrow Q^L \) by
\[
F(c)(x) = f(c(x+n_1),\ldots,c(x+n_k))
\]
Theorem (Hedlund, 1969)

Let $F: \mathbb{Q}^{\mathbb{Z}^d} \rightarrow \mathbb{Q}^{\mathbb{Z}^d}$. The following are equivalent:

1. $F$ is the global evolution function of a cellular automaton with lattice $\mathbb{Z}^d$.
2. $F$ is continuous in the product topology, and commutes with translations of $\mathbb{Q}^{\mathbb{Z}^d}$.

(Here translation means a map $t: \mathbb{Q}^{\mathbb{Z}^d} \rightarrow \mathbb{Q}^{\mathbb{Z}^d}$ given by $t(c)(y) = c(x+y)$ for some $x$ in $\mathbb{Z}^d$.)
Elementary 1-D CA’s (à la Wolfram)

- “Played” on a 1-d line of “cells.” Update rule based on states of self and the two adjacent neighbors.
- Wolfram code. Ex: “rule 22”

```
111 | 110 | 101 | 100 | 010 | 010 | 001 | 000
0   0   0   1   0   1   1   0 = 22
```

Show Golly
Langston’s Ant

Grid of square tiles, each either white or black. One tile has an “ant” on it. The ant follows these rules:

- If it is on a white cell it makes a 90 degree turn to the left.
- If it is on a black cell it makes a 90 degree turn to the right.
- As the moves to the next square, the square that it is on changes color.
Langston’s Ant

Grid of square tiles, each either white or black. One tile has an “ant” on it. The ant follows these rules:

- If it is on a white cell it makes a 90 degree turn to the left.
- If it is on a black cell it makes a 90 degree turn to the right.
- As the moves to the next square, the square that it is on changes color.

Open question: is the “highway” an attractor? (I.e. does every initial configuration -> highway?)

In 2000, Gajardo et al. proved the ant can simulate a Turing machine.

It is known ant’s trajectory is always unbounded, regardless of initial configuration (Cohen-Kung Theorem)
The von Neumann Universal Constructor

• A self-replicating machine in a CA environment.

• Uses 29 states. A 'tape' of cells encodes the sequence of actions to be performed by the machine. A ‘writing head’ constructs a new pattern of cells -- allowing it to make a complete copy of itself, and the tape.

• (Does this remind you of anything?…Turing machine, anyone?)
(2) mutation inherited in later generations

(1) mutation added here
Hand-waving: CAs ~ Turing Machines

Let T be a Turing machine with alphabet A and set of states \( \Delta \).

Construct a cellular automaton \((L,N,Q,f)\) so that:
- \( L = \) integers (i.e. a “tape”)
- \( Q = A \times (S \cup \{\text{no head}\}) \)
- \( N = \{-1,0,1\} \)
- construct \( f \) so that it reproduces:
  - the write operation of \( T \) on the first component
  - the state update of \( T \) and the movement of \( T \) on the second component

Then the CA \( \{L,Q,N,f\} \) simulates a Turing machine.

Numerous CAs (including the GOL and Rule 110) have been proven to be Turing complete.
More recent work on SRMs

• 1980, NASA & ASEE conducted a study “Advanced Automation for Space Missions” to produce a detailed proposal for self-replicating factories to develop lunar resources without requiring additional launches or human workers on-site.

• 2005, RepRap Project, U. Bath - a machine capable of manufacturing the parts it is made of (but not assembling them)

2005, Cornell U.’s self-assembler. Machine composed of a tower of four “molecubes,” which can revolve around a diagonal, acting as an arm - can pick up nearby cubes and assemble them. Arm is directed by a computer program that is contained in each cube. (Can’t manufacture molecubes.)

video link
String Biology

- Polymerase *transcribes* select portions of DNA into messenger RNA (mRNA).
- The mRNA is *translated* by the ribosome into strings of amino acids, forming proteins.
- DNA/RNA contain both sequences that are expressed as proteins, and regulatory sequences.
- When cells replicate, each cell gets a copy of the DNA.
• A codon is a string of 3 bases. 1 codon determines 1 amino acid. There are $4^3 = 64$ codons; each codon codes for either 1 amino acid or start/stop.

• Some simple pattern matching that happens during gene expression:
  – string recognition: binding sites, start/stop codons
  – In eukaryotes: splice site recognition by the splicosome
Operations on Strings

- Add or delete a character from beginning or end of a string.
- Cut a string into two strings
- Join two strings together
⇒ All these operations are performed on RNA.

Copying errors can occur during transcription/translation.
- replace a letter with a different letter
- insert/delete a letter
(Biological) Retroviruses

- A RNA virus; the RNA is reverse-transcribed into DNA, which is integrated into host cell’s genome, and then undergoes usual transcription/translation processes to express genes carried by virus.
- Examples: HIV, some cancer-causing
- How is this like a computer virus?
Neurons/synaptic signaling

- Neuron: type of cell, transmits/processes info by chemical/electrical signaling. Core component of the nervous system.
- Two states (on/off) - no in between
- Form networks (graphs)
- Compare to info transfer in cellular automata
Another example: eyesight

- The retina consists of photoreceptor cells. In humans these are rods and cones.
- They have two states (on/off). Connect to neurons.
- Remark: eyesight is discrete, has “pixels.”
- Eyes are not a window on the world; what you “see” is an internal “model” in your brain.
DNA:
- a self-replicating machine
- an alphabet of 4 symbols
- information is encoded on a “tape” / 1-d lattice
- biological equivalents of string searches, operations on strings
- complex behavior (basically every function of biological organisms) arises from simple rules (3 DNA bases -> 1 amino acid).

Higher level-functions (brain, vision, etc.) are modeled by 2-state automata (discrete collection of cells, each of which is either “on” or “off”.)

Evolution/genetic changes when offspring “code” is different from parent’s “code”
Reversible CAs

- A CA is said to be *reversible* if every state has a preimage (i.e. surjective).
- Configurations with no preimages are called “Garden of Eden” configurations.
- There is an algorithm for deciding if a 1-d CA is reversible; in 2-d or higher, it is undecidable. (Proof by Kari uses Wang tiles)
- Every d-dimensional cellular automaton can be simulated by a (d+1)-dimensional reversible cellular automaton (Toffoli, 1977)
- Preinjective if and only if surjective (Moore, Myhill, 1962). (Preinjective means if two configurations differ in finitely many places their images are distinct.)
• Some seem to exhibit "sensitive dependence on initial conditions."
• Can they be characterized by their dynamical properties (attractors, etc.)?
• Metrics on spaces of CA’s? Group actions?
• Can exhibit amazingly complex behavior.
“Emergent behavior”

- Why do geese fly in “V”s?
- How do bees (termites, etc.) build hives?
- Flocking (fish, birds, etc.)
- Molecules, proteins, etc.
- ...
- All of chemistry?
- Neurons?
- Image processing networks (eyesight)?
- Human “consciousness”? 
What would scientists living in “Rule 30” deduce about physics? About math?
Is our world discrete?

“Number of spatial dimensions: Exactly three
Number of different electrical charge states: Exactly two, + and −.
Number of chiral parity states: Exactly two, left-handed and right-handed.
Number of directions for time: Exactly two, forwards and backwards.
Number of CPT modalities: Exactly two out of eight, CPT and −C − P − T.
Number of spin state families: Exactly two, bosons and fermions.
Number of measurable spin states of an electron: Exactly two, up and down.
Number of particle conjugates: Exactly two: particle and antiparticle.
Number of different QCD color charge states: Exactly three, R, G, and B.
Number of lepton and quark generations: Exactly three.
Number of leptons or quarks per generation: Exactly two.
Spin of any particle that is a boson: Exactly $n$ ($n$ always a small integer).
Spin of any particle that is a fermion: Exactly $n + 1/2$.
Maximum number of inner-orbit electrons in an atom: Exactly two.

The above list is a small sample of the totality of small-integer phenomena in physics.”

-Intro. to digital philosophy, Fredkin
Is the Universe a Cellular Automaton?

- *Digital philosophy* “works through the consequences of assuming that the universe is a gigantic Turing-Complete cellular automaton.” (Wikipedia, Digital Philosophy)

- Proponents include: Edward Fredkin, Stephen Wolfram, Konrad Zuse.
"If the universe in all ways acts as if it was a computer, then what meaning could there be in saying that it is not a computer?"

-F. Tipler, Tulane U.