Evaluating complexity in cellular automata

Hugo Cisneros

May 16th 2019

Definition

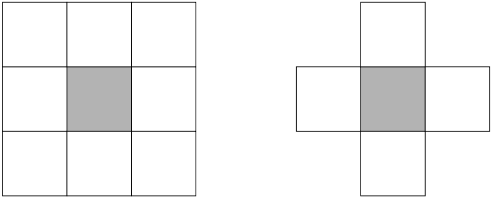
Cellular automaton (CA)

A mathematical model composed of elementary components (cells) that are updated in discrete time steps according to local rules.
The cells can take k possible values (states).

Although very simples, cellular automata exhibit very interesting and unusual properties.
CAs are defined by a local transition table.

1D Automaton:

2D Automaton:



Wolfram's classification

Wolfram has studied extensively the 1D elementary CA and created a classification of the 256 possible rules into 4 classes.

- **Class 1** Fixed homogeneous state is reached
- **Class 2** A pattern of periodic regions is produced
- **Class 3** A chaotic aperiodic pattern is produced
- **Class 4** Complex localized structures are generated

(a) Rule 8 (b) Rule 253

Wolfram's classification

Wolfram has studied extensively the 1D elementary CA and created a classification of the 256 possible rules into 4 classes.

Class 1 Fixed homogeneous state is reached

Class 2 A pattern of periodic regions is produced

Class 3 A chaotic aperiodic pattern is produced

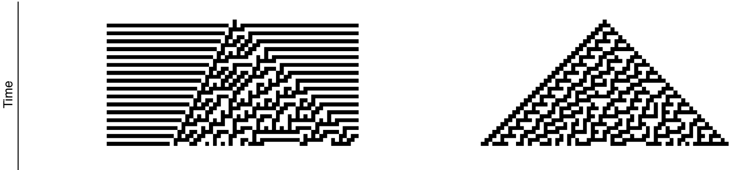
Class 4 Complex localized structures are generated

(a) Rule 4 (b) Rule 37

Wolfram's classification

Wolfram has studied extensively the 1D elementary CA and created a classification of the 256 possible rules into 4 classes.

- **Class 1** Fixed homogeneous state is reached
- **Class 2** A pattern of periodic regions is produced
- **Class 3** A chaotic aperiodic pattern is produced
- **Class 4** Complex localized structures are generated



(a) Rule 45 (b) Rule 30

Wolfram's classification

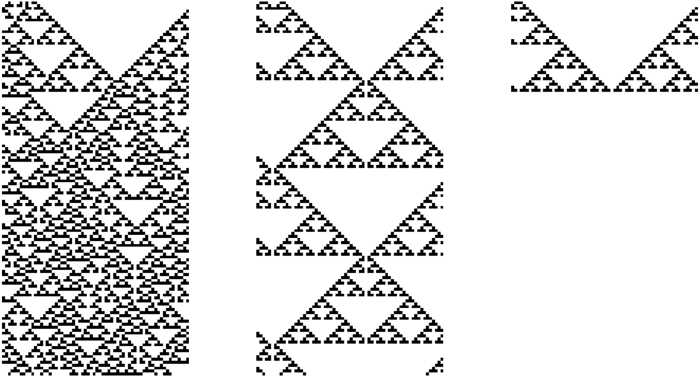
Wolfram has studied extensively the 1D elementary CA and created a classification of the 256 possible rules into 4 classes.

- Class 1 Fixed homogeneous state is reached
- Class 2 A pattern of periodic regions is produced
- Class 3 A chaotic aperiodic pattern is produced
- Class 4 Complex localized structures are generated

(a) Rule 110

Rule 110 is computationally universal.

Sensitivity to initial state



(a) Chaotic - (Class 3)

(b) Ordered - (Class 2)

(c) Homogenous - (Class 1)

Figure: Rule 22 - Random initial state top left 12 cells. Size: 64 cells, ran for 128 steps

Compression based study of 1D CAs

Idea:
Use the compressed length of a CA state as a proxy for its *complexity*.

Many definitions of complexity: here the Kolmogorov complexity is constant for a given rule.
We are looking for a more *qualitative* interpretation of complexity, similar to what human beings perceive.

Compression based study of 1D CAs

Idea:

Use the compressed length of a CA state as a proxy for its *complexity*.

A CA state is represented as a string of 0s and 1s (or something else for more states) that can be fed to a compression algorithm (gzip in the rest of the presentation).

 $\Rightarrow$ **`0110010`**

Examples:

 $\rightarrow \quad \ell = 13$

 $\rightarrow \quad \ell = 34$

Compressed length for single-cell initialization

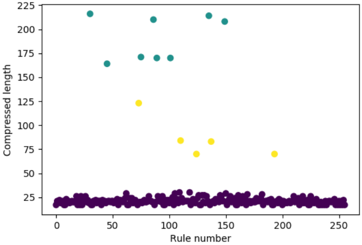


Figure: Single cell activated in initial configuration, evolve an automaton of size 1024 for 512 timesteps

→ **Obtained classification matches exactly the Wolfram "manual" classification (originally observed by Zenil, 2010)**

Influence of the compression algorithm

Setting: Random initialization for 3 rules, the state is compressed every 50 timesteps.

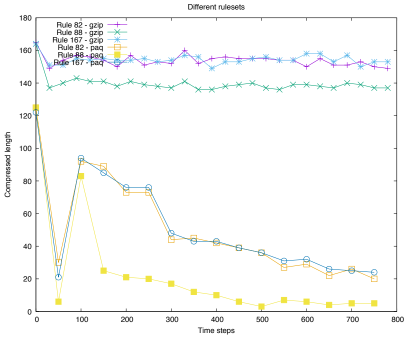


Figure: Temporal evolution of compressed length for 3 rules. Comparison of gzip with PAQ

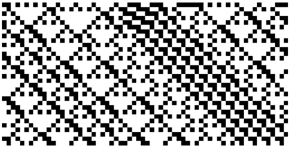


Figure: Rule 82

2D CAs

With 2D CAs, the number of rules is much higher: 2^{512} rules total, 2^{102} if we require the rules to have all symmetries.

Gets even bigger if we add more states and/or larger neighborhoods.

- There is no chance of sampling all the rules (not even a *significant* portion of it)
- Some parts of this space might be more interesting than others

$\rightarrow$ **We need a way of guiding this search towards interesting rules**

Compressed length repartition

Sample rules at random and compress the state as an "unrolled string".

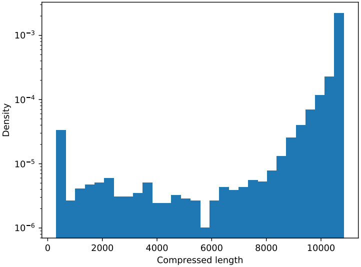


Figure: Compressed length distribution for $k = 2$, 2D CAs. Grid size is 256×256, automata are ran for 1000 time steps.

Extreme cases — Illustration

(a) High compressed length

(b) Low compressed length

Intermediate cases — Illustration

(a) Compressed length $= 2914$

(b) Compressed length $= 6753$

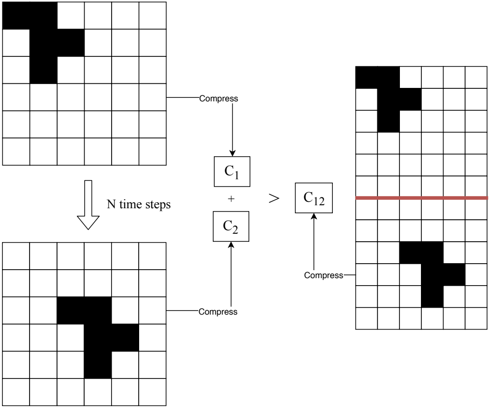
Beyond compressed length

Compressed length is not the right metric for 2D CAs

- Most rules are at the extremes of the graph.
- *Interesting* rules might have very different compressed lengths, what matters is the dynamic of this complexity.

Joint compression score

To measure the stability of patterns for an evolving 2D CA, we use the joint compression, i.e. the compressed length of the concatenation of two steps relatively far apart in time.



Results on 2D CA rules

The score we compute is $\dfrac{C_1 + C_2}{C_{12}}$. Distribution on the histogram below.

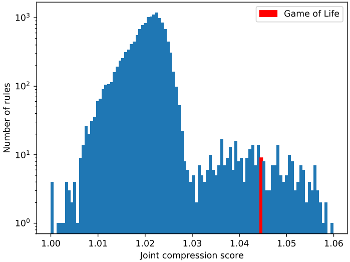
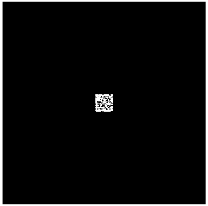


Figure: Histogram of joint compression score for 13000 random rules

Two parts:

- A large portion of rules that have **very high compressed length and no structure** (low joint compression score).
- Other group (∼1%)that seems to have **much more structure** (although not all rules do and not all “structured rules” exhibit interesting behavior).

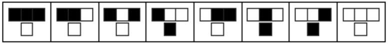
Example rules — 2 states



Langton's lambda parameter

The lambda parameter is defined with respect to a *quescient* state (usually 0), as the proportion of transitions that lead to any other state.

Example of 1D rule 22:



$$\boxed{\lambda = \frac{3}{8}}$$

Relation to lambda parameter

The vast majority of rules of 2-states 2D CAs have a λ close to 0.5. For the graph below the rules were sampled with a *lambda* uniform over $[0, 1]$.

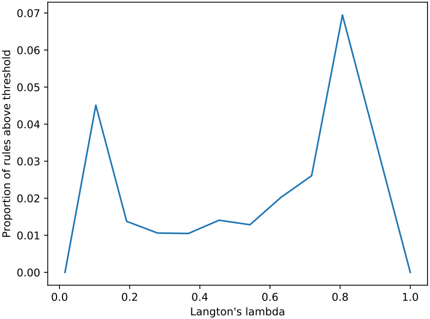
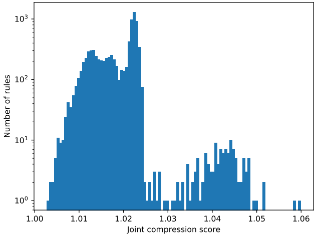
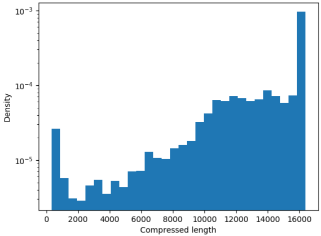


Figure: Proportion of rules that would qualify as interesting with the defined scheme

3 States — 2D CAs



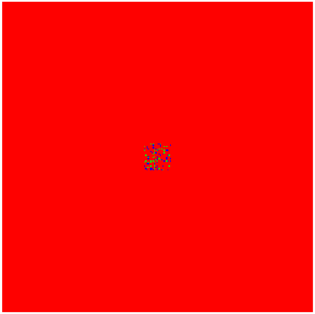
(a) Score histogram for 3-states automata

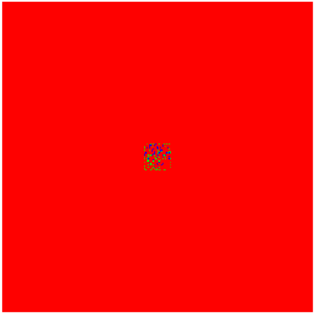


(b) Compressed length histogram for 3-states automata

Similar distributions → if this can be generalized, this would be a first toward creating a systematic approach for finding interesting rules.

Example rules — 3 states





Further work

- Study the influence of the initial state.
- Add some input/output capabilities ?
- Refine the metric or find some other ?

Thank you for your attention!