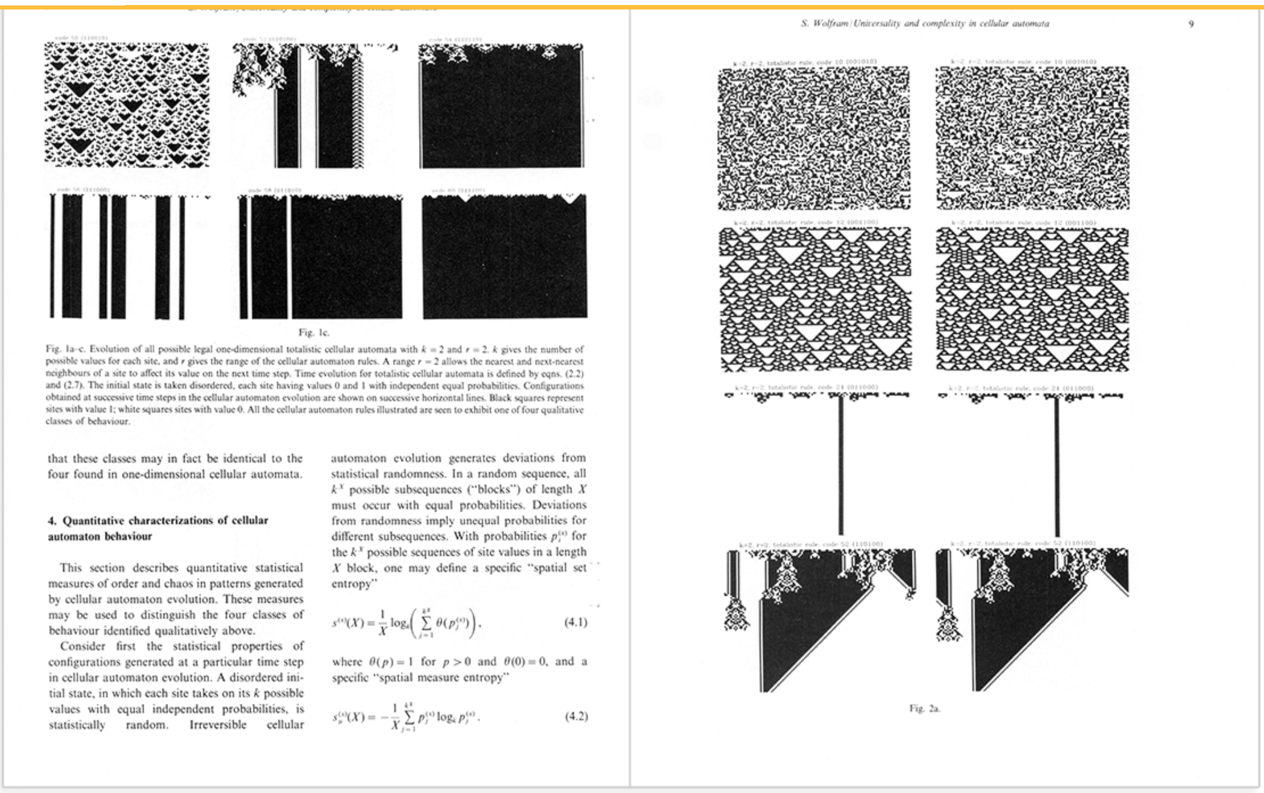


code 50 (110010)

code 52 (110100)

code 54 (110110)

code 56 (111000)

code 58 (111010)

code 60 (111100)

Fig. 1c.

Fig. 1a–c. Evolution of all possible legal one-dimensional totalistic cellular automata with $k = 2$ and $r = 2$. k gives the number of possible values for each site, and r gives the range of the cellular automaton rules. A range $r = 2$ allows the nearest and next-nearest neighbours of a site to affect its value on the next time step. Time evolution for totalistic cellular automata is defined by eqns. (2.2) and (2.7). The initial state is taken disordered, each site having values 0 and 1 with independent equal probabilities. Configurations obtained at successive time steps in the cellular automaton evolution are shown on successive horizontal lines. Black squares represent sites with value 1; white squares sites with value 0. All the cellular automaton rules illustrated are seen to exhibit one of four qualitative classes of behaviour.

that these classes may in fact be identical to the four found in one-dimensional cellular automata.

4. Quantitative characterizations of cellular automaton behaviour

This section describes quantitative statistical measures of order and chaos in patterns generated by cellular automaton evolution. These measures may be used to distinguish the four classes of behaviour identified qualitatively above.

Consider first the statistical properties of configurations generated at a particular time step in cellular automaton evolution. A disordered initial state, in which each site takes on its k possible values with equal independent probabilities, is statistically random. Irreversible cellular automaton evolution generates deviations from statistical randomness. In a random sequence, all k^X possible subsequences ("blocks") of length X must occur with equal probabilities. Deviations from randomness imply unequal probabilities for different subsequences. With probabilities $p_i^{(x)}$ for the k^X possible sequences of site values in a length X block, one may define a specific "spatial set entropy"

$$s^{(x)}(X) = \frac{1}{X} \log_k \left(\sum_{j=1}^{k^X} \theta(p_j^{(x)}) \right), \tag{4.1}$$

where $\theta(p) = 1$ for $p > 0$ and $\theta(0) = 0$, and a specific "spatial measure entropy"

$$s_\mu^{(x)}(X) = -\frac{1}{X} \sum_{j=1}^{k^X} p_j^{(x)} \log_k p_j^{(x)}. \tag{4.2}$$

S. Wolfram / Universality and complexity in cellular automata 9

k=2, r=2, totalistic rule, code 10 (001010)

k=2, r=2, totalistic rule, code 10 (001010)

k=2, r=2, totalistic rule, code 12 (001100)

k=2, r=2, totalistic rule, code 12 (001100)

k=2, r=2, totalistic rule, code 24 (011000)

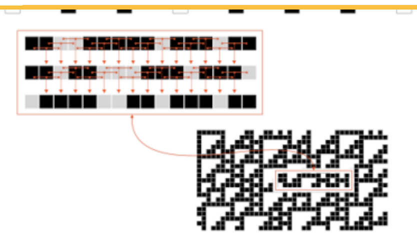
k=2, r=2, totalistic rule, code 24 (011000)

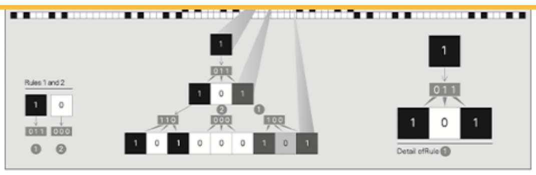
k=2, r=2, totalistic rule, code 52 (110100)

k=2, r=2, totalistic rule, code 52 (110100)

Fig. 2a.

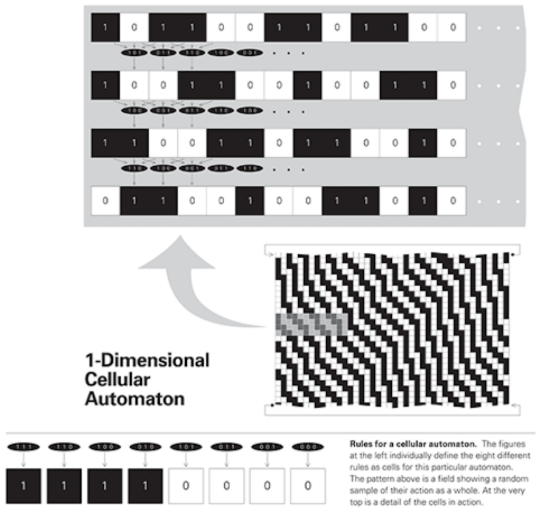
But what about "diagrams"? At first we toyed with drawing "textbook-style" diagrams—and produced some samples:



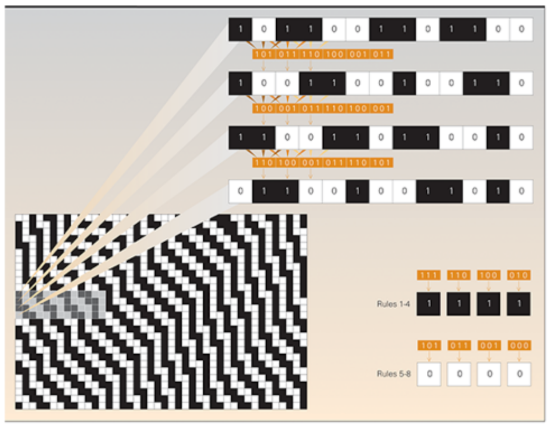


1- and 2-Dimensional Substitution Systems. The variegated pattern at the top of the illustration above is a field showing the beginning of a 1-dimensional substitution system as a whole. Very little of this system can actually be shown because of its exponential growth rate.

In the illustration below, the background image is a 2-dimensional system operating by similar rules. The rules defining the substitution processes in these sytems are shown in the left corners of each illustration. The enlarged details show the processes in action.

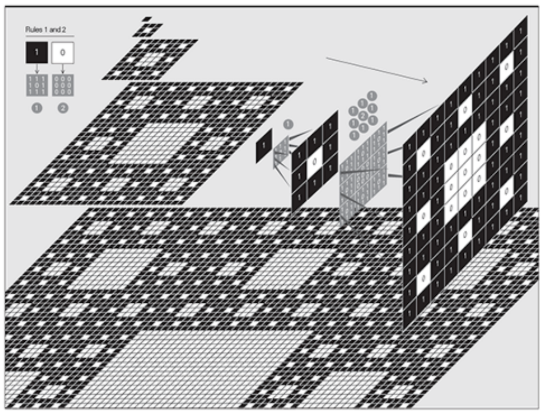


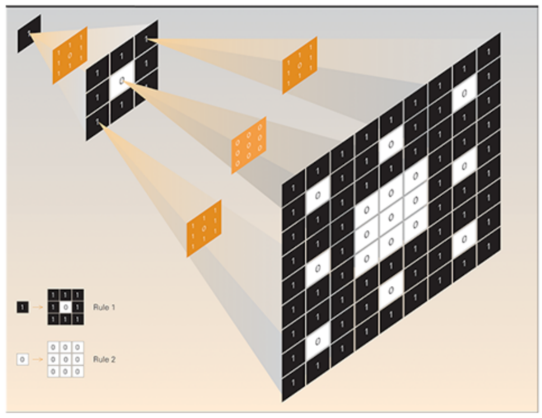
Rules for a cellular automaton. The figures at the left individually define the eight different rules as cells for this particular automaton. The pattern above is a field showing a random sample of their action as a whole. At the very top is a detail of the cells in action.



1-Dimensional Cellular Automaton. The figures at the top show 1-dimensional cellular automata in action. The variegated pattern is a field showing a random sample of their action as a whole. At the bottom right are the eight different rules that make up this particular automaton.

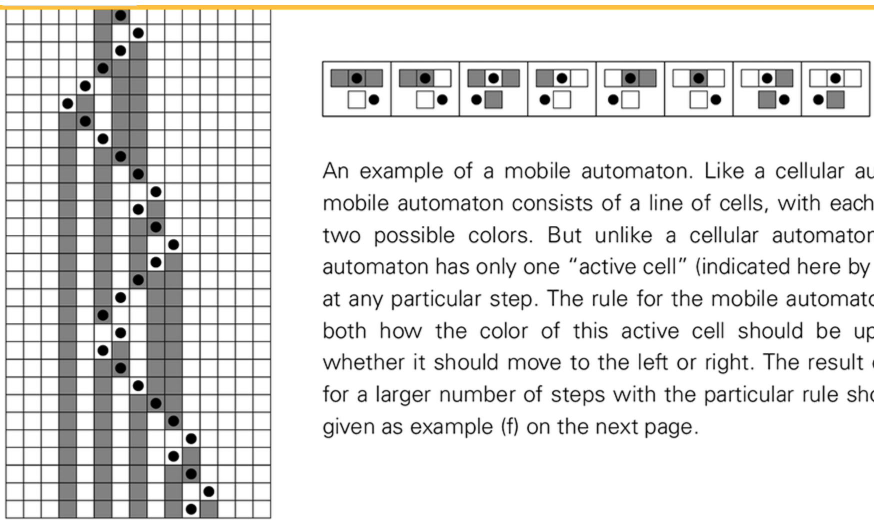
The rules in this example are typical of the systems we discuss throughout this chapter. Whatever else might need to be said can go here as a follow-up, maybe a reference to some notes.





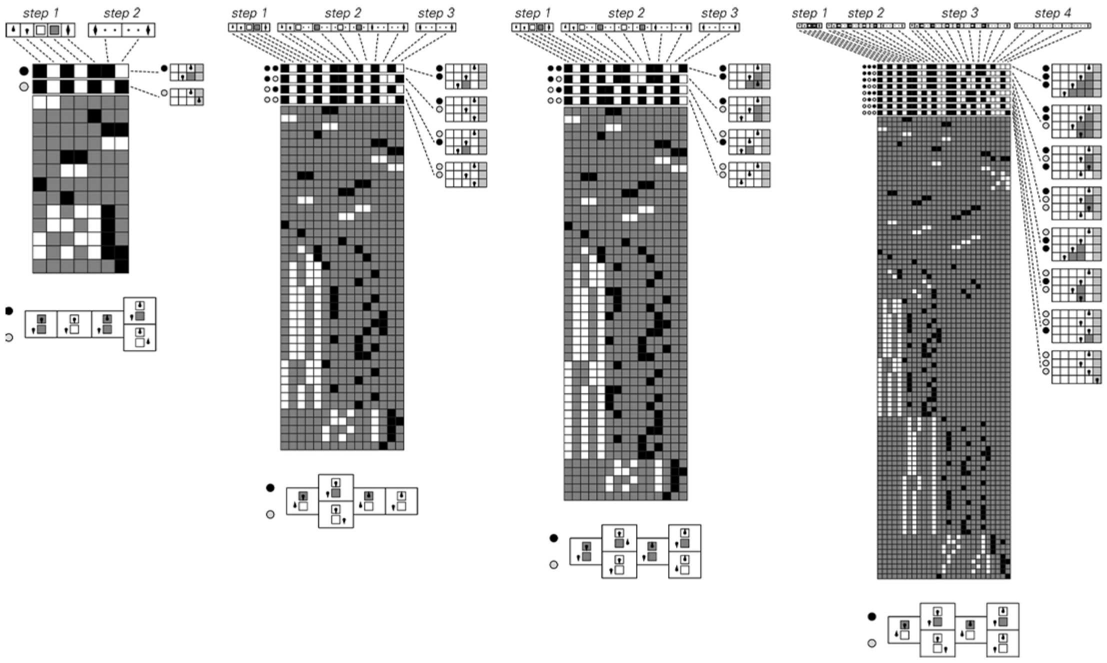
But these seemed to have way too much “conceptual baggage”, and when one looks closely at them, it’s easy to get confused. I wanted something more minimal—where the spotlight was as much as possible on the systems I was studying, not on “diagrammatic scaffolding”. And so I tried to develop a “direct diagramming” methodology, where each diagram could directly “explain itself”—and where every diagram would be readable “purely visually”, without words.

In a typical case I might show the behavior of a system (here a mobile automaton), next to an explicit “visual template” of how its rules operate. The idea then was that even a reader who didn’t understand the bigger story, or any of the technical details, could still “match up templates” and understand what was going on in a particular picture:



An example of a mobile automaton. Like a cellular automaton, a mobile automaton consists of a line of cells, with each cell having two possible colors. But unlike a cellular automaton, a mobile automaton has only one "active cell" (indicated here by a black dot) at any particular step. The rule for the mobile automaton specifies both how the color of this active cell should be updated, and whether it should move to the left or right. The result of evolution for a larger number of steps with the particular rule shown here is given as example (f) on the next page.

At the beginning of the project, the diagrams were comparatively simple. But as the project progressed I invented more and more mechanisms for them, until later in the project I was producing very complex "visually readable" diagrams like this:



A crucial point was that all these diagrams were being produced algorithmically—with Wolfram Language code. And in fact I was developing the diagrams as an integral part of actually doing the research for the book. It was a lesson I'd learned years earlier: don't wait until research is "finished" to figure out how to present it; work out the presentation as early as possible, so you can use it to help you actually do the research.

Another aspect of our first "textbook-like" style for the book was the idea of having additional elements, alongside the "main narrative" of the book. In early layouts we thought about having "Technical Notes", "Historical Notes", "Implementation Notes", etc. But it didn't take too long to decide that no, that was just going to be too complicated. So we made the decision to have one kind of note, and to collect all notes at the back of the book.