1 Introduction

Cellular Automata (CAs) are discrete dynamical systems, having a long and illustrious history of study. Pioneering work in the field was done during the 1950s by the polymath John von Neumann, who was interested in their use for modelling self-reproduction. They have captured the popular imagination via the Game of Life, received extensive academic study into their fundamental characteristics and cabilities, and been applied succesfully to the modelling of natural phenomena.

2 Definition of Cellular Automata

The idea of Cellular Automata is intuitive and simple, and will be sketched informally first. Informally, a CA is a lattice of cells, each of which may be in a predetermined number of discrete states. A neighbourhood relation is defined over this lattice, indicating for each cell which cells are considered to be its neighbours during state updates. Time in the model is also discrete; on each time step, every cell updates its state using a transition rule that takes as input the states of all the cells in its neighbourhood[1]. All cells in the CA are updated synchronously. At time $t = 0$ the initial state of the CA must be defined, but then repeated synchronous application of the transition function to all cells in the lattice will lead to the deterministic evolution of the CA over time.

We can outline a formal definition as follows, for what it is worth. First let the (possibly infinite) set of all cells in the lattice be denoted C. Then let the neighbourhood function

$$n : C \times C \rightarrow \{True, False\}$$

return $True$ iff the input pair of cells in C are neighbours of each other[2]. For example, in a two-dimensional lattice one particular neighbourhood function $n(x, y)$ could be $True$ iff $x = y$ or y is one cell directly up, down, left, or right from y in the lattice. Then the neighbourhood of c_0 is given by

$$N_{c_0} = \{c \in C | n(c_0, c) = True\}.$$

Neighbourhood size $|N_{c0}|$ is typically independent of c_0. Next let the set of possible cell states be denoted S. Then the state transition function, to be applied synchronously to every element of C, is of the form

$$u : S^{|N|} \rightarrow S,$$

where X^y is the y-th Cartesian power of the set X. So the state transition function takes the states of all cells in the neighbourhood of a focal cell at time t, and specifies the state of that cell at time $t + 1$.

Many variations on the above theme exists. CAs can be of arbitrary dimension, although one-dimensional and two-dimensional CAs have received the most study. CAs can be infinite, or finite. Finite CAs can have periodic boundaries (e.g. the opposite ends of a one-dimensional finite CA are joined together so the whole forms a ring). Updates can be synchronous, or asynchronous. Transition rules can be deterministic or stochastic. Many other variations exist, but the above are some of the most typical ones.

[1] the neighbours of a cell are often taken to include the cell itself

[2] this relationship will normally be symmetric

3 Self-Reproduction

During the 1950s John von Neumann became interested in the problem of machines (or automata) that are able to produce copies of themselves. von Neumann started off considering how physical automata would need to be constructed, if they were able to construct copies of themselves. The details of such construction were very complicated, and so following discussions with Stanislaw Ulam, von Neumann began studying a logical model of self-reproducing automata[3]. We now refer to the paradigm von Neumann worked under as Cellular Automata. von Neumann's approach was to design a Turing Machine implemented as a cellular automaton using 29 states and the 5 cell (often referred to now as von Neumann) neighbourhood. von Neumann's work was edited and published posthumously by his colleague Arthur Burks [Von Neumann and Burks, 1966]. Subsequently others improved on the complexity of von Neumann's automaton. In particular Chris Langton, a pioneering figure in the field of Artificial Life, managed to get a much simpler reproducing automaton that occupied a fraction of the space and used 8 states. This was achieved by removing the requirement that the reproducing automaton be equivalent to a Turing Machine [Langton, 1984].

4 Behaviour

During the early 1980s Stephen Wolfram began a systematic investigation into the (almost) simplest CAs possible, one-dimensional CAs with neighbourhood size 3, and two possible states. The lattice in these 1-d CAs is a line, and cells are updated based on their own state and their immediate neighbours to the left and right. As the neighbourhood size is 3, and the number of states is 2, there are $2^{2^3} = 256$ such CAs. During exhaustive study of all these CAs, Wolfram conceived four main classes, with any given CA being a member of exactly one class [Wolfram, 1984].

4.1 Class I Cellular Automata

Class I CAs evolve[4] to a uniform configuration of cell states, from almost any initial configuration. This state can be thought of in dynamical systems terms as a 'point attractor', or 'limit point'. As one would suspect, the rules for class I CAs map from most or all possible neighbour configurations to the same new state. Initial lattice configurations do exist for some class I CAs that lead to non-trivial cycles, but these are very rare.

4.2 Class II Cellular Automata

CAs in Class II evolve to produce simple stable or periodic configurations on the lattice, according to the initial lattice configuration. Changes of cell state in the initial configuration will only affect final cell states that are nearby (in comparison to the neighbourhood size). Class II CAs can be thought of as 'filters' acting on the initial lattice configuration. The evolution of class II CAs to periodic configurations can be thought of as analagous to 'limit cycles' in dynamical systems terms.

4.3 Class III Cellular Automata

Class III CAs evolve to aperiodic, or chaotic, configurations from almost all initial lattice configurations. Over sufficient time, from almost all initial states the statistical properties of the resulting configuration, such as proportion of non-zero cells, converge to some value. Small changes in initial lattice configuration lead to larger and larger changes in resulting configuration as time progresses, as is the case for chaotic dynamical systems.

[3]interestingly, there has been a recent return to the problem of self-reproduction by physical machines [Zykov et al., 2005] [Griffith et al., 2005]

[4]here 'evolution' is used in the physics sense, of the unfolding of a process according to the laws governing it

4.4 Class IV Cellular Automata

CAs in Class IV exhibit propagating structures. In some sense Class IV is 'between' the purely chaotic behaviour of Class III, and the static behaviour of Class II. Some authors have made strong but vague arguments that complex systems are those 'poised at the edge of order and chaos' [Waldrop, 1993]. However there may be something in this view as, more concretely, some CAs in Class IV have been demonstrated to have a very special property, described in the next section.

5 Universality and Undecidability

Since the 1930s it has been known that all 'feasible' computations can be performed by an abstract computational model known as a Turing Machine, suitably programmed [Turing, 1936]. Almost all modern computing devices are Turing-equivalent. A Turing Machine that is able to read its program is known as a Universal Turing Machine. These were the Machines that John von Neumann simulated in his CA when inventing his self-reproducing automaton. Thus it has been known since the 1950s that CAs can perform any computation, although von Neumann's CAs were hugely complex. In the early 1970s it was shown that Conway's Game of Life, described in the next section, was also able to perform any computation [Wainwright, 1974]. This was achieved by identifying propagating structures, or configurations of cell states, in the Game of Life that could be used to emulate the components of a digital computer. In the early 1980s Stephen Wolfram speculated that a computationally complete 1-d Class IV CA could exist [Wolfram, 1984], and this was subsequently proved [Cook, 2004]. As with the Game of Life, this was shown by discovering propagating structures that could be arranged to interact with each other to carry out computations in the CA's lattice. The discovery that a CA with only 2 states and 8 transition rules could perform any computation was, and remains, truly remarkable. The universality of certain simple CAs leads to an important conclusion, as follows. Turing Machines can perform all 'feasible' computations, but there are some computations that are provably 'infeasible'. This means that there is provably no classical computational method to perform these computations, and they are typically referred to as uncomputable or undecidable. One such undecidable problem is to determine automatically if a Turing Machine with arbitrary input will halt its computation, or go on computing indefinitely. The analogue of this for CAs would be questions on their limiting behaviour such as whether they ever reach a stable configuration. The universality of certain CAs means that there is no general method for automatically determining such questions for arbitrary CAs with arbitrary initial conditions. There is a fundamental limit on our general predictive power for CAs, meaning that we can do no better than explicitly simulate their evolution up to some point in time in order to answer a question about their behaviour. As we clearly cannot simulate for infinite time, we cannot generally make predictions about infinite time behaviour, because it is undecidable. This provides an explicit motivation and justification for computer simulation of such complex systems, as analytic results are in general provably impossible. The fact that a simple 1-d CA with only 8 possible transitions can be universal and hence undecidable is should be a strong hint that simulation may be an important technique for examining the behaviour of many complex systems.

6 Applications

As well as the theoretical interest that the study of CAs engenders, CAs are a widely used computational tool for modelling complex systems. One well known example, previouslyy mentioned, is John Conway's 'Game of Life' [Gardner, 1970]. This is a simple 2-d 'totalistic'[5] CA with only two cell states, alive (on) and dead (off). Cells are updated according to how many of their 4 neighbours are alive; if fewer than two neighbours are alive, the cell dies through isolation, if more than three neighbours are alive, the cell dies through overcrowding, a live cell with two or three neighbours remains alive, and a dead cell with exactly three neighbours comes to life.

Other applications include, but are not limited to, theoretical biology [Boerlijst and Hogeweg, 1991], game theory [Nowak and May, 1992], and non-equilibrium thermodynamics [Markus and Hess, 1990].

[5] a CA in which state transitions are a function of the total (or average) state value over the cell's neighbourhood

References

[Boerlijst and Hogeweg, 1991] Boerlijst, M. and Hogeweg, P. (1991). Spiral wave structure in pre-biotic evolution: hypercycles stable against parasites. *Physica D*, 48(1):17–28.

[Cook, 2004] Cook, M. (2004). Universality in elementary cellular automata. *Complex Systems*, 15(1):1–40.

[Gardner, 1970] Gardner, M. (1970). Mathematical games: the fantastic combinations of John Conway's new solitaire game 'Life'. *Scientific American*, 223(4):120–123.

[Griffith et al., 2005] Griffith, S., Goldwater, D., and Jacobson, J. (2005). Self-replication from random parts. *Nature*, 437(7059):636.

[Langton, 1984] Langton, C. (1984). Self-reproduction in cellular automata. *Physica D: Nonlinear Phenomena*, 10(1-2):135–144.

[Markus and Hess, 1990] Markus, M. and Hess, B. (1990). Isotropic cellular automaton for modelling excitable media. *Nature*, 347(6288):56–58.

[Nowak and May, 1992] Nowak, M. and May, R. (1992). Evolutionary games and spatial chaos. *Nature*, 359(6398):826–829.

[Turing, 1936] Turing, A. (1936). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42(2):230–265.

[Von Neumann and Burks, 1966] Von Neumann, J. and Burks, A. (1966). *Theory of Self-Reproducing Automata*. University of Illinois Press Champaign, IL, USA.

[Wainwright, 1974] Wainwright, R. (1974). Life is universal! *Winter Simulation Conference: Proceedings of the 7 th conference on Winter simulation-*, 2:449–459.

[Waldrop, 1993] Waldrop, M. (1993). *Complexity: The Emerging Science at the Edge of Order and Chaos*. Simon & Schuster.

[Wolfram, 1984] Wolfram, S. (1984). Universality and complexity in cellular automata. *Physica D*, 10(1):1–35.

[Zykov et al., 2005] Zykov, V., Mytilinaios, E., Adams, B., and Lipson, H. (2005). Self-reproducing machines. *Nature*, 435(7039):163–164.