Minimal Cellular Automaton Approximations to Continuum Systems

1986

1. Introduction

The basic components of cellular automata are discrete. But at least in some cases the aggregate behaviour of large numbers of these components can be effectively continuous. As a result, it is possible to use cellular automata as models of continuum systems, such as fluids.

The mathematical origins of continuum behaviour in cellular automata are much the same as they are for many physical systems. A gas, for example, consists of many discrete molecules. Nevertheless, on a large scale, it can be described as a fluid.

Several conditions are necessary for the overall behaviour of a system with discrete elements to seem continuous.

First, continuum behaviour must be associated with some kind of extensive quantity. Such a quantity must be additive, and must be conserved in the dynamical evolution of the system. In a gas, one example of such a quantity is particle number. Other examples are energy and momentum.

A continuum system such as a fluid has the feature that its state can be described (locally) by just a few extensive quantities. To describe the precise microscopic state of a real gas one must, of course, specify the precise configuration of molecules. But it is believed that unless the gas is highly rarefied, this precise configuration is irrelevant to the macroscopic behaviour of the gas. Only the values of the few, averaged, extensive quantities are significant, so that a fluid approximation can be used.

The basis for this belief is embodied in the Second Law of thermodynamics. It seems that almost regardless of the initial microscopic configuration, collisions rapidly tend to randomize the configuration of gas molecules, so that at least for macroscopic purposes, it suffices to specify merely the values of certain average quantities.

Originally presented at Cellular Automata '86 (June 1986).

The true basis for this phenomenon has never been very clear. Some descriptions of it can be given in terms of the apparent increase of coarse-grained entropy. But no fundamental derivation has ever been given. The investigation of cellular automaton models seems likely to provide some new insights.

If microscopic randomization is assumed, then overall continuum behaviour can be derived using statistical mechanics. Based on master or transport equations, one can find partial differential equations satisfied by the densities of the extensive quantities conserved by the cellular automaton evolution.

Thus for example there has been much recent work on cellular automata which reproduce the Navier-Stokes equations for viscous fluid flow (see various other CA '86 posters).

Statistical mechanics, and the continuum equations derived from it, provide a considerably reduced description of the system. There may in fact be many systems with different detailed microscopic dynamics, which nevertheless yield identical large-scale statistical or continuum behaviour. Thus, for example, the Navier-Stokes equations describe the aggregrate behaviour of fluids such as air and water with very different microscopic constitutions.

Given generic macroscopic behaviour, it is important for both theoretical and practical purposes to try and find the simplest microscopic dynamics which can reproduce the macroscopic behaviour. One may, for example, seek the simplest cellular automaton rule which reproduces a particular form of continuum behaviour. ("Simplest" can be defined for example as requiring minimum storage space and minimum number of logical operations to implement.)

Specific rules which reproduce given macroscopic behaviour can conceivably be produced by explicit construction. Different elements of the rules can for example be arranged to mimic particular forms of particle collisions, and so on. The result of such a procedure will be some rule with the desired behaviour. But it will most likely not be the simplest such rule. Finding the simplest rule is in general a difficult optimization problem.

It is in some respects akin to problems such as logic circuit design in which a device with a particular form of overall behaviour must be constructed with the minimum number of circuit elements. Such problems have recently increasingly been tackled by iterative or adaptive procedures. Some dynamics in the space of possible circuits is defined, and the optimization process consists in applying this dynamics with certain constraints imposed.

Thus one can consider finding minimal cellular automaton rules by various iterative and adaptive procedures.

Such methods are examples of a general approach to computer programming and other design problems which one expects will become increasingly common. At present, most systems are designed in a step-by-step fashion, with their complete progression of states foreseen in detail by the designer. But more efficient designs may potentially be found by a more "goal-oriented" approach. Having specified the constraints, a definite adaptive or iterative procedure traverses the space of possible designs, seeking the one which optimizes some measure of success. The result will typically be a more efficient "computer-generated" design, whose operation cannot necessarily be "understood" in an explicit step-by-step fashion.

This poster considers as an example the problem of finding the simplest cellular automaton rule which reproduces the one-dimensional diffusion equation.

The potential interest of these investigations is severalfold.

1. They may provide practical methods for solving problems related to continuum systems (and these methods may be compared in detail with existing methods).
2. They provide examples of systems which exhibit the basic phenomena of thermodynamics, and should allow further elucidation of the foundations of thermodynamics.
3. They give examples of the procedure of "adaptive programming".

2. The Approach

The diffusion equation can be derived by considering the behaviour of the aggregate density of a large number of particles, each of which executes a random walk. The random walk may result from collisions with other particles of the same kind (as in self diffusion), or from interactions with some separate stochastic background.

The overall statistical behaviour of random walks is well known to be highly insensitive to the precise details of the walk. Thus for example walks whose steps are constrained to lie on various discrete lattices give in the large scale limit the same statistical behaviour as walks whose steps have no constraints.

By constructing a cellular automaton rule which involves various discrete particles, whose total number is conserved, one should thus be able to reproduce the diffusion equation.

A crucial issue, which relates to the foundations of thermodynamics, is the degree of randomness which is produced by a cellular automaton, or which, for that matter, is really necessary to reproduce macroscopic diffusion phenomena.

Nevertheless, following the approach discussed in the introduction, one is concerned not merely with finding some cellular automaton rule which reproduces diffusion, but rather with finding the simplest or optimal one. One must delineate a class of rules capable of reproducing diffusion, and then search within these to find the optimal one.

The conservation laws necessary for macroscopic diffusion turn out to be quite straightforward to ensure in a class of cellular automata. The capability for randomness generation cannot, it seems, be guaranteed directly by the structure of the rule, but must rather be deduced by studying the explicit behaviour of the system.

Diffusion requires that a scalar quantity (which in some cases can be identified as a particle number) is additively conserved.

In the simplest cellular automata, one considers rules which specify the new value of a single site in terms of the values of a neighbourhood of sites around it on the previous time step. In most such rules, no additive quantities can be conserved. In addition, such rules are usually highly irreversible, so that they evolve towards attractors which contain only a subset of the possible states. The accessibility of only a subset of states makes an adequate degree of randomness less likely. It does however necessarily preclude diffusion equation behaviour; the various statistical mechanical tools used in derivations can still be applied, but now not to all possible states, but only to those on the attractor.

There are several methods for constructing classes of cellular automata whose evolution satisfies certain conservation laws. (See Y. Pomeau "Invariant in cellular automata", J. Phys. A17 (1984) L415 and N. Margolus "Physics-like models of computation", Physica 10D (1984) 81, both reprinted in *Theory and Applications of Cellular Automata* (edited by S. Wolfram).) The method used here involves considering cellular automata which map one block of sites into another block of the same size.

In the simplest case, one considers a one-dimensional cellular automaton which maps pairs of binary site values to other pairs of binary values. The dynamics is chosen to be such that the boundaries of the pairs are taken to be at even and at odd sites on alternate time steps.

Figure 2.1 shows patterns generated by all the $4^4 = 256$ possible cellular automata of this kind. A variety of phenomena are observed.

Most of the cellular automata of this class show neither additive conservation laws nor reversibility. But unlike cellular automata whose rules are constructed in the usual way, the conditions for conservation and reversibility in these blocked cellular automata are comparatively simple to state.

The condition for reversibility is simply that the mapping from one set of blocks to another be a permutation (so that this mapping is invertible). (There are 24 such rules in the set shown in figure 2.1.)

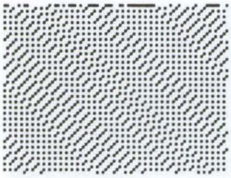
The condition for additive conservation laws is that for some values v_0 and v_1 the quantity $v_0 N_0 + v_1 N_1$ be conserved, where N_i is the number of sites with value i in each possible block.

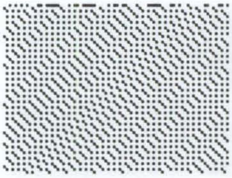
Table 2.1 gives the possible rules which satisfy this condition. Two are reversible; two are not. Inspection of figure 2.1 shows that in none of the cases is sufficient randomness generated.

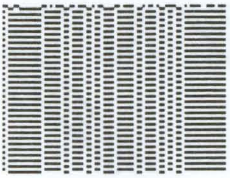
As a result, one must conclude that two possible values at each site ($k = 2$) and block size 2 ($b = 2$) are not sufficient to yield diffusion equation behaviour.

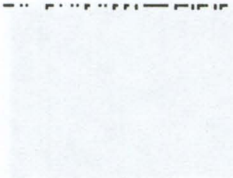
Figure 2.1. Patterns generated by evolution from disordered initial states according to all possible one-dimensional $k = 2$, $b = 2$ blocked cellular automaton rules. These rules have 2 possible values at each site. They are updated by mapping each block of two adjacent sites on to another block of two sites. On one "half step", blocks which begin on even-numbered sites are updated; on the other "half step", blocks beginning at odd-numbered sites are updated. The rules are numbered as follows. The output blocks for each of the possible input blocks 11, 10, 01 and 00 are written down in order. Then each output block is converted to a base 4 digit. The resulting base 4 number is then quoted in base 10. ▶

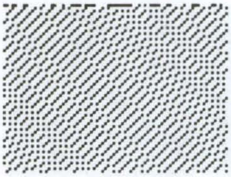
rule 0 (0000)

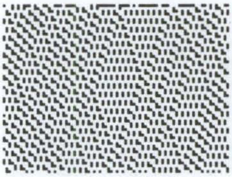

rule 1 (0001)

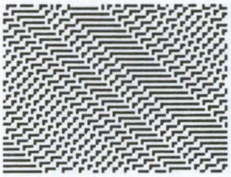

rule 2 (0002)

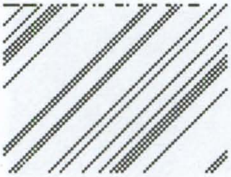

rule 3 (0003)

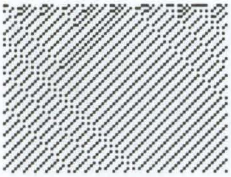

rule 4 (0010)

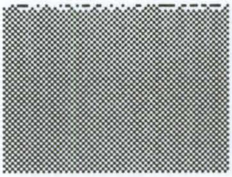

rule 5 (0011)

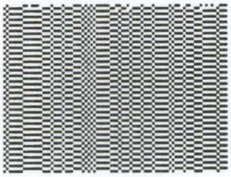

rule 6 (0012)

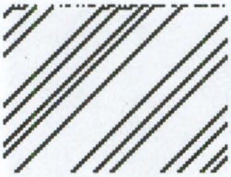

rule 7 (0013)

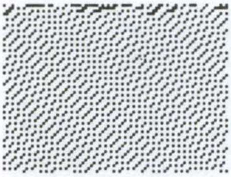

rule 8 (0020)

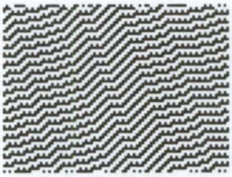

rule 9 (0021)

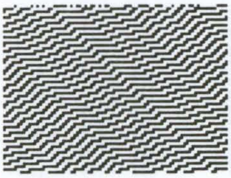

rule 10 (0022)

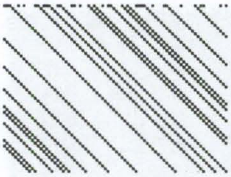

rule 11 (0023)


rule 12 (0030)

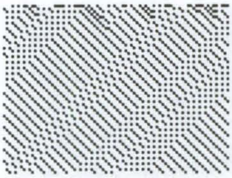

rule 13 (0031)

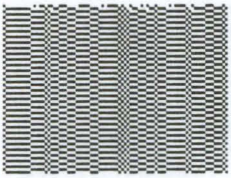

rule 14 (0032)

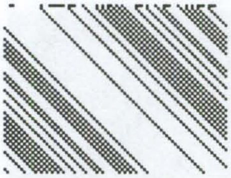

rule 15 (0033)

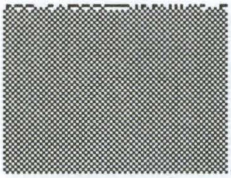

rule 16 (0100)

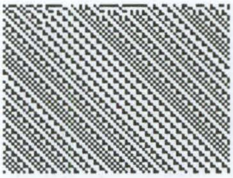
rule 17 (0101)

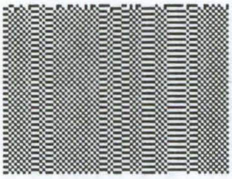

rule 18 (0102)

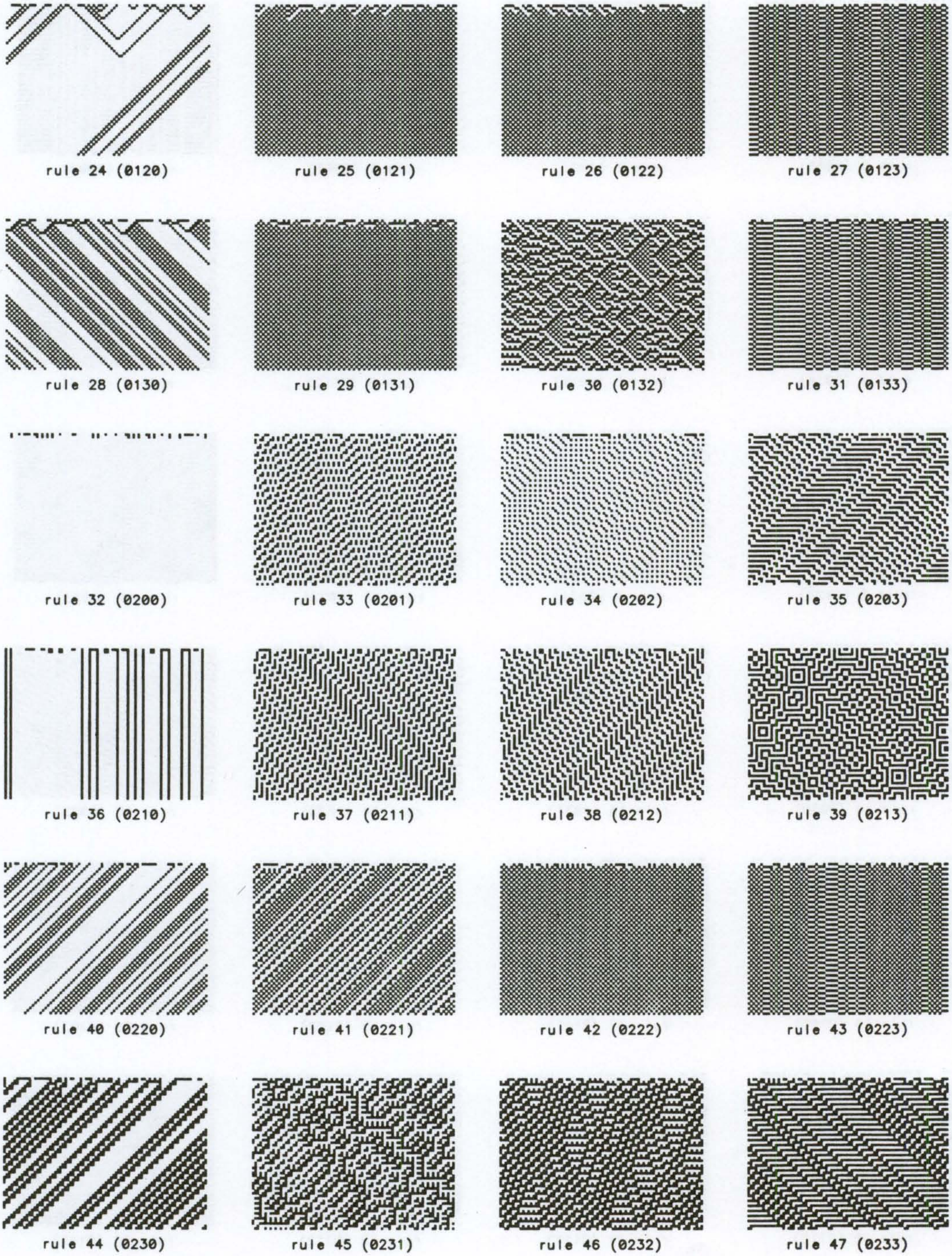

rule 19 (0103)


rule 20 (0110)


rule 21 (0111)


rule 22 (0112)


rule 23 (0113)



rule 24 (0120)

rule 25 (0121)

rule 26 (0122)

rule 27 (0123)

rule 28 (0130)

rule 29 (0131)

rule 30 (0132)

rule 31 (0133)

rule 32 (0200)

rule 33 (0201)

rule 34 (0202)

rule 35 (0203)

rule 36 (0210)

rule 37 (0211)

rule 38 (0212)

rule 39 (0213)

rule 40 (0220)

rule 41 (0221)

rule 42 (0222)

rule 43 (0223)

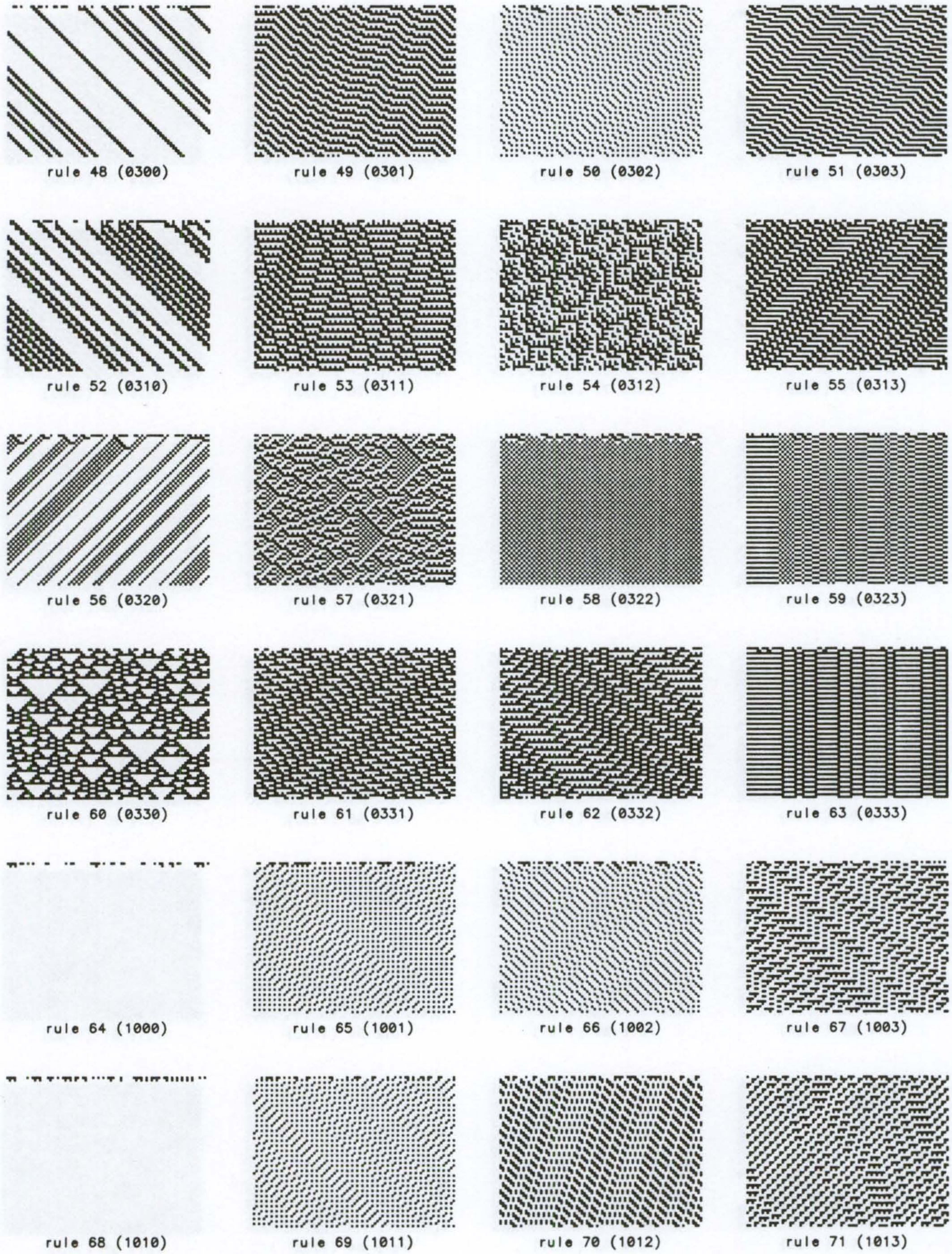
rule 44 (0230)

rule 45 (0231)

rule 46 (0232)

rule 47 (0233)

Figure 2.1 (continued).



rule 48 (0300)
rule 49 (0301)
rule 50 (0302)
rule 51 (0303)

rule 52 (0310)
rule 53 (0311)
rule 54 (0312)
rule 55 (0313)

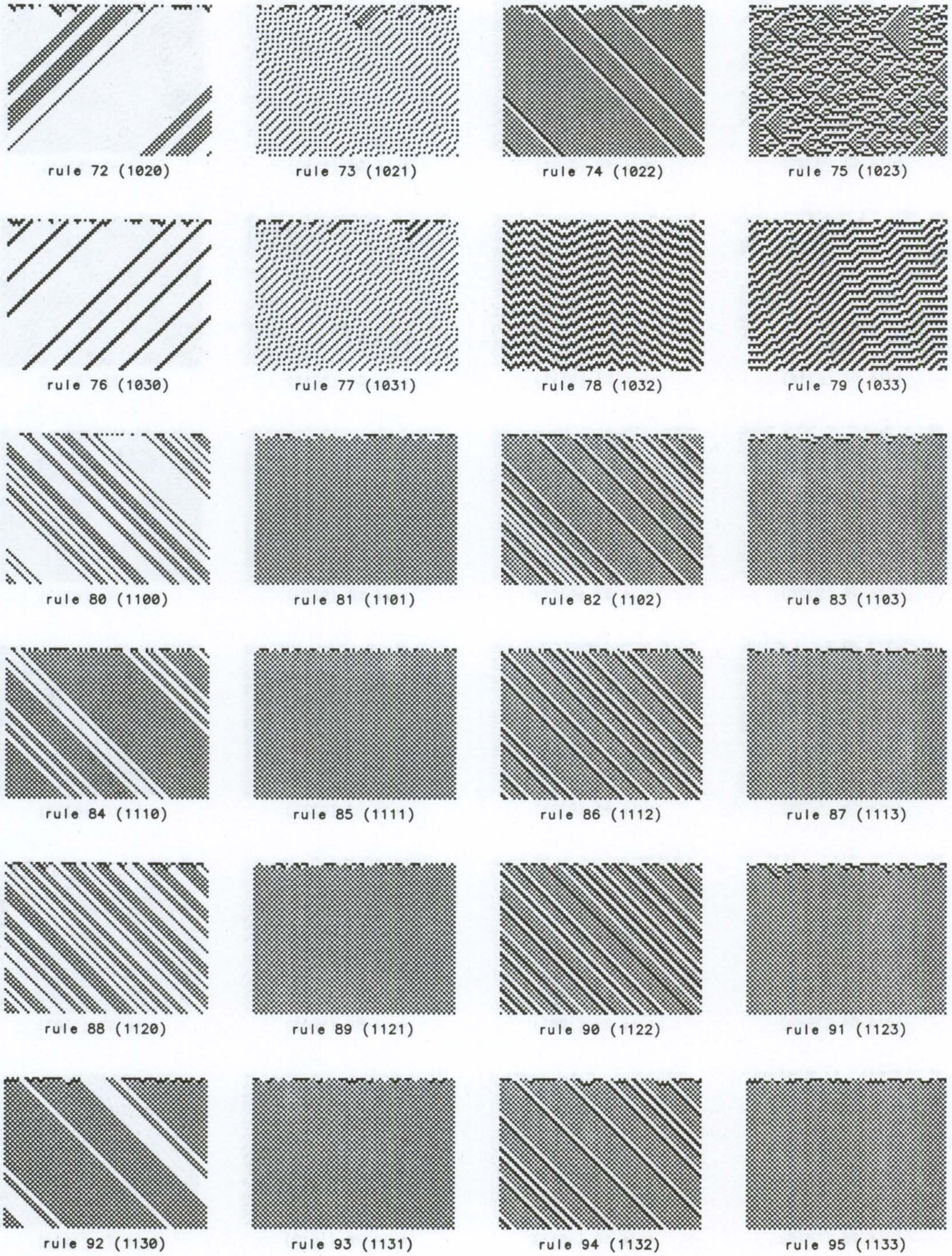
rule 56 (0320)
rule 57 (0321)
rule 58 (0322)
rule 59 (0323)

rule 60 (0330)
rule 61 (0331)
rule 62 (0332)
rule 63 (0333)

rule 64 (1000)
rule 65 (1001)
rule 66 (1002)
rule 67 (1003)

rule 68 (1010)
rule 69 (1011)
rule 70 (1012)
rule 71 (1013)

Figure 2.1 (continued).



rule 72 (1020)

rule 73 (1021)

rule 74 (1022)

rule 75 (1023)

rule 76 (1030)

rule 77 (1031)

rule 78 (1032)

rule 79 (1033)

rule 80 (1100)

rule 81 (1101)

rule 82 (1102)

rule 83 (1103)

rule 84 (1110)

rule 85 (1111)

rule 86 (1112)

rule 87 (1113)

rule 88 (1120)

rule 89 (1121)

rule 90 (1122)

rule 91 (1123)

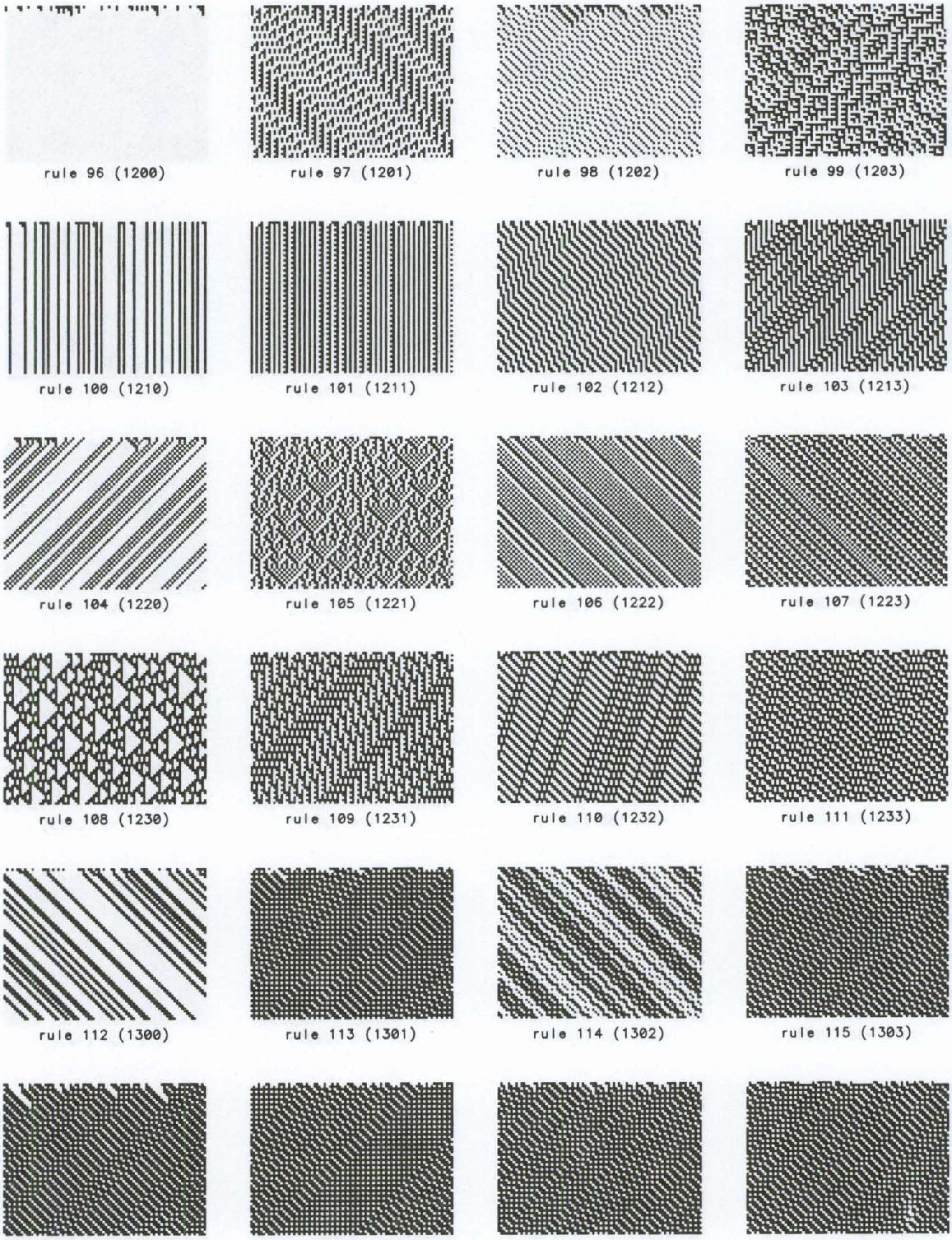
rule 92 (1130)

rule 93 (1131)

rule 94 (1132)

rule 95 (1133)

Figure 2.1 (continued).



rule 96 (1200)

rule 97 (1201)

rule 98 (1202)

rule 99 (1203)

rule 100 (1210)

rule 101 (1211)

rule 102 (1212)

rule 103 (1213)

rule 104 (1220)

rule 105 (1221)

rule 106 (1222)

rule 107 (1223)

rule 108 (1230)

rule 109 (1231)

rule 110 (1232)

rule 111 (1233)

rule 112 (1300)

rule 113 (1301)

rule 114 (1302)

rule 115 (1303)

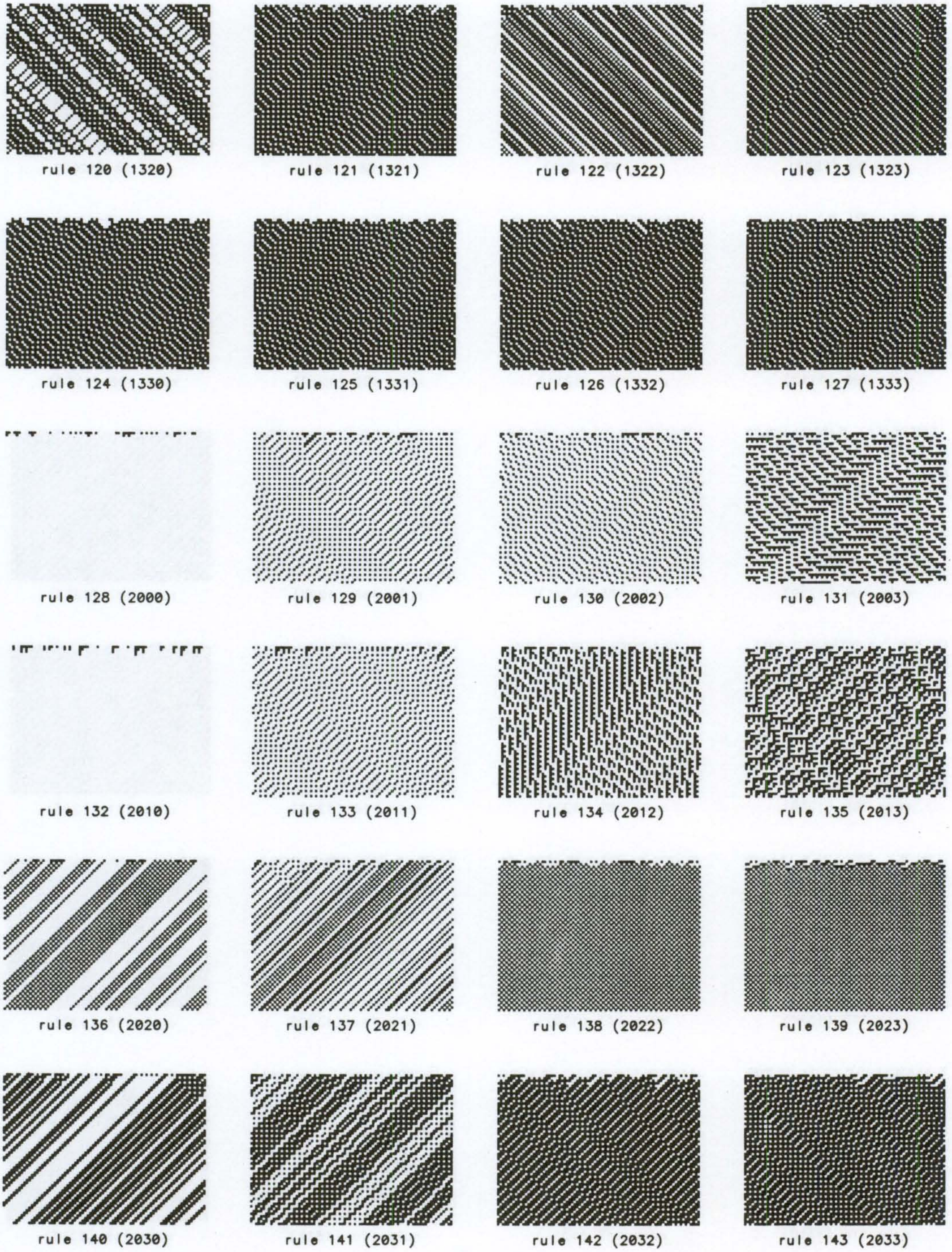
rule 116 (1310)

rule 117 (1311)

rule 118 (1312)

rule 119 (1313)

Figure 2.1 (continued).



rule 120 (1320)

rule 121 (1321)

rule 122 (1322)

rule 123 (1323)

rule 124 (1330)

rule 125 (1331)

rule 126 (1332)

rule 127 (1333)

rule 128 (2000)

rule 129 (2001)

rule 130 (2002)

rule 131 (2003)

rule 132 (2010)

rule 133 (2011)

rule 134 (2012)

rule 135 (2013)

rule 136 (2020)

rule 137 (2021)

rule 138 (2022)

rule 139 (2023)

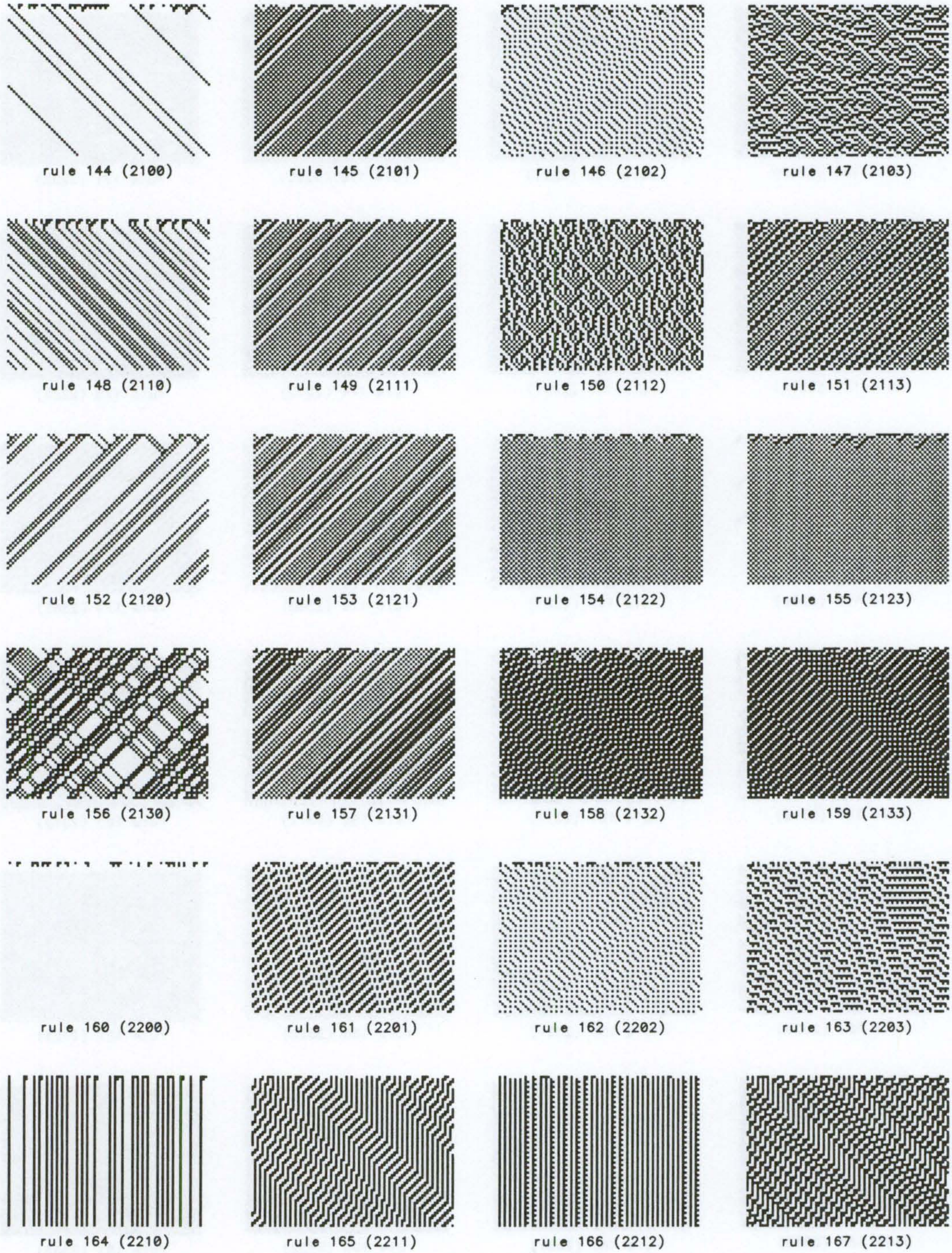
rule 140 (2030)

rule 141 (2031)

rule 142 (2032)

rule 143 (2033)

Figure 2.1 (continued).



rule 144 (2100)

rule 145 (2101)

rule 146 (2102)

rule 147 (2103)

rule 148 (2110)

rule 149 (2111)

rule 150 (2112)

rule 151 (2113)

rule 152 (2120)

rule 153 (2121)

rule 154 (2122)

rule 155 (2123)

rule 156 (2130)

rule 157 (2131)

rule 158 (2132)

rule 159 (2133)

rule 160 (2200)

rule 161 (2201)

rule 162 (2202)

rule 163 (2203)

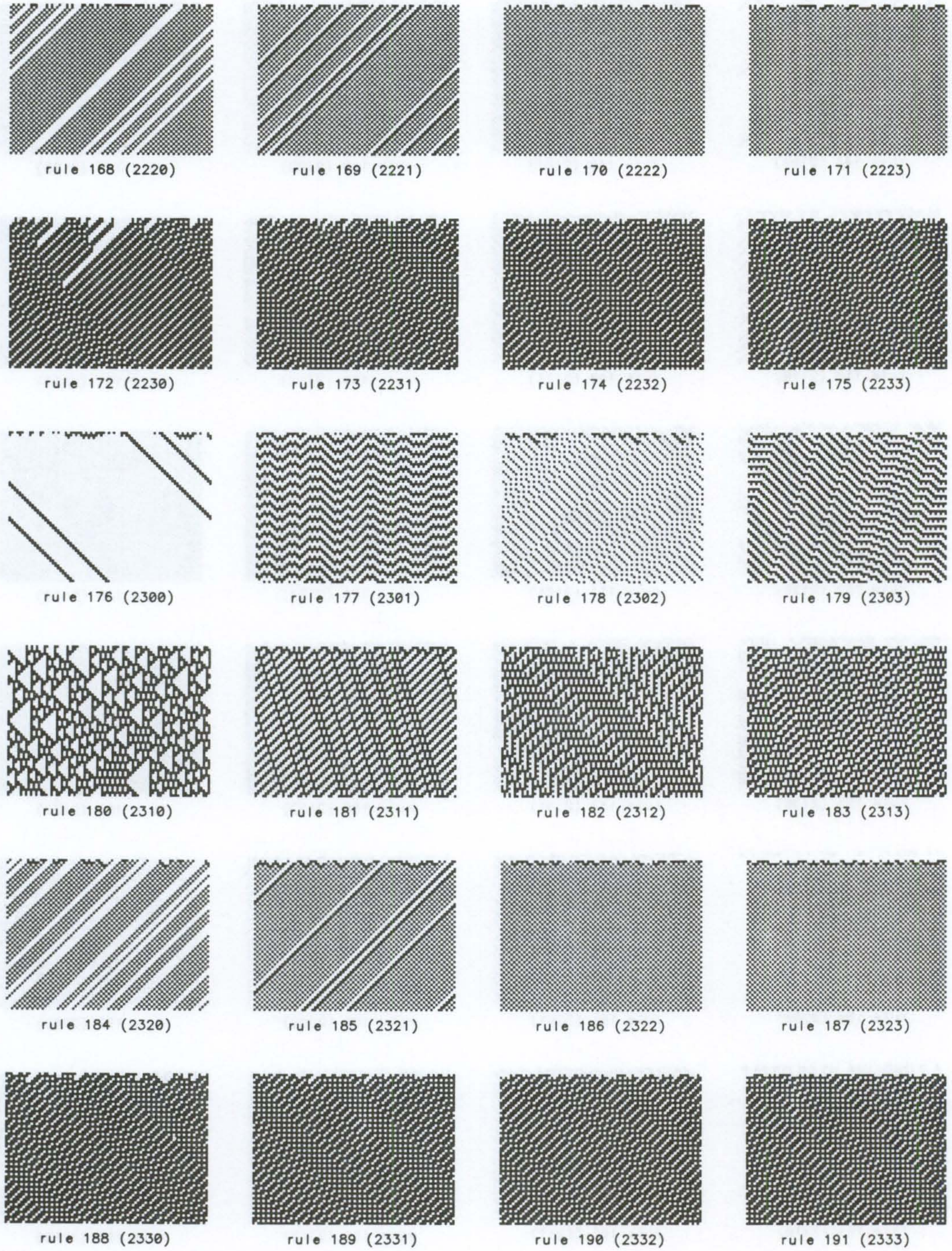
rule 164 (2210)

rule 165 (2211)

rule 166 (2212)

rule 167 (2213)

Figure 2.1 (continued).



rule 168 (2220)
rule 169 (2221)
rule 170 (2222)
rule 171 (2223)

rule 172 (2230)
rule 173 (2231)
rule 174 (2232)
rule 175 (2233)

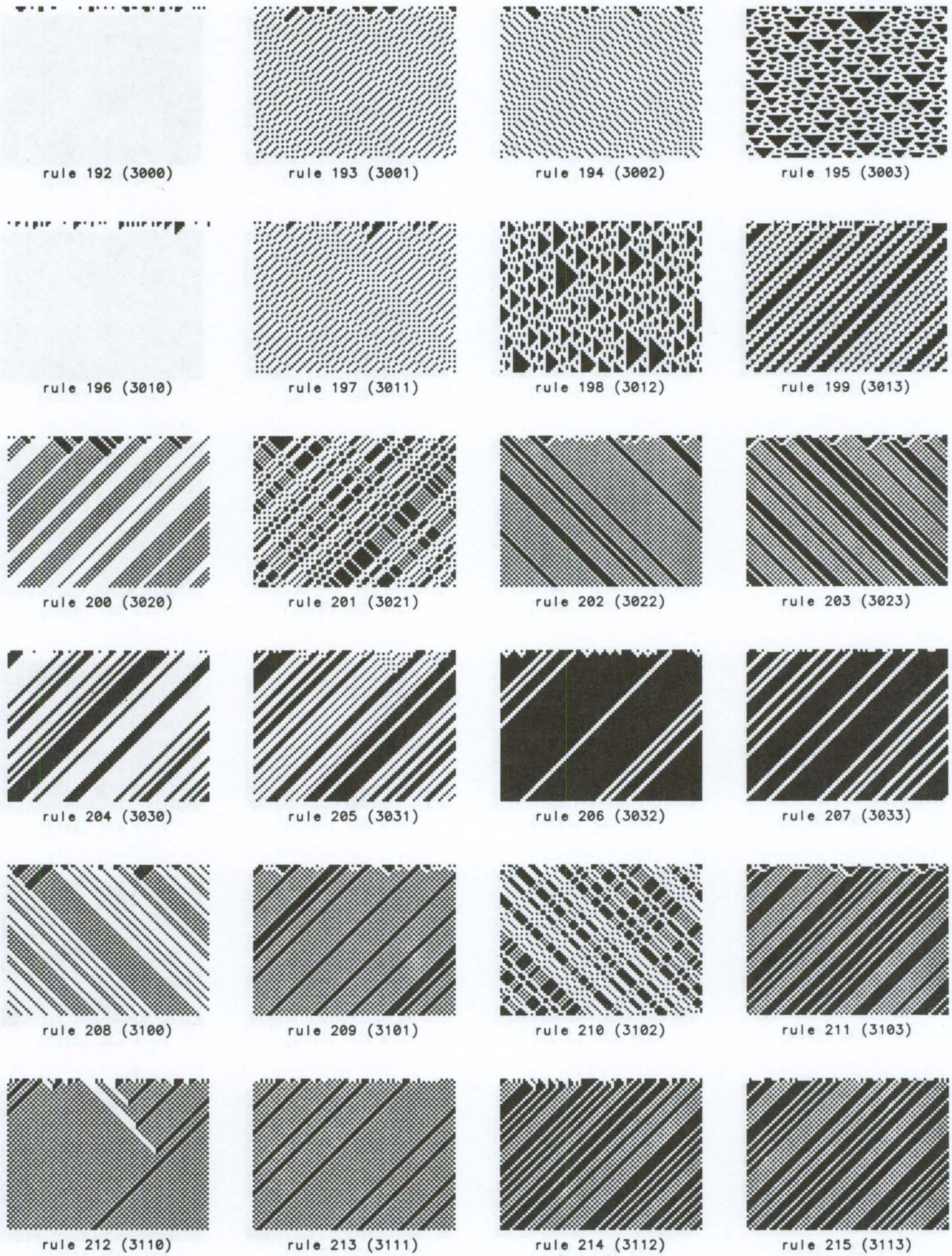
rule 176 (2300)
rule 177 (2301)
rule 178 (2302)
rule 179 (2303)

rule 180 (2310)
rule 181 (2311)
rule 182 (2312)
rule 183 (2313)

rule 184 (2320)
rule 185 (2321)
rule 186 (2322)
rule 187 (2323)

rule 188 (2330)
rule 189 (2331)
rule 190 (2332)
rule 191 (2333)

Figure 2.1 (continued).



rule 192 (3000)

rule 193 (3001)

rule 194 (3002)

rule 195 (3003)

rule 196 (3010)

rule 197 (3011)

rule 198 (3012)

rule 199 (3013)

rule 200 (3020)

rule 201 (3021)

rule 202 (3022)

rule 203 (3023)

rule 204 (3030)

rule 205 (3031)

rule 206 (3032)

rule 207 (3033)

rule 208 (3100)

rule 209 (3101)

rule 210 (3102)

rule 211 (3103)

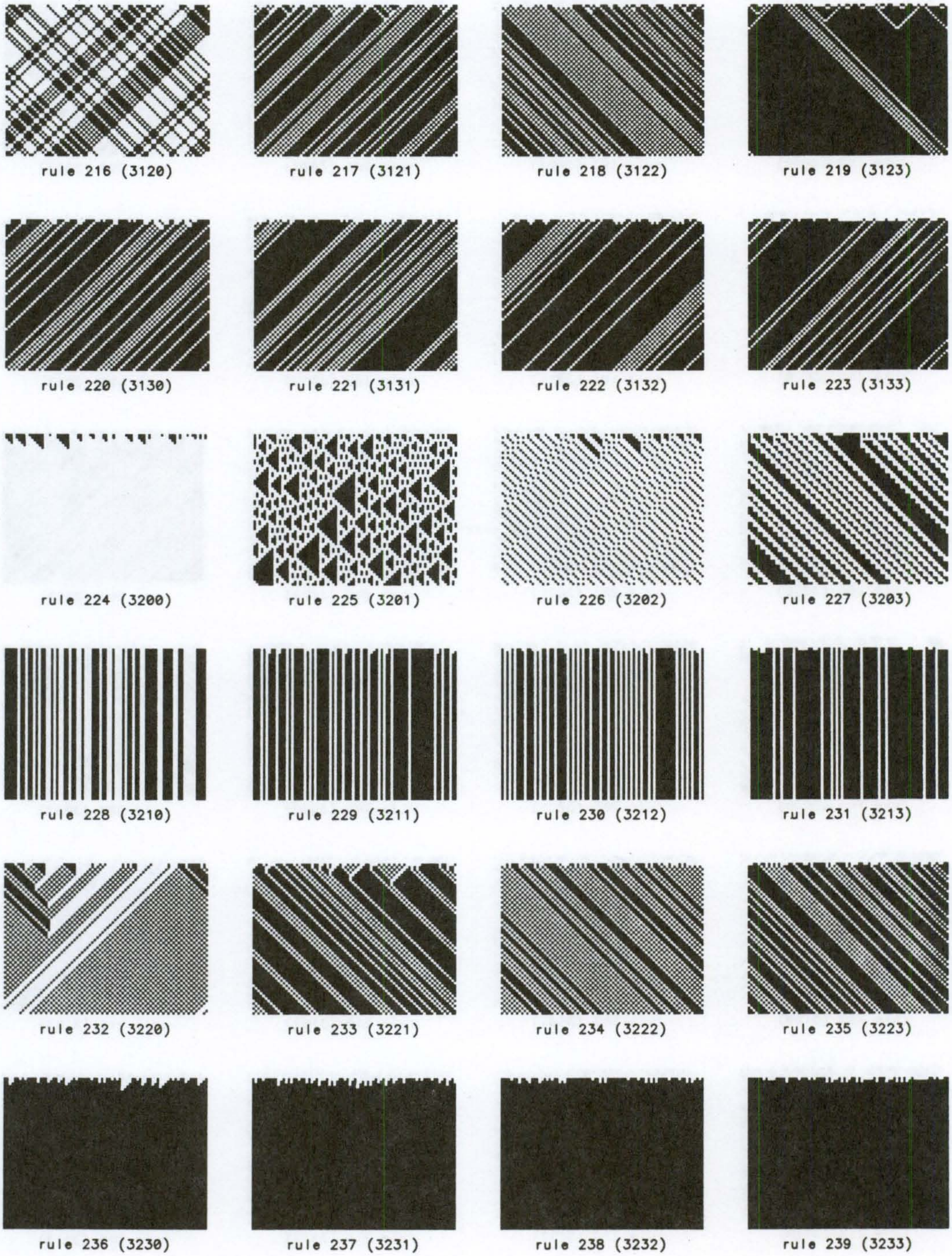
rule 212 (3110)

rule 213 (3111)

rule 214 (3112)

rule 215 (3113)

Figure 2.1 (continued).



rule 216 (3120)

rule 217 (3121)

rule 218 (3122)

rule 219 (3123)

rule 220 (3130)

rule 221 (3131)

rule 222 (3132)

rule 223 (3133)

rule 224 (3200)

rule 225 (3201)

rule 226 (3202)

rule 227 (3203)

rule 228 (3210)

rule 229 (3211)

rule 230 (3212)

rule 231 (3213)

rule 232 (3220)

rule 233 (3221)

rule 234 (3222)

rule 235 (3223)

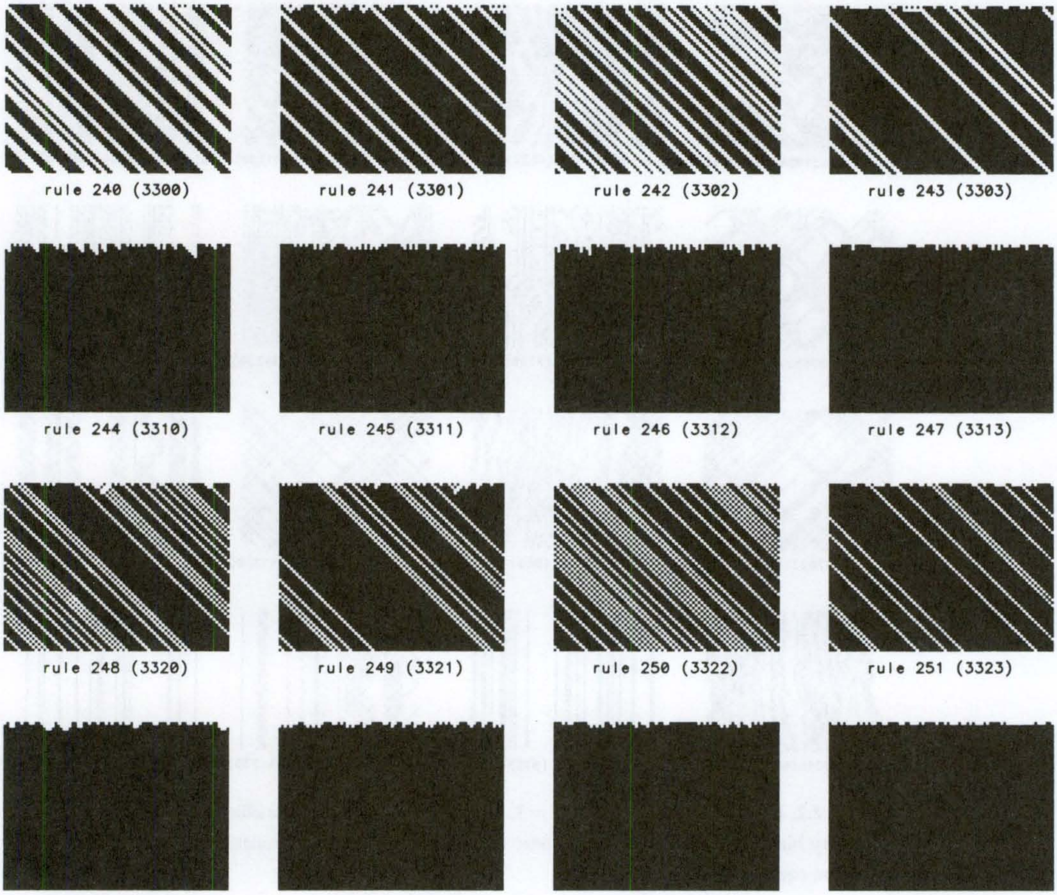
rule 236 (3230)

rule 237 (3231)

rule 238 (3232)

rule 239 (3233)

Figure 2.1 (continued).



rule 240 (3300)

rule 241 (3301)

rule 242 (3302)

rule 243 (3303)

rule 244 (3310)

rule 245 (3311)

rule 246 (3312)

rule 247 (3313)

rule 248 (3320)

rule 249 (3321)

rule 250 (3322)

rule 251 (3323)

rule 252 (3330)

rule 253 (3331)

rule 254 (3332)

rule 255 (3333)

Figure 2.1 (continued).

$n_0 + n_1$ constant	216	invertible
	228	identity
	232	
	212	
$n_0 + n_1$ mod 2 constant	27	invertible
	39	invertible
	23	
	43	

Table 2.1. $k = 2$, $b = 2$ block cellular automaton rules as illustrated in figure 2.1, with certain conservation laws relating to the total numbers n_i of sites with values i.

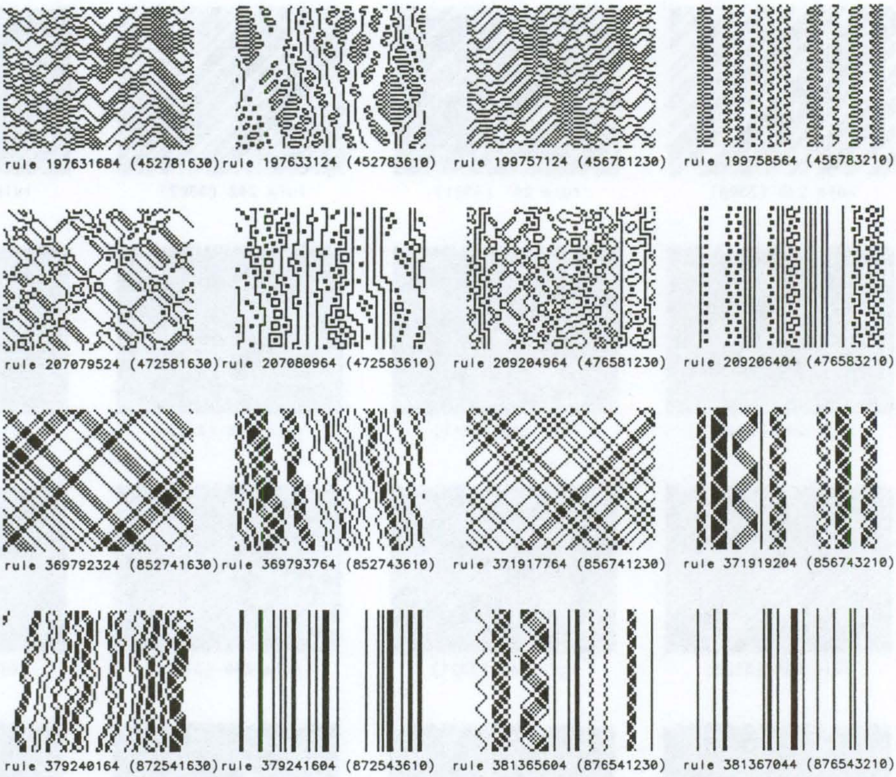

rule 197631684 (452781630)
rule 197633124 (452783610)
rule 199757124 (456781230)
rule 199758564 (456783210)
rule 207079524 (472581630)
rule 207080964 (472583610)
rule 209204964 (476581230)
rule 209206404 (476583210)
rule 369792324 (852741630)
rule 369793764 (852743610)
rule 371917764 (856741230)
rule 371919204 (856743210)
rule 379240164 (872541630)
rule 379241604 (872543610)
rule 381365604 (876541230)
rule 381367044 (876543210)

Figure 2.2. Patterns generated by all $k = 3$, $b = 2$ rules which are reversible and conserve the number of binary bits in each configuration. These rules are candidates for simulation of the one-dimensional diffusion equation.

It turns out that $k = 2$, $b = 3$ is also not sufficient.

As a result, one must consider $k = 3$, $b = 2$ rules. (A rough estimate of the "complexity" of rules can be obtained from the number of bits necessary, without compression, to specify their complete rule tables. This number is given by $\log_2\left((k^b)^{(k^b)}\right)$. It is slightly larger for $k = 3$, $b = 2$ than for $k = 2$, $b = 3$.)

With $k = 3$, there is slightly more freedom in the definition of additive quantities. One might, for example, consider adding the numerical values of sites. It turns out, again, that the set of rules with this quantity conserved is too highly constrained to allow a sufficient degree of randomness generation.

An alternative class of rules are those which conserve not the sum of the numerical values of sites, but the total number of binary bits contained in these values. There are 16 possible rules which satisfy this condition, and are reversible. Patterns generated by them are illustrated in figure 2.2.

Some of these rules obviously do not show sufficient randomness to yield diffusion behaviour. But others require more sophisticated analysis.

3. Randomization and Thermodynamics

It is observed that many systems, starting from almost any state, evolve rapidly to states which seem for practical purposes random. The sense in which the states are random is that their properties (say, statistical ones) are typical of the ensemble of all possible states. Several explanations and conditions for such randomness have been given. No complete understanding yet exists.

A common approach is based on ergodicity. Only reversible systems can be ergodic. The condition for ergodicity is that starting from any initial state, the evolution of the system eventually visits all possible states. The state transition diagram for the system thus consists of a single large cycle. If a system is ergodic, then at least after a sufficiently long time, it must evolve to an arbitrary, and thus "typical" state. In practice, however, the maximum period of time necessary to reach arbitrary states is usually astronomically large (it is typically exponential in the system size, and comparable to the recurrence time). Evolution for practical times reaches only some small subset of possible states.

What must now be explained is why these states seem random.

This is a subtle issue. There are always special choices of initial conditions for which the states reached are far from random. For example, one could choose initial conditions which are obtained from some orderly state by time reversal of the dynamics for some number of steps. These initial conditions would yield evolution which would not show degradation to randomness: rather it would suddenly yield orderly behaviour, seemingly violating the Second Law of thermodynamics.

One approach often taken is to consider the dependence of the evolution on small changes in initial conditions. It is supposed that the initial conditions cannot be determined precisely, so that in practice, measurements or experimental preparations can be guaranteed to yield only one of an ensemble of states, which differ slightly. The effects of small changes in initial conditions can be seen quite clearly in cellular automata.

One considers the evolution of a cellular automaton from two states, which differ say by a change in the value of a single site. The pattern of differences between states produced as a function of time shows the effect of this small initial perturbation. Figure 3.1 shows such difference patterns for the rules of figure 2.2. In some cases, initial changes remain localized; the evolution in such cases may be considered "stable". (Notice that in a reversible cellular automaton, the effects of changes in initial conditions can never die out completely, because information on the initial state must be preserved.) In other cases ("class 3" cellular automata), small initial changes are progressively amplified by the evolution. Change of the value of one site can ultimately affect the values of sites an arbitrary distance away. The patterns produced by such cellular automata can thus be considered unstable with respect to arbitrarily small perturbations.

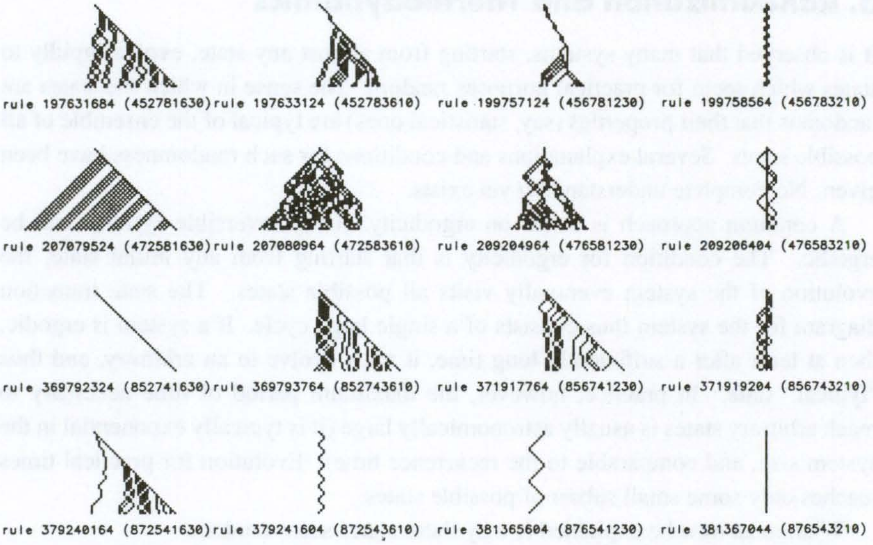

rule 197631684 (452781630)
rule 197633124 (452783610)
rule 199757124 (456781230)
rule 199758564 (456783210)
rule 207079524 (472581630)
rule 207080964 (472583610)
rule 209204964 (476581230)
rule 209206404 (476583210)
rule 369792324 (852741630)
rule 369793764 (852743610)
rule 371917764 (856741230)
rule 371919204 (856743210)
rule 379240164 (872541630)
rule 379241604 (872543610)
rule 381365604 (876541230)
rule 381367044 (876543210)

Figure 3.1. Difference patterns for the rules of figure 2.2. The patterns show the evolution of the difference between two random configurations which initially differ just by a change in a single bit. Some rules are seen to be stable under such perturbations; for other rules, the effect of these changes grows with time. The rate of growth gives the Lyapunov exponent of dynamical systems theory. Such instability leads to a sensitive dependence on the initial conditions for the evolution.

This phenomenon is central to much of what has been studied in the theory of chaotic dynamical systems. It implies that with incomplete knowledge of initial conditions, a time must ultimately come at which the results of measurements can no longer be predicted, because they depend on unknown features of the initial conditions.

It is not however guaranteed that the system will at this point be random. Its randomness depends on the randomness of the unknown features of the initial conditions. It is by no means clear in fact that in actual experiments, these features are indeed adequately random. Certainly one can consider cases in which for example only a few cellular automaton sites are nonzero, and all sites beyond some point are zero. In this case, randomness in final configurations cannot be directly attributed to random unknown data in the initial conditions.

A further, related, problem is the exact definition of "apparently random" states. A sequence or configuration is commonly considered "random" if no pattern can be discerned in it, so that no procedure can be used to predict additional elements of it, or to compress the information associated with it. The meaning of randomness depends on the kinds of pattern recognition which are considered.

If one starts with an orderly initial state, all states generated with time can be specified by giving this state, and the number of steps required to generate them. Such a specification will usually represent a substantial compression in the information associated with the state. Yet despite the possibility for such compression, many aspects of the state may still seem random. Although compression is possible, it may not be revealed by the kinds of statistical procedures commonly used to analyse the states.

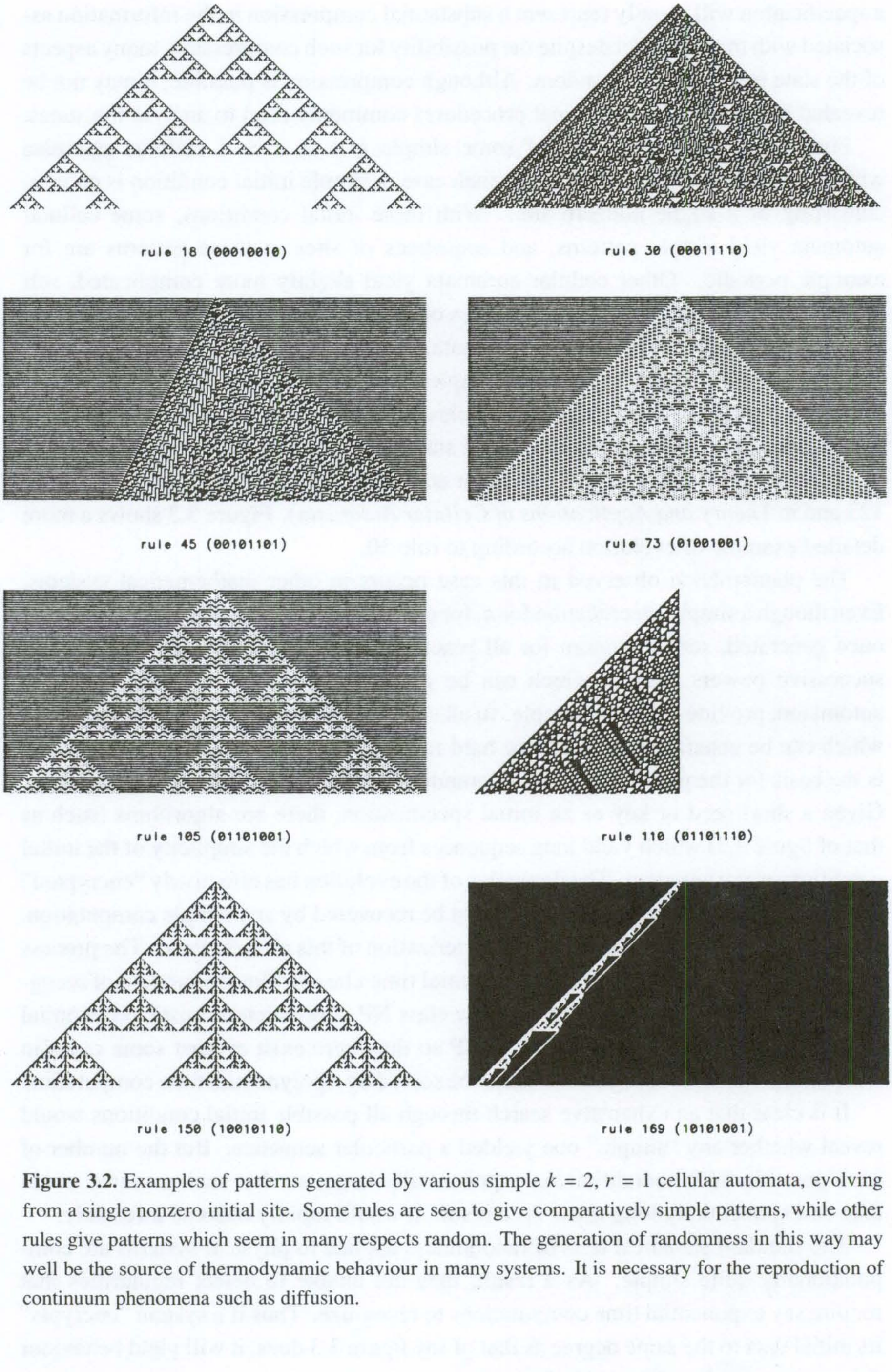
Figure 3.2 shows examples of some simple $k = 2$, $r = 1$ cellular automata which illustrate this phenomenon. In each case, a simple initial condition is chosen, consisting of a single nonzero site. With these initial conditions, some cellular automata yield simple patterns, and sequences of sites in these patterns are for example periodic. Other cellular automata yield slightly more complicated, self similar, patterns. But here again sequences of site values are almost periodic, and are readily predictable. Some cellular automata, however, can yield apparently random sequences even starting from these simple initial conditions. The two simplest examples (found by explicit search) are rules 30 and 45. In both cases, the sequences generated seem random according to all standard statistical tests (see S. Wolfram, "Random sequence generation by cellular automata", Adv. Applied Math. 7 (1986) 123 and in *Theory and Applications of Cellular Automata*). Figure 3.3 shows a more detailed example of evolution according to rule 30.

The phenomenon observed in this case occurs in other mathematical systems. Even though a simple specification for π, for example, can be given, its digit sequence, once generated, seems random for all practical purposes. The fractional parts of successive powers of $3/2$, which can be generated by a $k = 6$, $r = 1$ cellular automaton, provide another example. In all cases, what is observed is that a sequence which can be generated easily can be hard to invert or compress. This phenomenon is the basis for the possibility of pseudorandom number generation or cryptography. Given a short seed or key as an initial specification, there are algorithms (such as that of figure 3.3) which yield long sequences from which the simplicity of the initial conditions is not apparent. The dynamics of the evolution has effectively "encrypted" the initial data to the point where it cannot be recovered by any simple computation.

Computation theory provides a characterization of this phenomenon. The process of generating a sequence is in the polynomial time class P. But the process of recognizing the origins of the sequence is in the class NP of non-deterministic polynomial time computations. It seems that P ≠ NP so that there exist at least some cases in which the problem of recognition cannot be solved by a polynomial time computation.

It is clear that an exhaustive search through all possible initial conditions would reveal whether any "simple" one yielded a particular sequence. But the number of such possible initial conditions is exponentially large, so that such a search could take an exponentially long time. As a result, it would rapidly become infeasible.

The standard statistical tests of randomness applied to physical systems are computationally quite simple. As a result, they are unable to detect regularities that require say exponential time computations to recognize. Thus if a system "encrypts" its initial data to the same degree as that of say figure 3.3 does, it will yield behaviour that appears random for practical purposes.



rule 18 (00010010)

rule 30 (00011110)

rule 45 (00101101)

rule 73 (01001001)

rule 105 (01101001)

rule 110 (01101110)

rule 150 (10010110)

rule 169 (10101001)

Figure 3.2. Examples of patterns generated by various simple $k = 2$, $r = 1$ cellular automata, evolving from a single nonzero initial site. Some rules are seen to give comparatively simple patterns, while other rules give patterns which seem in many respects random. The generation of randomness in this way may well be the source of thermodynamic behaviour in many systems. It is necessary for the reproduction of continuum phenomena such as diffusion.

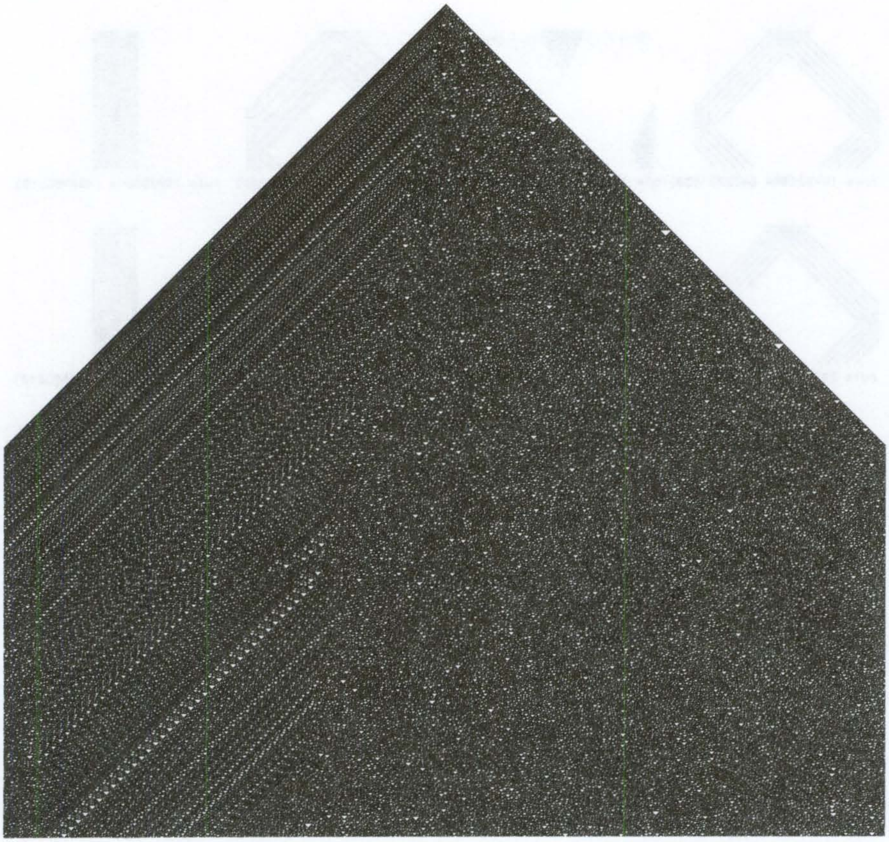


Figure 3.3. The pattern generated by $k = 2$, $r = 1$ rule 30 starting from a single site initial seed. This rule has the form

$$a_i' = (a_{i-1} + a_i + a_{i+1} + a_i a_{i+1}) \bmod 2.$$

Despite the simplicity of this rule, the patterns it generates are so complicated as to seem in many respects random. Thus for example the centre column in this picture seems random for at least a million sites according to standard statistical randomness tests. This rule is probably the simplest cellular automaton which generates random behaviour in this way. It was found by an explicit search over all possible rules.

I believe that most of the randomization associated with thermodynamic behaviour is of the mathematical type illustrated in figure 3.3. Even though the initial conditions are simple, the system encrypts them to the point where no feasible measurements or computations can recover them.

4. The Winning Rule

The phenomenon of randomization from simple initial conditions occurs in some but not all of the candidate diffusion equation cellular automata of figure 2.2. Figure 4.1

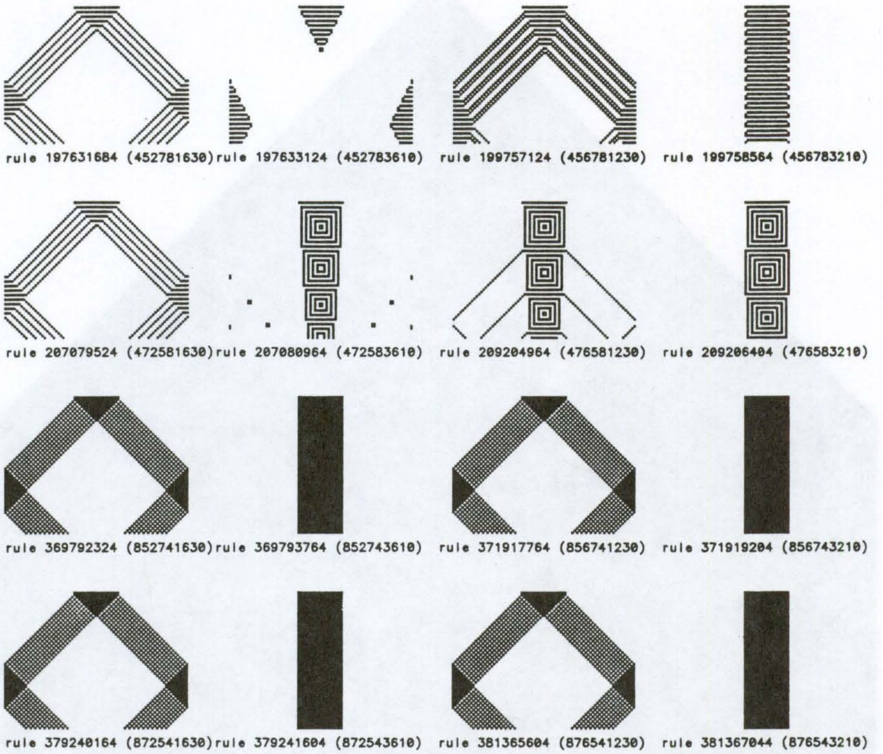

rule 197631684 (452781630)
rule 197633124 (452783610)
rule 199757124 (456781230)
rule 199758564 (456783210)
rule 207079524 (472581630)
rule 207080964 (472583610)
rule 209204964 (476581230)
rule 209206404 (476583210)
rule 369792324 (852741630)
rule 369793764 (852743610)
rule 371917764 (856741230)
rule 371919204 (856743210)
rule 379240164 (872541630)
rule 379241604 (872543610)
rule 381365604 (876541230)
rule 381367044 (876543210)

Figure 4.1. Patterns generated by the $k = 3$, $b = 2$ blocked cellular automata of figure 2.2, starting from a simple initial condition. The cellular automata are shown on a size 80 lattice with periodic boundary conditions. The initial condition consists of a block of 20 sites with value 1 in the centre of the system. Most of the rules are seen to give rise to simple periodic patterns.

shows evolution from a simple initial condition for all of these rules. Only one rule, and its (2,1) conjugate, show randomization in this case. Figure 4.2 shows the longer time evolution of this rule, on a size 80 with periodic boundary conditions. Regularities are still seen, but many features seem random.

The degree of randomness generated by this rule can be tested by applying certain statistical procedures. A simple one is the computation of coarse-grained entropy. Figure 4.3 shows the coarse-grained entropy for the system. It is seen to tend rapidly to a maximum value, as expected for an apparently random system.

Table 4.1 gives the block transformations for this rule. Interpretations in terms of particles and so on can be given. But it is noteworthy that making the rule "increasingly mixing" by including transitions for various other blocks does not yield an increase in the randomness of the overall behaviour. In fact, as figures 2.2 and 4.1 show, such "additional mixing" usually leads to simpler overall behaviour.

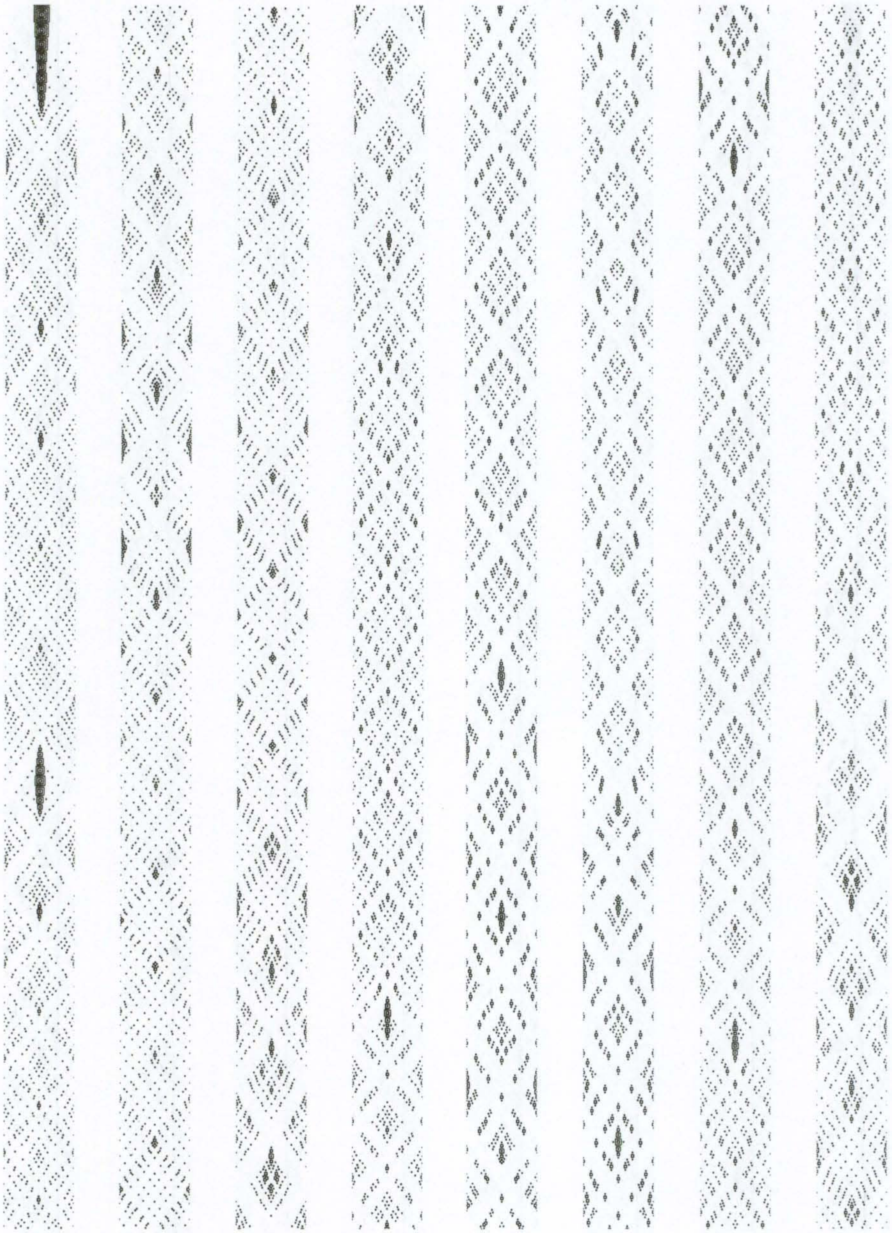


Figure 4.2. Longer sequences generated by one rule from figure 4.1 which seems to generate randomness from simple initial conditions. The patterns on this page were made on a size 80 lattice, with a size 20 initial block. The patterns on the next page were made with a size 21 initial block. The degradation of orderly initial conditions into apparent randomness is clearly visible in these pictures.

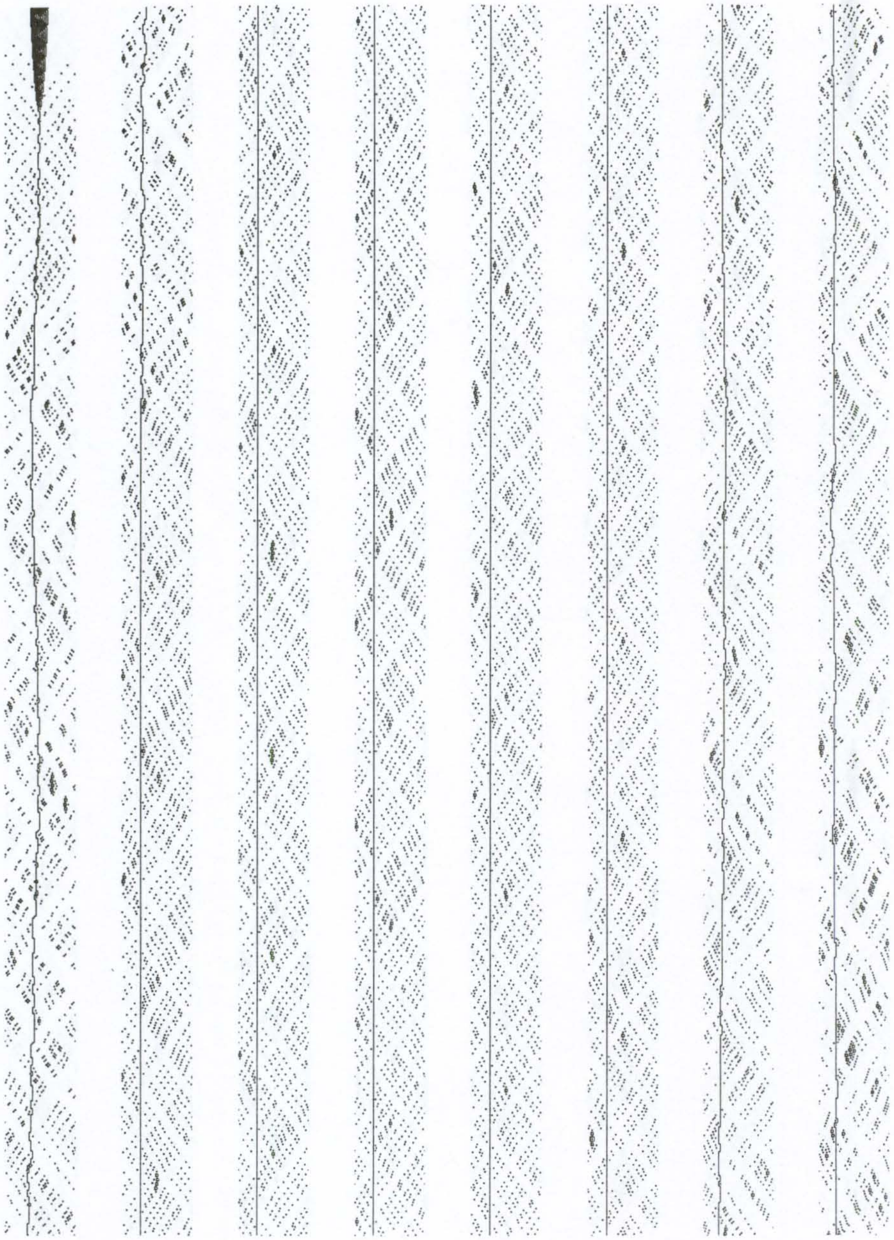


Figure 4.2 (continued).

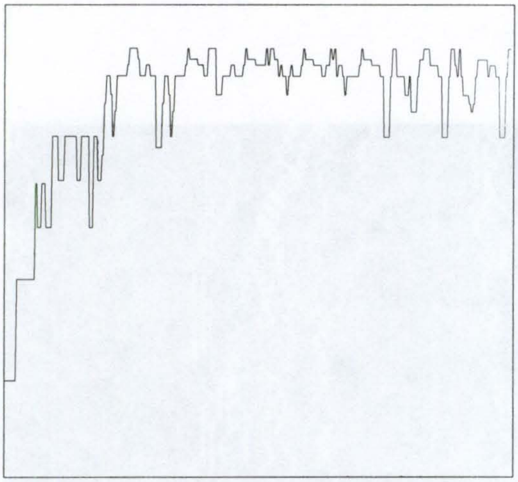


Figure 4.3. Time development of the coarse-grained entropy associated with the process of figure 4.2. The density of bits was computed in 10 bins across the system. Then the densities p_i found were combined to give the entropy

$$s = -\sum_i p_i \log p_i \,.$$

This coarse-grained entropy is seen to tend to a maximum, as expected from the Second Law of thermodynamics.

The large scale average density of bits in evolution according to the rule of table 4.1 should satisfy a diffusion equation. Figure 4.4 shows the microscopic dynamics of this rule for the cases of low and high bit density. At low bit densities, the rule exhibits particle dynamic phenomena, as might be seen in a rarefied gas. At high bit densities, however, it acts like a dense gas, and defects or particles executing apparently random walks can be seen.

22 → 11
21 → 21
20 → 02
12 → 12
11 → 22
10 → 10
02 → 20
01 → 01
00 → 00

Table 4.1. Block transitions which define the $k = 3$, $b = 2$ rule which reproduces the diffusion equation. Blocks which change under the rule are shown in bold. The rule is applied on alternate time steps to even and odd blocks in the one-dimensional cellular automaton configuration. The rule is arranged to be reversible, so that each block has a unique predecessor as well as a unique successor under the time evolution. It is also bit conserving, so that the total number of binary bits in each block is invariant under these transitions.

The microscopic configurations of this system are highly sensitive to small changes in initial conditions. Figure 4.5 shows the pattern of differences associated with the change in single initial site value. The pattern of differences is seen to expand at a fixed "speed of sound".

The overall average behaviour of this system however obeys the diffusion equation, and so is insensitive to small changes. This phenomenon is just the same as occurs in real gases.

The cellular automaton of table 4.1 can be considered as a system which contains particles executing random walks. What is perhaps remarkable about it is that the randomness necessary to produce appropriate average behaviour in these walks is generated intrinsically by the system, apparently at a low computational cost.

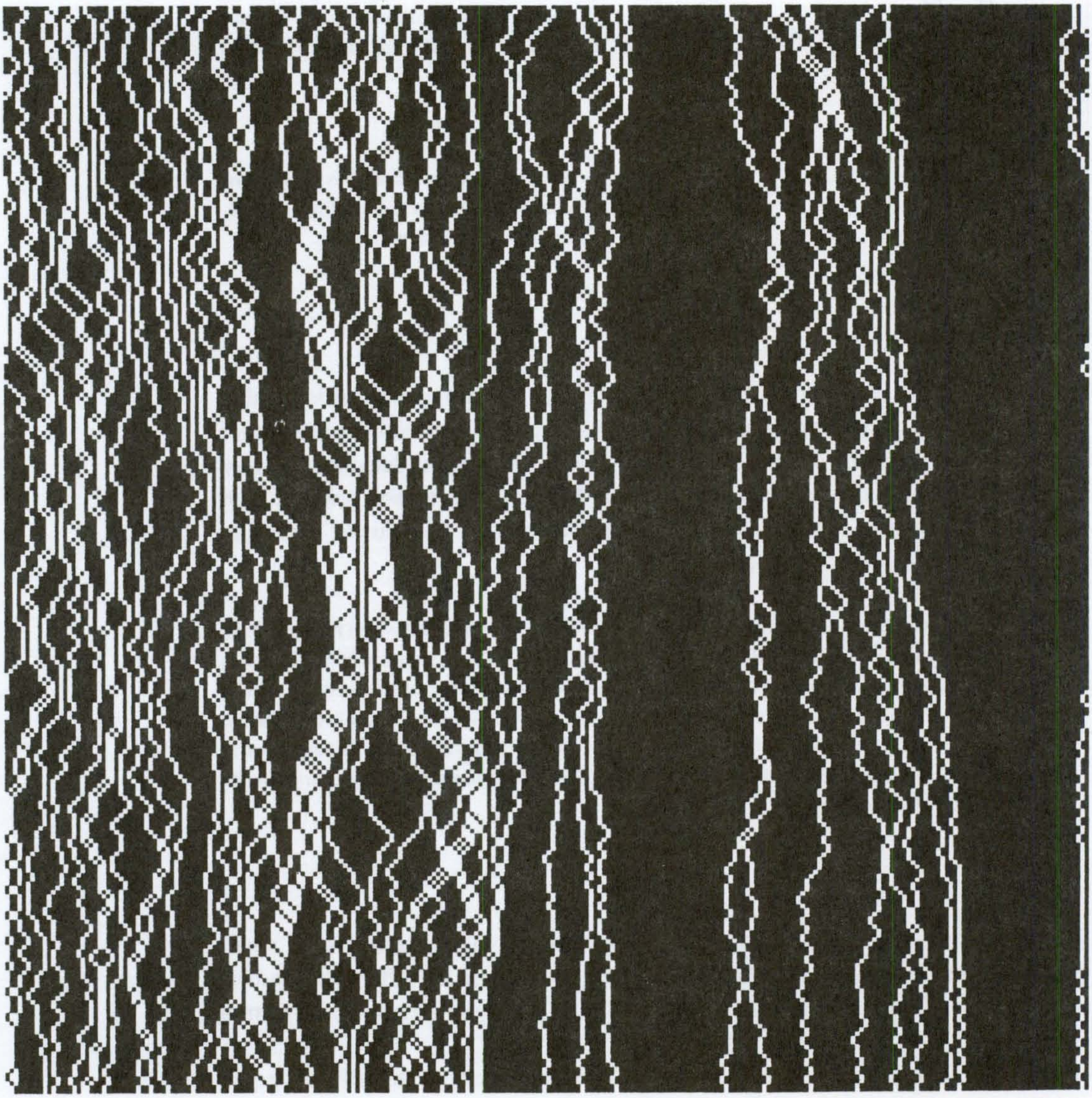


Figure 4.4. Microscopic diffusion at two densities in the minimal cellular automaton approximation to the one-dimensional diffusion equation.

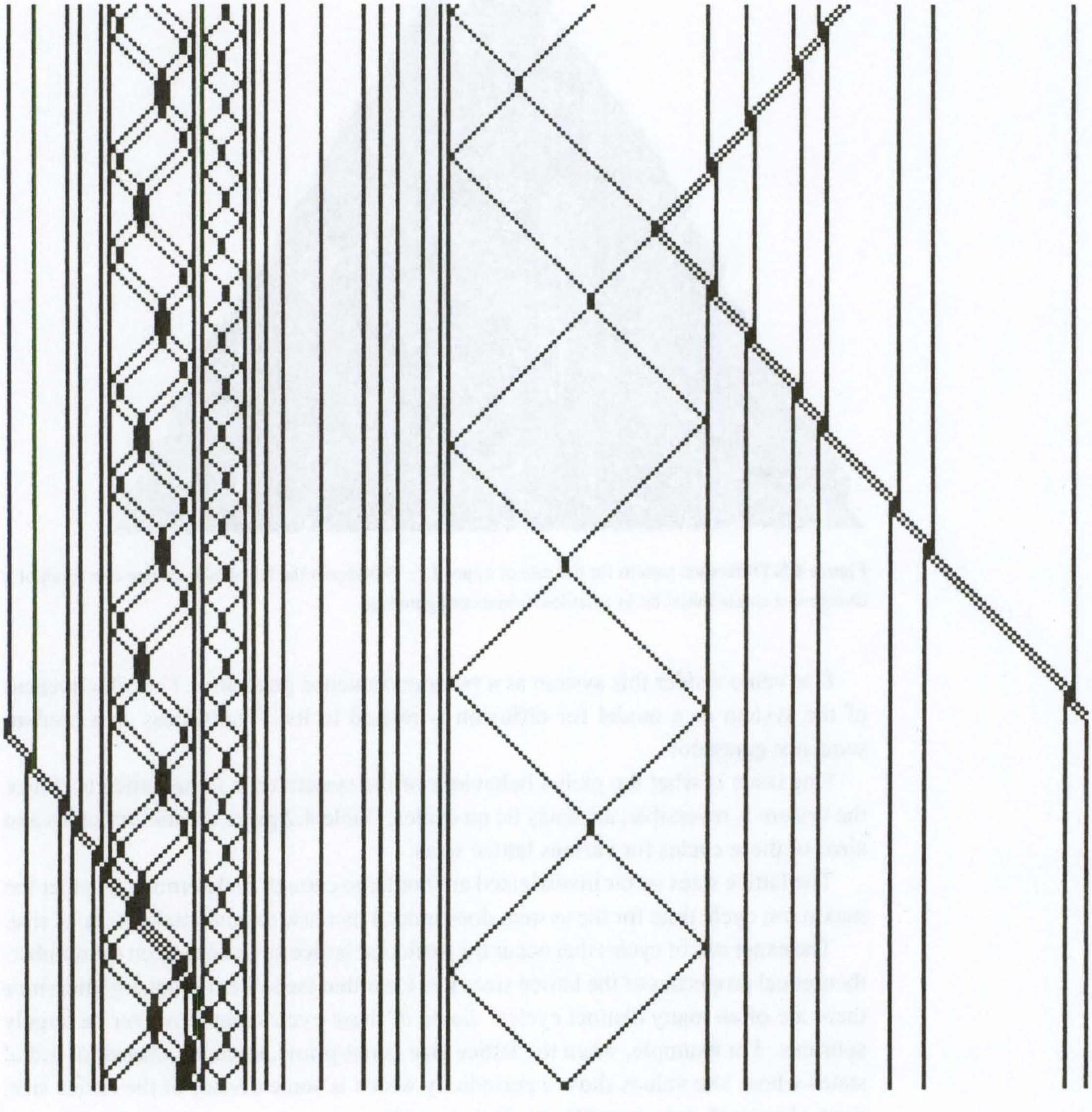


Figure 4.4 (continued).

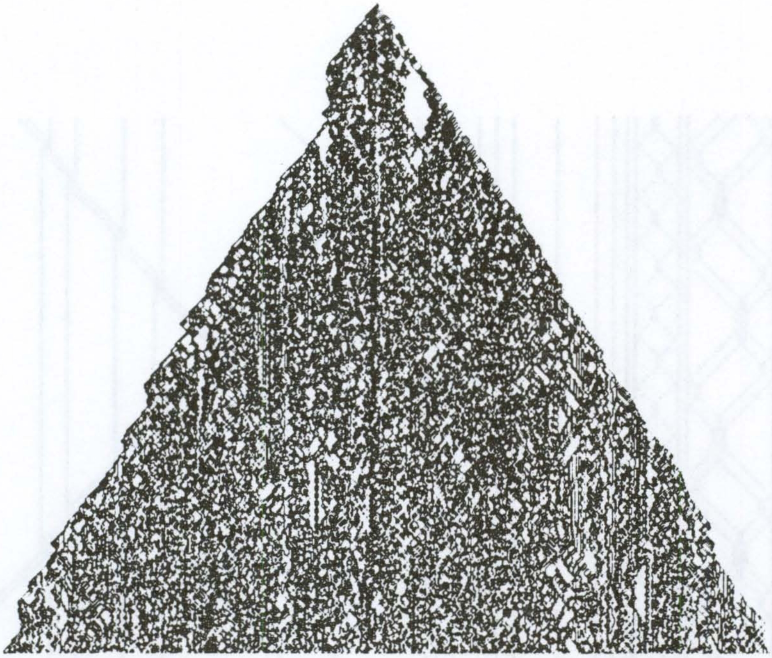


Figure 4.5. Difference pattern for the rule of figure 4.2. This shows the bits which change as a result of a change in a single initial bit in a random initial configuration.

One can consider this system as a random sequence generator. The effectiveness of the system as a model for diffusion is related to its effectiveness as a random sequence generator.

One issue is what the global behaviour of the system on a finite lattice is. Since the system is reversible, all states lie on cycles. Table 4.2 gives the multiplicities and sizes of these cycles for various lattice sizes.

The lattice sizes so far investigated are not large enough to determine whether the maximum cycle time for the system does indeed increase exponentially with its size.

The exact sets of cycles that occur for particular lattice sizes depend on the number-theoretical properties of the lattice size. It is clear that the system is not ergodic, since there are often many distinct cycles. Some of these cycles may however be largely spurious. For example, when the lattice size is not prime, there are classes of initial states whose site values show a periodicity which is some divisor of the lattice size. Such classes of states must lie on distinct cycles.

For complete randomization to occur, the system should have no conservation laws other than that of total bit number. The presence of multiple cycles implies that some other conservation laws may exist. However, no simple invariant quantities seem likely to be associated with these additional conservation laws.

size 7

total number: 1918; number distinct lengths: 6

3 × 6, 2 × 5, 8 × 4, 24 × 3, 174 × 2, 1707 × 1

size 9

total number: 17135; number distinct lengths: 11

1 × 12, 1 × 10, 1 × 9, 7 × 8, 12 × 7, 19 × 6, 31 × 5, 93 × 4, 182 × 3, 1537 × 2, 15251 × 1

size 11

total number: 24219; number distinct lengths: 143

4 × 816, 4 × 672, 4 × 654, 12 × 547, 4 × 540, 4 × 372, 8 × 366, 4 × 354, 12 × 349, 4 × 342, 6 × 330, 12 × 315
4 × 312, 2 × 270, 12 × 264, 2 × 246, 12 × 244, 8 × 240, 12 × 239, 8 × 234, 8 × 225, 8 × 222, 24 × 220, 24 × 219
12 × 194, 8 × 183, 12 × 179, 14 × 174, 12 × 169, 36 × 168, 4 × 162, 12 × 161, 8 × 159, 12 × 153, 18 × 150
12 × 149, 36 × 148, 12 × 143, 12 × 141, 24 × 139, 10 × 138, 12 × 137, 12 × 135, 6 × 130, 36 × 126, 36 × 124
12 × 121, 40 × 120, 12 × 119, 12 × 117, 24 × 115, 28 × 114, 12 × 110, 12 × 109, 48 × 108, 24 × 103, 36 × 102
28 × 96, 12 × 95, 48 × 94, 48 × 93, 24 × 91, 46 × 90, 48 × 89, 12 × 86, 12 × 85, 16 × 84, 12 × 83, 24 × 81, 24 × 80
12 × 79, 26 × 78, 12 × 77, 24 × 75, 24 × 74, 60 × 73, 64 × 72, 24 × 71, 32 × 69, 72 × 68, 12 × 67, 84 × 66, 60 × 65
24 × 64, 64 × 63, 168 × 60, 36 × 59, 32 × 57, 48 × 56, 48 × 55, 168 × 54, 12 × 52, 24 × 51, 12 × 50, 108 × 49
164 × 48, 84 × 47, 132 × 46, 124 × 45, 108 × 44, 168 × 43, 266 × 42, 132 × 41, 72 × 40, 96 × 39, 156 × 38
96 × 37, 136 × 36, 132 × 35, 108 × 34, 116 × 33, 72 × 32, 84 × 31, 218 × 30, 60 × 29, 84 × 28, 156 × 27, 258 × 26
192 × 25, 140 × 24, 168 × 23, 162 × 22, 408 × 21, 222 × 20, 360 × 19, 304 × 18, 372 × 17, 492 × 16, 1018 × 15
498 × 14, 528 × 13, 576 × 12, 546 × 11, 612 × 10, 1415 × 9, 1710 × 8, 1194 × 7, 1740 × 6, 495 × 5, 1248 × 4
1725 × 3, 1460 × 2, 1190 × 1

Table 4.2. Cycles in finite size systems evolving according to the rule of table 4.1. In each case, the cellular automaton is taken to have periodic boundary conditions. The multiplicities and sizes of all cycles are given. For width 11, the cycles have lengths which contain all primes up to 149, excluding 101, 107, 113, 127 and 131. An ergodic system would have just one cycle.

5. Discussion

This poster has illustrated a simple cellular automaton rule which exhibits continuum average behaviour in the large scale limit. It is possible to construct similar rules in two dimensions, and to give various other kinds of continuum behaviour (see several CA '86 posters).

In each case, one may compare the cellular automaton rules with traditional approaches to emulating these continuum systems on digital computers. In the conventional approach, one starts from partial differential equations, then makes discrete numerical approximations to them. These approximations involve considering a discrete lattice of points. But unlike in cellular automata, each of these lattice points carries a continuous variable which represents the precise value of a continuum quantity, such as particle density, at that point. In actual computer implementations, the continuous variable is represented by a floating-point number, say 64 bits in length. The number is updated in a sequence of time steps, according to a discrete rule. The rule in general involves arithmetic operations, which cannot be carried out precisely on the finite precision number. As a result, low-order bits of the number are truncated.

Numerical analysis has studied in detail the propagation of such round-off errors, and has suggested schemes which minimize their effects.

In a cellular automaton, the values of variables such as particle density are stored in a distributed fashion. It is necessary to average over a region of the system to find the values of such macroscopic variables. Each bit which contributes to this average is however treated according to a precise deterministic rule, and each bit is equally important. Nevertheless, the need for averaging introduces $1/\sqrt{N}$ fluctuations in the values of measured quantities. For some systems, such as turbulent ones, only statistical averages are expected to be reproducible. But in others, such as the diffusion equation, the need for averaging represents a limit on accuracy.

One can imagine a hybrid of cellular automaton and numerical analysis schemes. Consider the case of the diffusion equation. On a lattice of sites, one stores values which consist of sequences of bits. The high-order bits are encoded digitally, so that n bits can represent 2^n possible numbers of particles. The low-order bits are however encoded in unary, and correspond to individual particles. The update scheme can conserve the total number of particles.

Viewed as a numerical analysis procedure, the dynamics of the low-order bits represents a dynamics of round-off errors. Instead of systematically truncating the numbers, their low-order bits are modified according to dynamics which yields effectively random behaviour. The result is similar to random round-off, but includes a precise particle conservation law.

By adjusting the number of unary and digital bits, one can determine the tradeoffs between cellular automaton and numerical analysis approaches.

One of the major issues in numerical analysis is convergence. This is very difficult to prove for all but the simplest equations and the simplest schemes. But in cellular automata, the analogue of convergence is the process of coming to "thermodynamic" equilibrium. Thus the problem of "convergence" is related to a fundamental problem of physics.