

An Overview of Cellular Automata and their Applications

Petar Peychev
30 November 2018

A report submitted in fulfillment of the requirements for the professional development strand of the ISYS10301 module, School of Science and Technology, Nottingham Trent University

“Maria could almost see it: a vast lattice of computers, a seed of order in a sea of random noise, extending itself from moment to moment by sheer force of internal logic, “accreting” the necessary building blocks from the chaos of non-space-time by the very act of defining space and time.”

(Egan, 1994)

Abstract

This report introduces the concept of cellular automata, illustrates cellular automaton variants and overviews applications for cellular automata systems in road traffic modeling and procedural content generation.

1. Introduction

Cellular automata is a term, referring to a type of discrete computational system, which was first discovered by the Hungarian-American mathematician John von Neumann (1951), but gained wider popularity outside academia in the 1970's with John Conway's Game of Life (Gardner, 1970). Cellular automata vary widely in their properties, but can be categorised by the common feature of being "comprised of a number of discrete elements called cells", where "each cell encapsulates some portion of the state of the system" (Yuen and Kay, 2009). Typically, these cells are positioned uniformly in a grid and governed by spatially localised update rules, dependant on the state of neighbouring cells. Most automata systems represent time discretely by handling cell updates simultaneously, resulting in a series of "generations" or "steps".

2. Elementary Cellular Automata

The simplest non-trivial example of a CA system is Stephen Wolfram's "elementary cellular automata", which consists of a 1-dimensional grid of cells, each carrying a binary value of either 1 or 0 and a universal update rule, which takes into account the states of the two immediately neighbouring cells in the previous generation and either retains or changes the state of the cell in the next generation (1982). With 1-dimensional cell grids, each ruleset can be easily explicitly illustrated, as shown in figure 1.

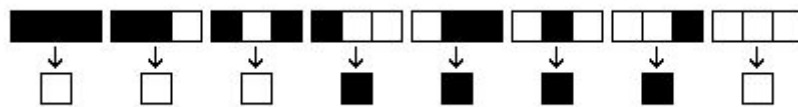


Figure 1: Visual representation of a ruleset for an elementary cellular automaton

These rulesets are also often described by the decimal representation of the binary number, which is formed by listing the possible outputs of the ruleset. The ruleset in figure 1 would be represented in binary as 00011110 and in decimal as rule 30.

2.1. Classes of ECA

Stephen Wolfram, in his paper "Universality and Complexity in Cellular Automata", further analyses ECA's and categorizes them into four types (1984):

- Class 1: Convergence to uniformity (See figure 2)
- Class 2: Convergence to repetition or stability (See figure 3)
- Class 3: Randomness (See figure 4)
- Class 4: Complexity (See figure 5)

Automata of class 1, 2 and 3 have interesting properties, such as high sensitivity to initial conditions, pseudo-random generation with simple non-random initial conditions and fractal pattern generation (See figure 6). Wolfram conjectured that class 4 systems are capable of universal computation, which was later proved by Matthew Cook (1984; 2004).



Figure 2: Class 1 cellular automata - uniformity¹

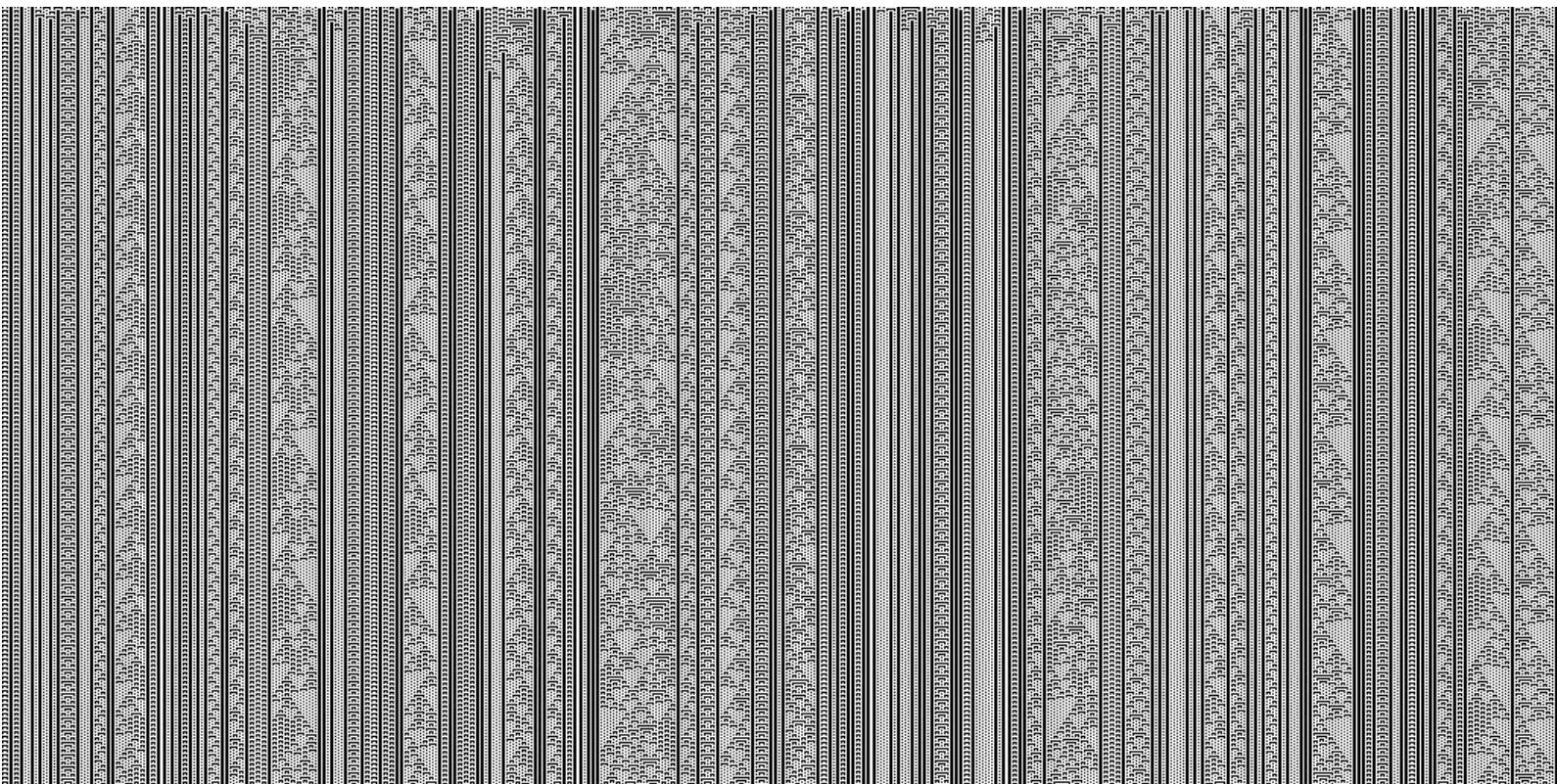


Figure 3: Class 2 cellular automata - repetition¹

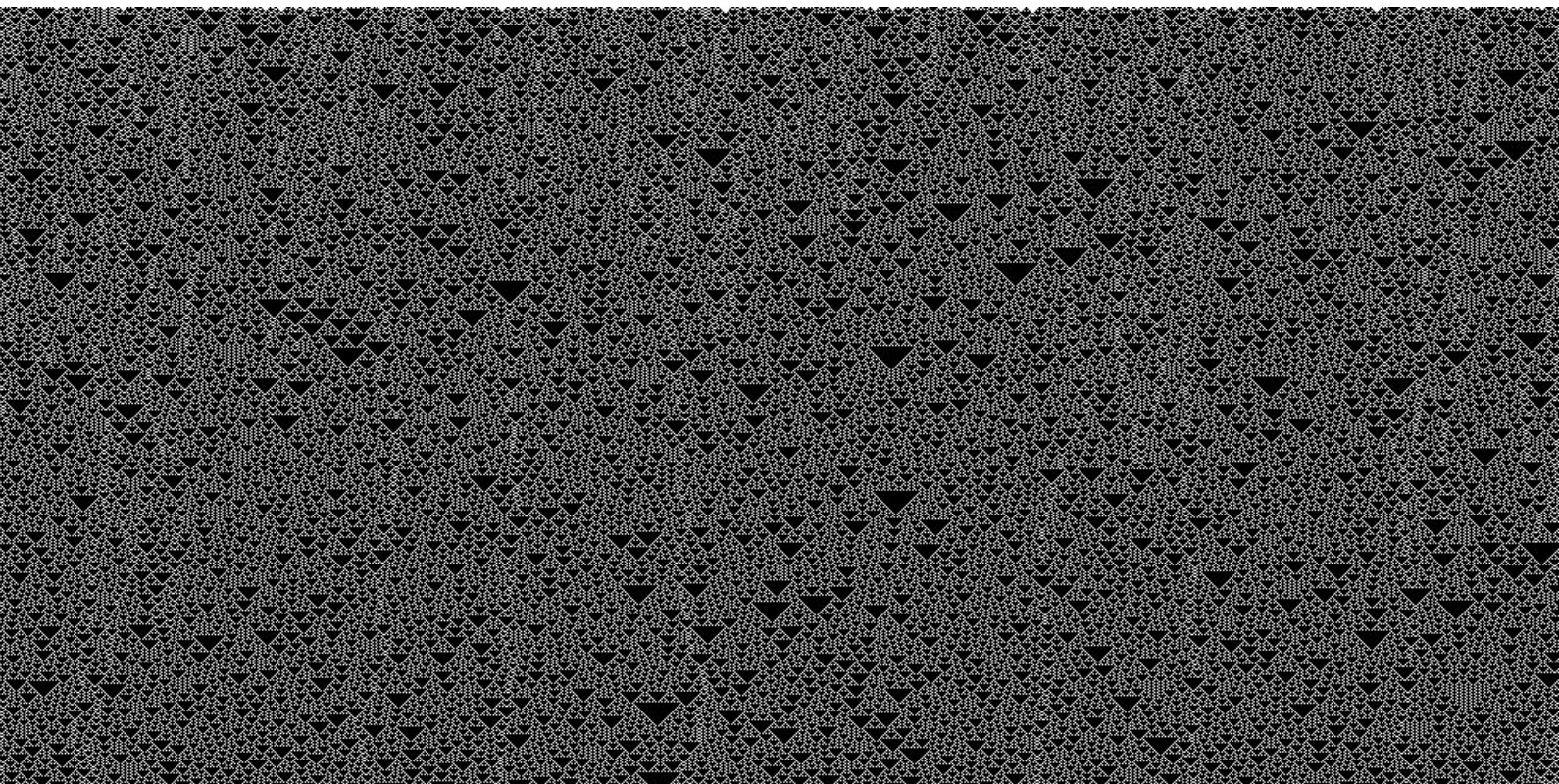


Figure 4: Class 3 cellular automata - randomness¹

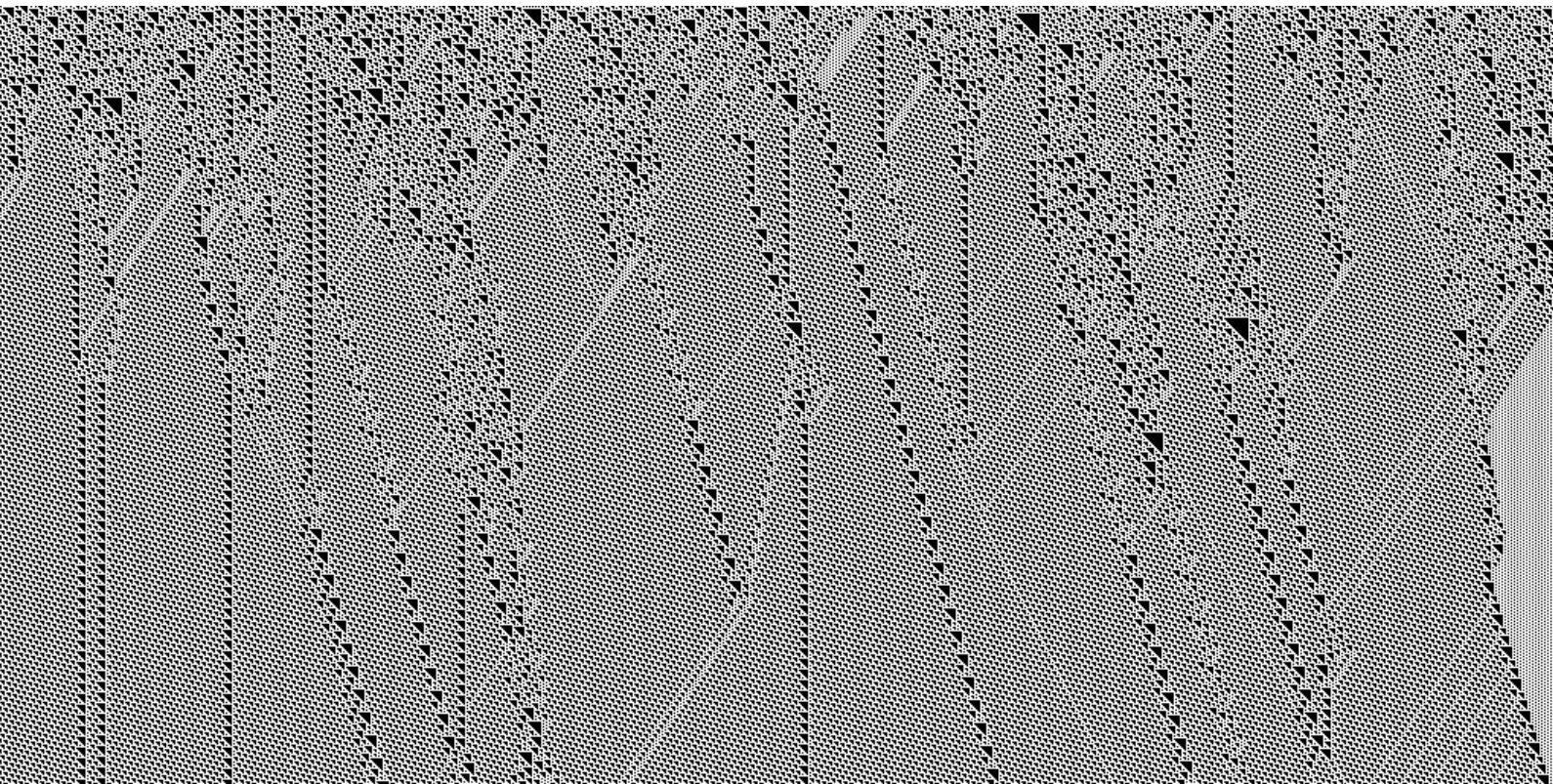


Figure 5: Class 4 cellular automata - complexity¹

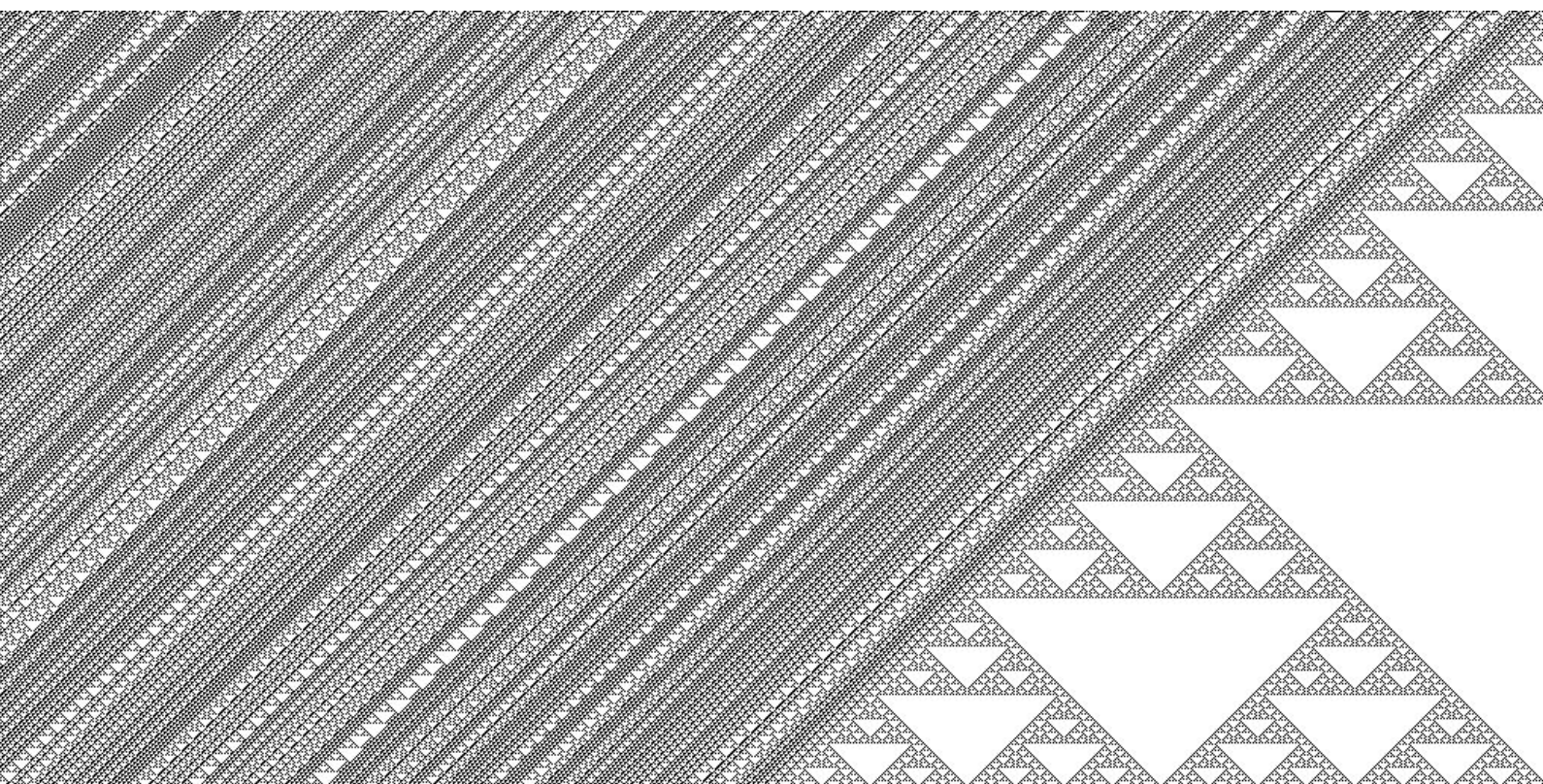


Figure 6: ECA rule 26 generating a series of Sierpinski Triangle fractals.¹

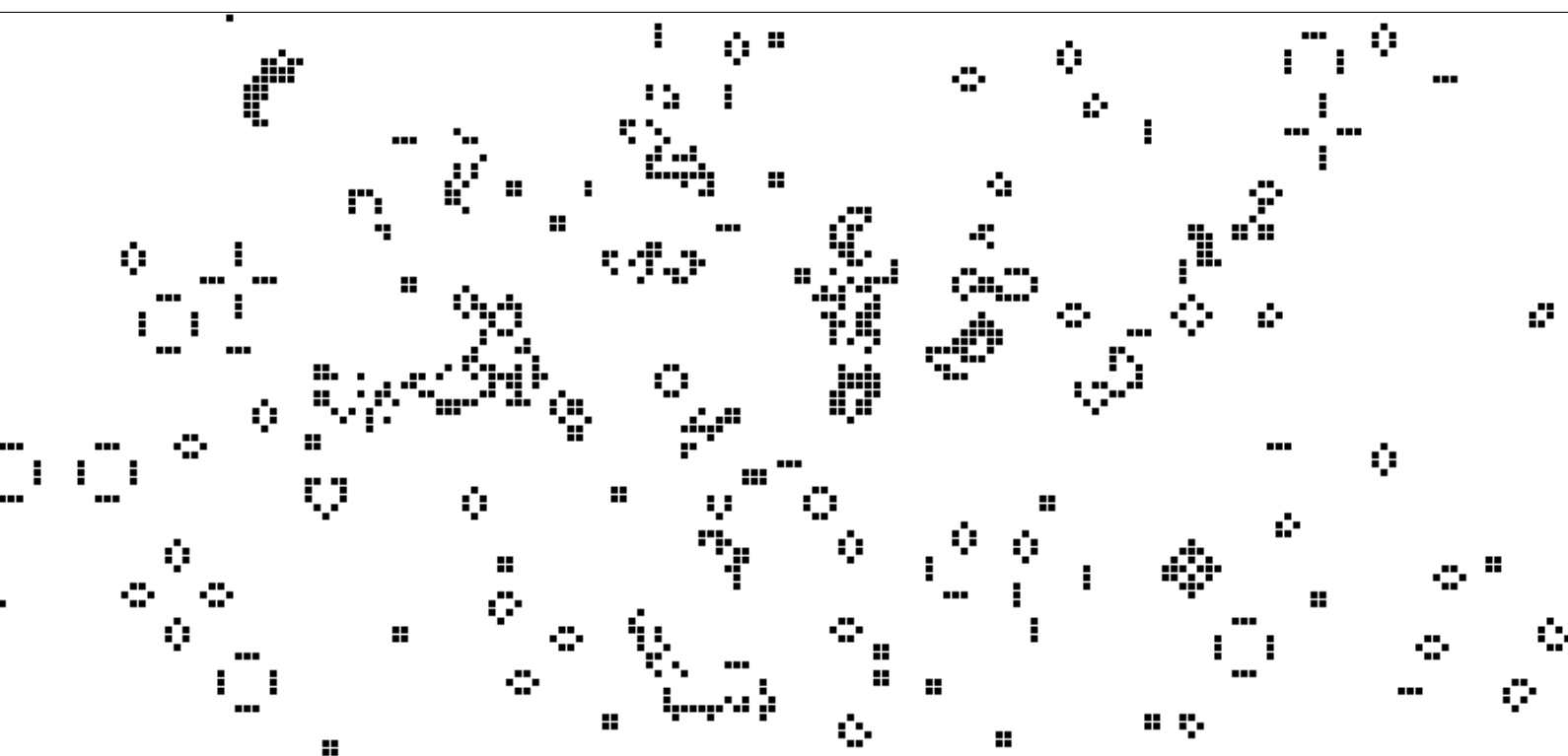


Figure 7: 300th generation of Conway's Life, starting with a random initial configuration²

¹ Code for generating these is published at <https://github.com/PetarPeychev/elementary-cellular-automata>

² Generated using <http://pmav.eu/stuff/javascript-game-of-life-v3.1.1/>

3. Two-Dimensional CA

Moving away from the simplest case of ECA, there exist many possible CA, varying in number of dimensions, neighbourhood size, number of possible states, continuous states, non-rectangular grids, non-deterministic update rules, etc.

3.1. Conway's Life

Perhaps the most famous and deeply studied two-dimensional CA is John Conway's Life, which was created by the British mathematician through experimentation, with the goal of creating an unpredictable and interesting CA, which would exhibit life-like behaviour (See figure 7). This ruleset was later proved to be turing-complete (Rendell, 2016).

3.2. Wireworld

Another fascinating cellular automata, which is particularly well-suited to expressing and modeling logic gates is "wireworld". It was originally invented by Brian Silverman in 1987 (Dewdney, 1990). The cells in wireworld take one of 4 different states: empty, electron head, electron tail or conductor and update in a Moore neighbourhood with the following rules:

- empty -> empty
- electron head -> electron tail
- electron tail - conductor
- conductor -> electron head (if one or two of the neighbouring cells are electron heads, otherwise remain conductor)

Being turing complete, wireworld allows for the construction of any electronic logic circuit, with Mark Owen and David Moore going as far as building a full wireworld programmable computer, including five 7-segment digit displays (See figure 8) (2004).

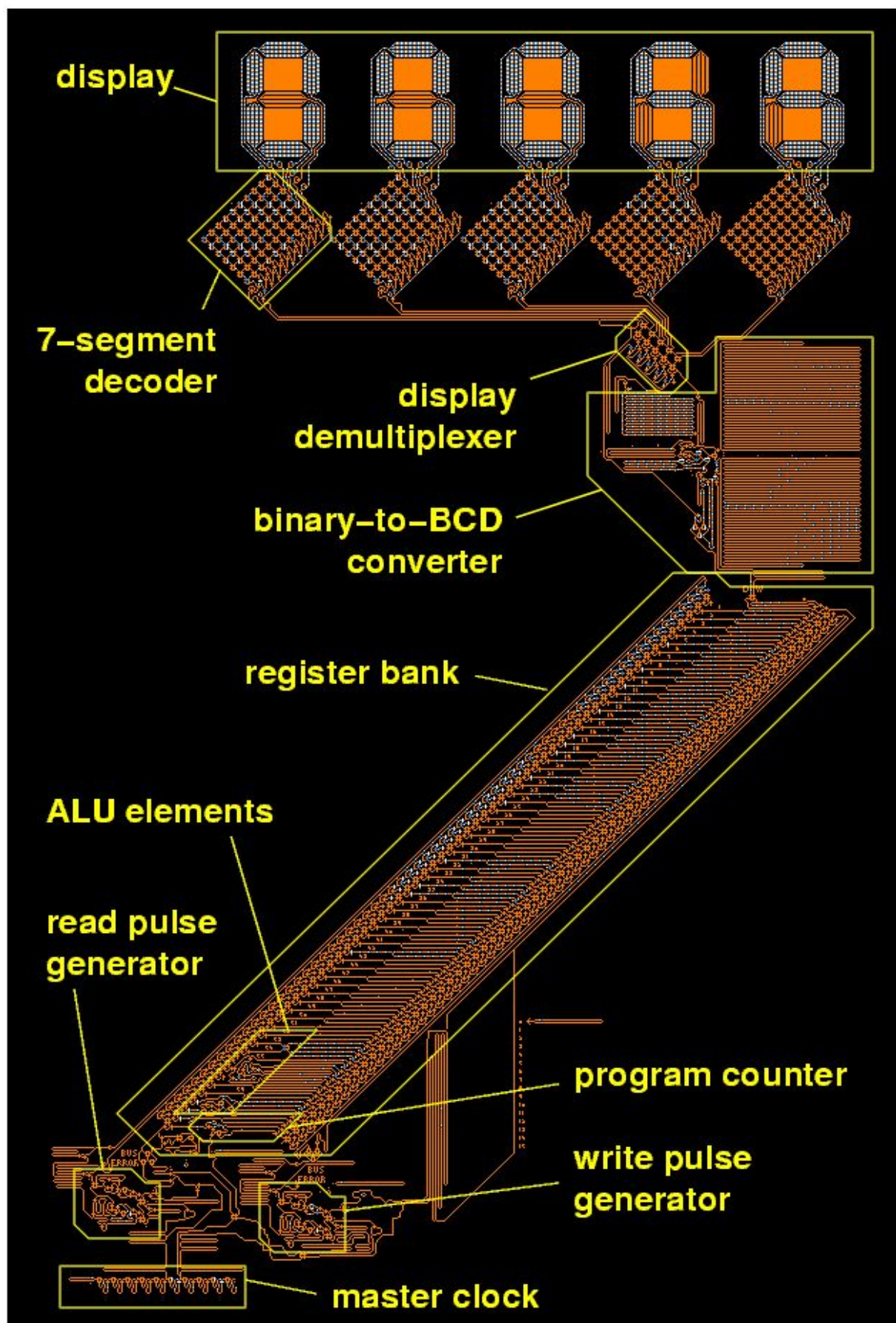


Figure 8: Moore and Owen's wireworld computer (2004)

4. Applications of Cellular Automata

There exist many possible applications of CA, such as musical composition, structural design, modeling fluid dynamics, modeling seismic wave propagation, digital mechanics, etc. Localisation of update rules for CA models allow for high parallelisation of computation, making CA simulations highly efficient (Spezzano and Talia, 1999). This section will outline two examples: road traffic modeling and procedural content generation for games.

4.1. Road Traffic Modeling

Similar to wireworld's modeling of electron movements, CA have been used to model road traffic. The simplest deterministic model used is Wolfram's rule 184 ECA to model unidirectional single-lane traffic. This model is rather limiting, since "there are only two possible kinematic wave speeds, i.e., +1 and -1 cell/time step" and the model does not allow for non-homogeneous vehicles (Maerivoet and De Moor, 2005). Rule-184 models have been extended further to move past these limitations, with one example of this being the Fukui & Ishibashi model (FI), which allows for "cars advancing by multiple sites per one time step" (Fukui and Ishibashi, 1996). The field of CA road traffic modeling has kept evolving, with more complex models, such as Brake-light, being used to produce more and more accurate and useful predictions (Maerivoet and De Moor, 2005).

4.2. Procedural Content Generation

Another usage for cellular automata is procedural content generation (PCG) for games. This approach relies on the self-organization capabilities of CA and allows for the generation of natural-looking terrain. The approach, which is taken when generating two-dimensional content is this:

- The map is initialised as an $N \times M$ grid of empty cells (floors).
- Using a uniform distribution random number generator, some of the cells are changed to filled (walls).
- A two-dimensional CA ruleset is applied for an X number of steps over the grid, which has the effect of reducing the random noise and forming larger cave-like structures.
- Optionally, smaller caves are filled in using a flood-fill algorithm.

In two dimensions "the computational effort equals $n(wh(2M + 1)^2)$, where w and h are the width and height of the grid, respectively, n is the number of CA iterations" and M is the Moore neighbourhood size, but the algorithm can be generalised for any number of dimensions by changing the neighbourhood type (Johnson, Yannakakis and Togelius, 2010). For an example of the resulting map, see figures 9.1 and 9.2.

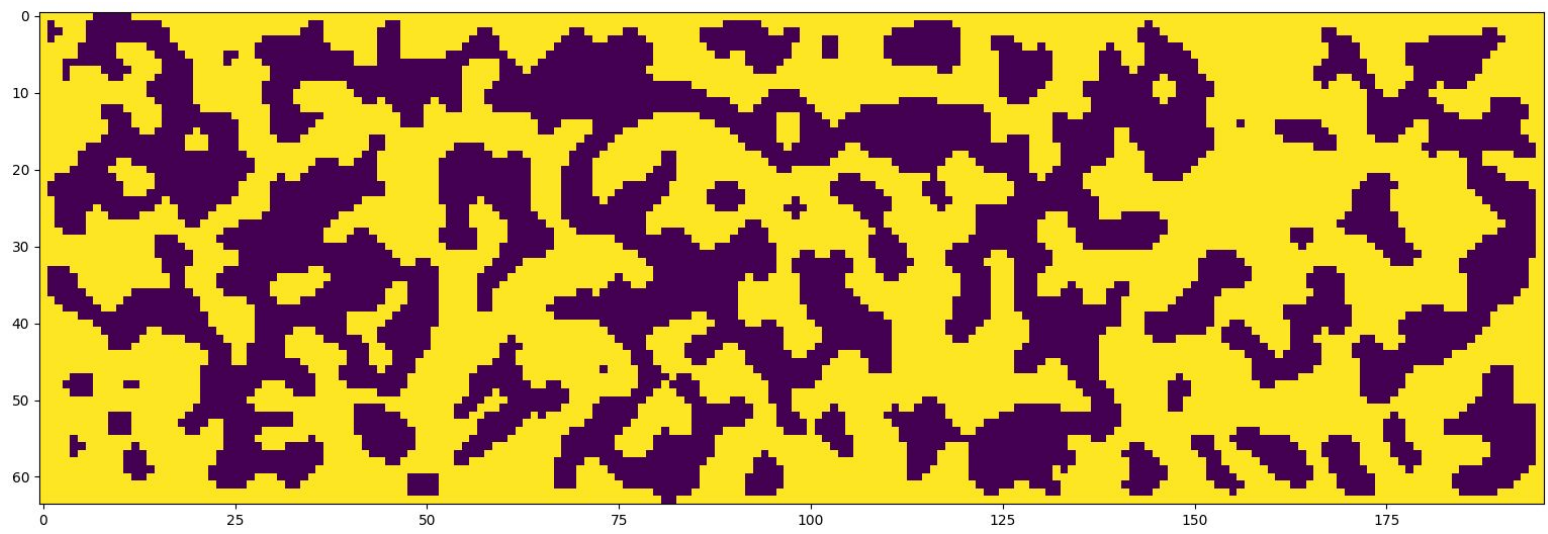


Figure 9.1: Randomly generated cave map using CA²

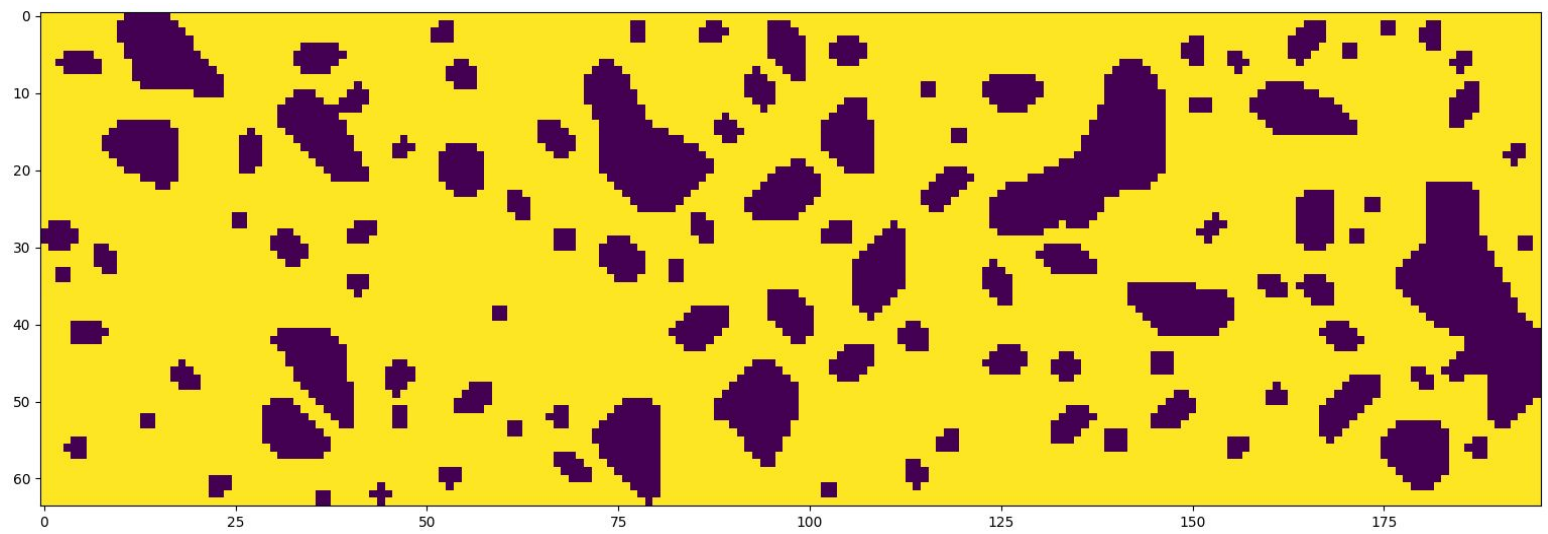


Figure 9.2: Randomly generated cave map using CA with a different ruleset¹

² These examples and the code for generating them is published at <https://github.com/PetarPeychev/automatagen>

5. Summary

Cellular automata have properties which make them viable for a variety of different applications, but are most well suited for simulating physical systems, due to the localisation of their update rule, which mimics the physical world. Cellular automata models are computationally efficient and allow for easy parallelisation. They have been used to successfully solve scientific and engineering problems in road traffic flow, fluid dynamics, cryptography, structural design, molecular biology, finance, etc. In road traffic modeling, elementary cellular automata simulations have been used to efficiently approximate single and multi-lane traffic flow, with some limitations caused by the simplicity and homogeneity of these systems. More complex cellular automata rulesets have been devised to handle these limitations, such as the Fukui & Ishibashi and Brake-light models. Cellular automata have also found a use in procedural content generation for games, due to their self-organization capabilities and computational practicality for creating natural-looking terrain.

6. References

- Cook, M. (2004). Universality in Elementary Cellular Automata. *Complex Systems*, vol. 15 (1), pp. 1-40.
- Dewdney, A. K. (1990). Computer Recreations. *Scientific American*, vol. 262, (1), pp 146-149.
- Egan, G. (1994). *Permutation City*. Millennium Orion Publishing Group. London.
- Fukui, M., & Ishibashi, Y. (1996). Traffic Flow in 1D Cellular Automaton Model Including Cars Moving with High Speed. *Journal of the Physical Society of Japan*, 65(6).
- Gardner, M. (1970) . The fantastic combinations of John Conway's new solitaire game "life". *Scientific American*, 223, pp. 120-123.
- Johnson, L.,Yannakakis, G. and Togelius, J. (2010). Cellular automata for real-time generation of infinite cave levels. DOI: 10.1145/1814256.1814266
- Maerivoet, S. and De Moor, B. (2005). Cellular automata models of road traffic. *Physics Reports* vol. 419 pp. 1 – 64.
- Owen, M. and Moore, D. (2004). The Wireworld Computer. [Online]. [Accessed 23 November 2018]. Available from: <https://www.quinapalus.com/wi-index.html>
- Rendell, P. (2016). Turing Machine in Conway Game of Life. *In Designing Beauty: The Art of Cellular Automata* pp. 149-154. Springer, Cham.
- Spezzano, G. and Talia, D. (1999). Programming cellular automata algorithms on parallel computers. *Future Generation Computer Systems*. Volume 16, Issues 2–3. pp. 203-216.
- von Neumann, J. (1951). The general and logical theory of automata. In L. A. Jeffress (Ed.), *Cerebral mechanisms in behavior; the Hixon Symposium* pp. 1-41. Oxford, England: Wiley.
- Wolfram, S. (1982). Cellular automata as simple selforganizing systems. *Caltech preprint*. pp. CALT-68-938.
- Wolfram, S. (1984). Universality and Complexity in Cellular Automata. *Physica D: Nonlinear Phenomena* 10, no. 1–2: pp. 1–35.
- Yuen, A.H. and Kay, R.H. (2009). Applications of Cellular Automata. Retrieved from <http://www.cs.bham.ac.uk/~rjh/courses/NatureInspiredDesign/2009-10/StudentWork/Group2/design-report.pdf> [Accessed 17 November, 2018]