

Twenty Problems in the Theory of Cellular Automata

1985

Cellular automata are simple mathematical systems that exhibit very complicated behaviour. They can be considered as discrete dynamical systems or as computational systems. Progress has recently been made in studying several aspects of them. Twenty central problems that remain unsolved are discussed.

Many of the complicated systems in nature have been found to have quite simple components. Their complex overall behaviour seems to arise from the cooperative effect of a very large number of parts that each follow rather simple rules. Cellular automata are a class of mathematical models that seem to capture the essential features of this phenomenon. From their study one may hope to abstract some general laws that could extend the laws of thermodynamics to encompass complex and self-organizing systems.

There has been recent progress in analysing some aspects of cellular automata. But many important problems remain. This paper discusses some of the ones that have so far been identified. The problems are intended to be broad in scope, and are probably not easy to solve. To solve any one of them completely will probably require a multitude of subsidiary questions to be asked and answered. But when they are solved, substantial progress towards a theory of cellular automata and perhaps of complex systems in general should have been made.

The emphasis of the paper is on what is not known: for expositions of what is already known about cellular automata, see [1–4]. The paper concentrates on theoretical aspects of cellular automata. There is little discussion of models for actual natural systems. But many of the theoretical issues discussed should have direct consequences for such models.

Originally published in *Physica Scripta*, volume T9 [proceedings of the fifty-ninth Nobel Symposium], pages 170–183 (1985).

Cellular automata consist of a homogeneous lattice of sites, with each site taking on one of k possible values. The sites are updated according to a definite rule that involves a neighbourhood of sites around each one. So in a one-dimensional cellular automaton the value $a_i^{(t)}$ of a site at position i evolves according to

$$a_i^{(t+1)} = \phi[a_{i-r}^{(t)}, a_{i-r+1}^{(t)}, \dots, a_{i+r}^{(t)}].$$

The local rule ϕ has a range of r sites. Its form determines the behaviour of the cellular automaton. Some examples of patterns generated by cellular automata are shown in Figs. 1 and 2. Figure 1 shows examples of the four basic classes of behaviour seen in the evolution of cellular automata from disordered initial states. Figure 2 shows patterns generated by evolution from initial configurations containing a single nonzero site.

Cellular automata may be considered as discrete dynamical systems. Their global properties are studied by considering evolution from the set of all possible initial configurations (e.g., [5]). Since most cellular automata are irreversible, the set of configurations that is generated typically contracts with time. Its limiting form at large times determines the asymptotic behaviour of the cellular automaton, and is dominated by the attractors for the evolution. Some of the properties of cellular

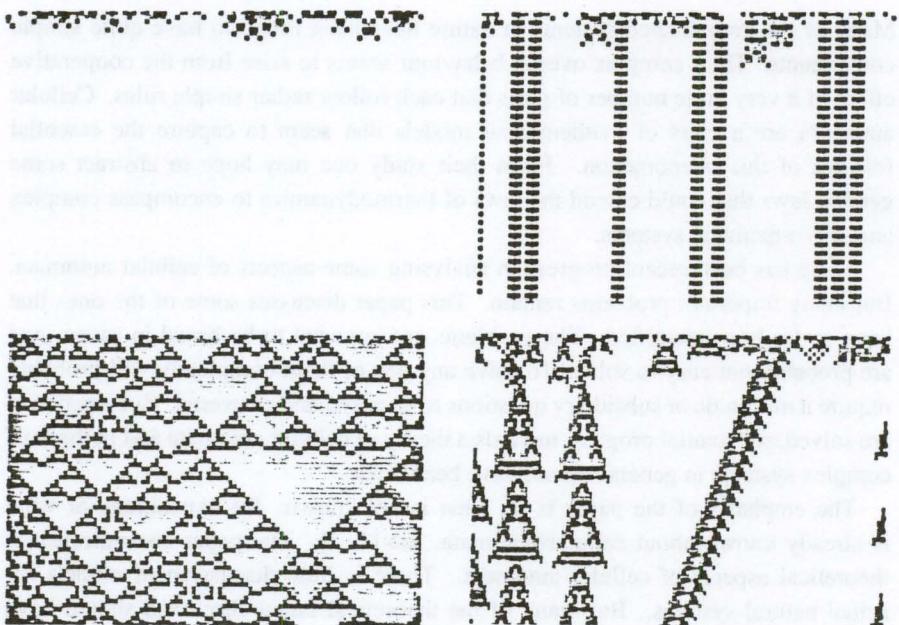


Figure 1. Examples of the four qualitative classes of behaviour seen in the evolution of one-dimensional cellular automata from disordered initial states. Successive time steps are shown on successive lines. Complex and varied behaviour is evident. The sites in the cellular automata illustrated have three possible values ($k = 3$); value 0 is shown blank, 1 is grey, and 2 is black. The value of each site at each time step is given by rules that depend on the sum of its own and its nearest neighbours' old values ($r = 1$ totalistic). The cases shown have rules specified by code numbers [5] 1302, 1005, 444 and 792, respectively.

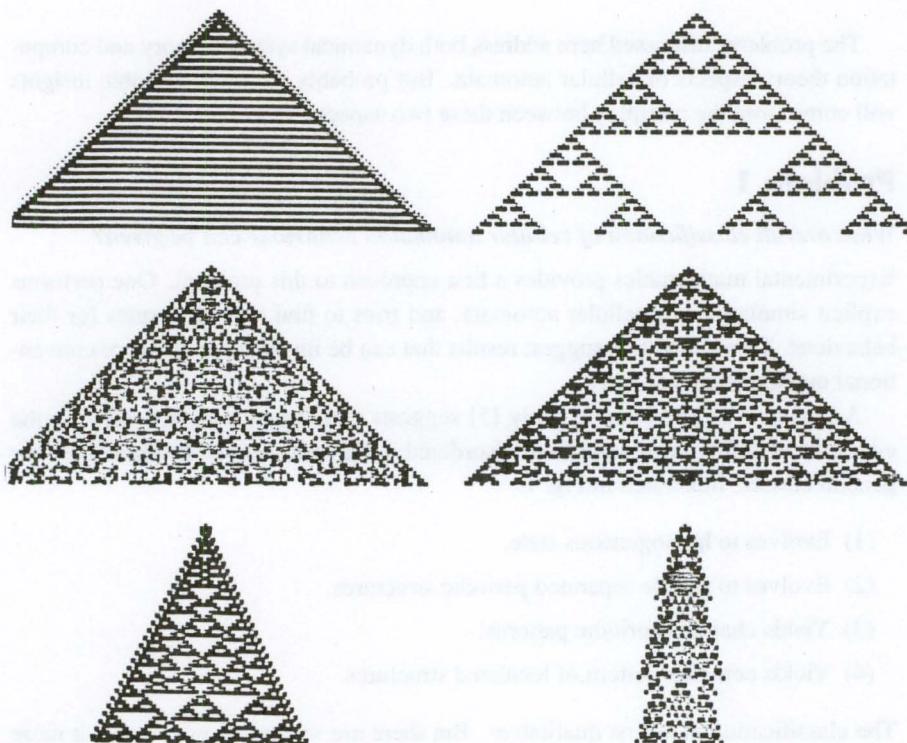


Figure 2. Examples of patterns generated by the evolution of various cellular automata starting from single site seeds. In the second case shown, a fractal pattern is generated. The subsequent cases shown illustrate the remarkable phenomenon that complicated and in some cases apparently random patterns can be generated by cellular automaton rules even from simple initial states. The cellular automata shown have $k = 3, r = 1$ totalistic rules with code numbers 1443, 312, 1554, 1617, 1410 and 600, respectively.

automata may be characterized in terms of quantities such as entropies and Lyapunov exponents that are used in studies of continuous dynamical systems (e.g., [6]).

An alternative view of cellular automata is as information-processing systems [7]. Cellular automaton evolution may be considered to carry out a computation on data represented by the initial sequence of site values. The nature of the evolution may then be characterized using methods from the theory of computation (e.g., [8]). So for example the sets of configurations generated in the evolution may be described as formal languages: a one-dimensional cellular automaton gives a regular formal language after any finite number of time steps [7]. One suspects that in many cases the computations corresponding to cellular automaton evolution are sufficiently complicated as to be irreducible (cf. [9]). In that case, there can be essentially no short-cut to determining the outcome of the cellular automaton evolution by explicit simulation or observation of each step. This implies that certain limiting properties of the cellular automaton are undecidable, since to find them would require an infinite computation.

The problems discussed here address both dynamical systems theory and computation theory aspects of cellular automata. But probably the most valuable insights will come from the interplay between these two aspects.

Problem 1

What overall classification of cellular automaton behaviour can be given?

Experimental mathematics provides a first approach to this problem. One performs explicit simulations of cellular automata, and tries to find empirical rules for their behaviour. These may then suggest results that can be investigated by more conventional mathematical methods.

An extensive experimental study [5] suggests that the patterns generated in the evolution of cellular automata from disordered initial states can be grouped into four general classes, illustrated in Fig. 1:

- (1) Evolves to homogeneous state.
- (2) Evolves to simple separated periodic structures.
- (3) Yields chaotic aperiodic patterns.
- (4) Yields complex pattern of localized structures.

The classification is at first qualitative. But there are several ways to make it more quantitative, and to formulate precise definitions for the four classes. For some cellular automaton rules, one expects that all definitions will agree. But there are likely to be borderline cases where definitions will disagree.

Continuous dynamical systems provide analogues for the classes of behaviour seen in cellular automata. Class 1 cellular automata show limit points, while class 2 cellular automata may be considered to evolve to limit cycles. Class 3 cellular automata exhibit chaotic behaviour analogous to that found with strange attractors. Class 4 cellular automata effectively have very long transients, and no direct analogue for them has been identified among continuous dynamical systems.

Dynamical systems theory gives a first approach to the quantitative characterization of cellular automaton behaviour. Various kinds of entropy may be defined for cellular automata. Each counts the number of possible sequences of site values corresponding to some spacetime region. For example, the spatial entropy gives the dimension of the set of configurations that can be generated at some time step in the evolution of the cellular automaton, starting from all possible initial states. There are in general $N(X) \leq k^X$ (k is the number of possible values for each site) possible sequences of values for a block of X sites in this set of configurations. The spatial topological entropy $d^{(x)}$ is given by $\lim_{X \rightarrow \infty} (1/X) \log_k N(X)$. One may also define a spatial measure entropy $d_\mu^{(x)}$ formed from the probabilities of possible sequences. Temporal entropies $d^{(t)}$ may then be defined to count the number of sequences that occur in the time series of values taken on by each site. Topological entropies reflect

the possible configurations of a system; measure entropies reflect those that are probable, and are insensitive to phenomena that occur with zero probability. A tentative definition of the four classes of cellular automaton behaviour may be given in terms of measure entropies. Class 1 has zero spatial and temporal measure entropy. Class 2 has zero temporal measure entropy, since it almost always yields periodic structures,

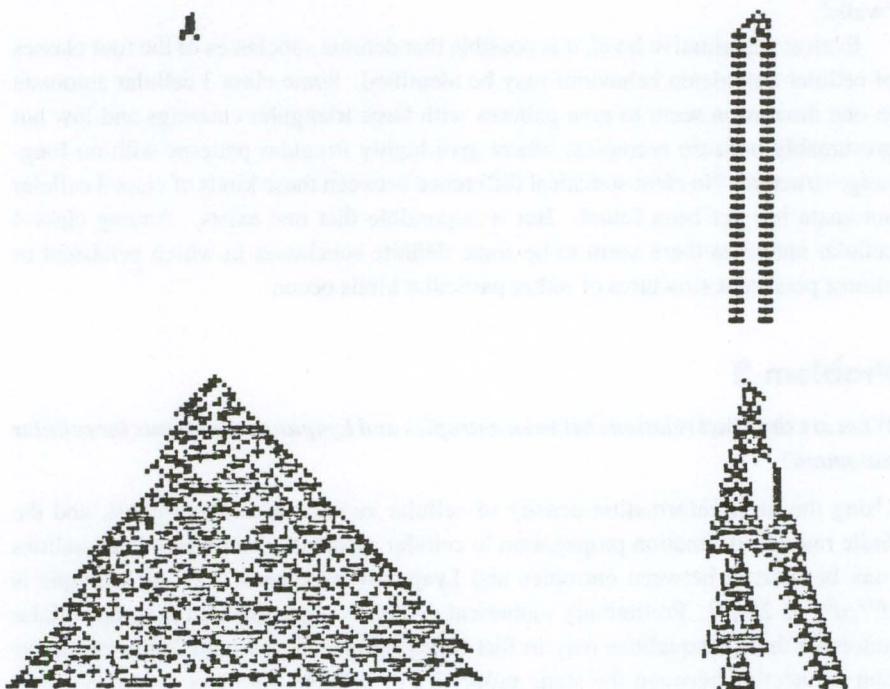


Figure 3. Patterns of differences generated by changing a single initial site value in the cellular automata of Fig. 1. In the first two cases, the difference (shown modulo three) is seen to remain localized. In the second two cases, it grows progressively with time.

but has positive spatial measure entropy. Class 3 has positive spatial and temporal measure entropies.

Another property of cellular automata is their stability under small perturbations in initial conditions. Figure 3 shows differences in patterns generated by cellular automata induced by changes in a single initial site value. Such differences almost always die out in class 1 cellular automata. In class 2 cellular automata, they may persist, but remain localized. In class 3 cellular automata, however, they typically expand at an asymptotically constant rate. The rate of this expansion gives the Lyapunov exponent for the evolution [5, 10], and measures the speed of propagation of information about the initial configuration in the cellular automaton. Class 4 cellular automata give rise to a pattern of differences that typically expands irregularly with time.

The four classes of cellular automaton behaviour identified here can be defined to be complete. But there are some cellular automata whose behaviour should probably be considered intermediate between the classes. In particular, there are many where there is a clear superposition of two classes of behaviour. So for example sites with values 0 and 1 can exhibit class 2 behaviour, while sites with values 0 and 2 show class 3 behaviour. The result is a sequence of chaotic regions separated by rigid “walls”.

Even at a qualitative level, it is possible that definite subclasses of the four classes of cellular automaton behaviour may be identified. Some class 3 cellular automata in one dimension seem to give patterns with large triangular clearings and low but presumably nonzero entropies; others give highly irregular patterns with no long-range structure. No clear statistical difference between these kinds of class 3 cellular automata has yet been found. But it is possible that one exists. Among class 4 cellular automata there seem to be some definite subclasses in which persistent or almost persistent structures of rather particular kinds occur.

Problem 2

What are the exact relations between entropies and Lyapunov exponents for cellular automata?

Using the finite information density of cellular automaton configurations, and the finite rate of information propagation in cellular automata, a number of inequalities may be derived between entropies and Lyapunov exponents (λ). An example is $d^{(t)}/d^{(x)} \leq 2\lambda$ [5]. Preliminary numerical evidence suggests that for some cellular automata these inequalities may in fact be equalities. This would imply an important connection between the static properties of cellular automata, as embodied in entropies, and their dynamic properties, as measured by Lyapunov exponents. One is hampered in these studies by the lack of an efficient method for computing entropies. The best approach so far uses a conditional entropy method [11].

Lyapunov exponents can be considered to measure the rate of divergence of trajectories in the space of configurations. In continuous dynamical systems, a geometry is defined for this space, and one can identify Lyapunov exponents for various directions.

Problem 3

What is the analogue of geometry for the configuration space of a cellular automaton?

Several simple observations may be made. First, if the cellular automaton lattice is more than one-dimensional, one may consider Lyapunov exponents in different directions on this lattice. A remarkable empirical observation is that for cellular

automata these exponents are approximately equal in all directions, even those not along the axes of the lattice, and even for cellular automata with asymmetric rules [12]. Second, in a one-dimensional cellular automaton one may consider Lyapunov exponents for subsets of configurations, or for particular components of configurations. For example, for a cellular automaton in which a class 1 component involving sites with values 0 and 2 is superimposed on class 3 behaviour involving sites with values 0 and 1, the Lyapunov exponent is positive in the "value 1" direction, and negative in the "value 2" direction. In general it seems that the cellular automaton evolution induces a form of geometry on the configuration space [13]. But the details are unclear; one does not know, for example, the analogue of the tangent space considered in continuous dynamical systems.

Problem 4

What statistical quantities characterize cellular automaton behaviour?

There are several direct statistical measurements that can be made on cellular automaton configurations. Very simple examples are densities of sites or blocks of sites with particular values. Such densities are closely related to block entropies; their limit for large block sizes is the spatial entropy of the cellular automaton configurations, equal to the dimension of the Cantor set formed by the configurations (e.g., [5]). Another direct statistical measurement that can be made is of correlation functions, which describe the interdependence of the values of separated sites [2]. For class 1 and 2 cellular automata, one expects that the correlation functions vanish beyond some critical distance. For class 3 cellular automata there are indications that the correlation functions typically fall off exponentially with distance. For class 4 cellular automata, the large distance part of the correlation function is dominated by propagating persistent structures, and may decrease slowly.

Power spectra or Fourier transforms provide other statistical measures of cellular automaton configurations. (Entirely discrete Walsh-Hadamard transforms [14] may be slightly more suitable.) Their form is not yet known. But many processes in cellular automata occur on a variety of spatial or temporal scales, so one expects definite structure in their transforms.

Beyond entropies and Lyapunov exponents, dynamical systems theory suggests that zeta functions may give a characterization of the global behaviour of cellular automata. Zeta functions measure the density of periodic sequences in cellular automaton configurations, and may possibly be related to Fourier transforms. The fact that the set of configurations generated from all possible initial states at a particular time step in the evolution of a cellular automaton forms a regular language (or "sofic system") implies that the corresponding zeta function is rational [15].

Problem 5

What invariants are there in cellular automaton evolution?

The existence of invariants or conservation laws in the evolution of a cellular automaton would imply a partitioning of its state space, much as energy provides a partitioning of the state space for Hamiltonian (energy-conserving) dynamical systems. For some class 1 and 2 cellular automata it is straightforward to identify invariants. In other cases, one can specifically construct cellular automaton rules that exhibit certain conservation laws [16–18]. For example, the cellular automata may evolve as if on several disjoint spatial lattices. Or they may support a set of persistent structures or “particles” that interact in simple ways. But in general, the identification of numerical invariants in cellular automata will probably be as difficult as it is in other non-linear dynamical systems.

It is nevertheless often possible to find partitionings of the state space for a cellular automaton that are left invariant by its evolution. The partitionings may be formed for example from sets of configurations corresponding to particular regular formal languages (cf. [7]). For example, the set of configurations with a particular period under a cellular automaton mapping is invariant, and in one dimension forms a finite-complement regular language (or “subshift of finite type”). Different elements in such partitionings may be considered to carry different values of what is often an infinite set of conserved quantities.

A particular cellular automaton rule usually evolves to give qualitatively similar behaviour from almost all initial states (each site is chosen to have each of the k possible values with equal probabilities). Often there are sets of initial states that occur with probability zero (for example, states in which all sites have the same value) that evolve differently from the rest. Such states may be distinguished by invariant or conserved quantities. But most initial states evolve to configurations with the same statistical properties. This suggests that even if the possible states could be partitioned according to the value of some invariant, they would be essentially equivalent. It remains conceivable, however, that there exist cellular automata in which two sets of initial states that occur with nonzero probabilities could lead to two qualitatively different forms of behaviour.

Problem 6

How does thermodynamics apply to cellular automata?

Thermodynamics is supposed to describe the average overall behaviour of physical systems with many components. The microscopic dynamics of these systems is assumed to be reversible, so that the mapping from one state to another with time is invertible. Most cellular automata are irreversible, so that a particular configuration may arise from several distinct predecessors. However, a small subset of cellular automaton rules are bijective or invertible. Complete tables of invertible rules exist

for $k = 2, r \leq 2$ [19, 20] and for $k = 3, r = 1$ [20], but in general no efficient procedure for finding such rules is known. Nevertheless, it is possible to construct particular classes of invertible rules [16, 21].

To apply thermodynamics one must also “coarse-grain” the system, grouping together many microscopically-different states to mimic the effect of imprecise measurements. Coarse-graining in cellular automata may be achieved by applying an irreversible transformation, perhaps a cellular automaton rule, to the cellular automaton configurations. A simple example would be to map the value of every other site to zero.

Coarse-grained entropy in reversible cellular automata should follow the second law of thermodynamics, and be on average non-decreasing with time. One may start from a set or ensemble of configurations with non-maximal coarse-grained entropy. The degrees of freedom that do not affect the coarse-grained entropy are undetermined, and are assumed to have maximal (fine-grained) entropy. In reversible class 2 cellular automata, the determined and undetermined degrees of freedom do not mix significantly with time, and the coarse-grained entropy remains essentially constant. But for class 3 and 4 cellular automata, the degrees of freedom mix, and the coarse-grained entropy increases towards its maximum possible value.

As in all applications of thermodynamics, the question arises of what coarse-graining prescriptions and ensembles of initial states are permissible. The initial states could for example be specially chosen so as to be the predecessors of a low coarse-grained entropy ensemble. The coarse-grained entropy would then decrease. Such examples do not seem physically reasonable. But it has never been clear exactly what mathematical criteria should be imposed to exclude them. One possibility is that one could require the coarse-graining procedure and the initial ensemble to be computationally simple (cf. [22]). If the cellular automaton evolution were computationally irreducible, then such a criterion could exclude ensembles obtained by reversing the evolution for many steps.

For the usual case of irreversible cellular automata, coarse-graining is usually of little consequence: the progressive contraction in the number of states generated by the cellular automaton evolution soon far outweighs the reduction associated with coarse-graining.

Problem 7

How is different behaviour distributed in the space of cellular automaton rules?

Random sampling yields some empirical indications of the frequencies of different classes of behaviour among cellular automaton rules of various kinds. For symmetric one-dimensional cellular automata, class 1 and 2 cellular automata appear to become progressively less common as k and r increase; class 3 becomes more common, and class 4 slowly becomes less common. In two-dimensional cellular automata, class 3 is overwhelmingly the most common; class 4 is very rare [12]. It seems that class

3 behaviour in any “direction” in the cellular automaton state space leads to overall class 3 behaviour. And as the number of degrees of freedom in the rules increases, the chance that this happens for one of the directions increases. For very large k and r a direct statistical treatment of the set of cellular automaton rules may well be possible.

There are many common features in the behaviour of cellular automata with apparently very different rules. It is not clear to what extent a direct equivalence exists between rules with qualitatively similar behaviour. In some cases, different rules may be related through invertible cellular automaton mappings. The nature of the equivalence classes of cellular automata generated in this way is presumably determined largely by the structure of the group of invertible cellular automaton mappings.

There are various ways to define distances in the space of cellular automaton rules. There are often cellular automata whose rules differ only slightly, but whose behaviour is very different. Nevertheless, it should be possible to find families of cellular automaton rules with closely related behaviour. For example, one may consider totalistic rules [5] in which the function that gives the new value of a site in terms of the sum of the old values in its neighbourhood is a discrete approximation to a function that involves a continuous parameter [23]. The behaviour of different cellular automaton rules obtained by changing this parameter may be compared with the behaviour found in iterated mappings of an interval of the real line (e.g., [24]) according to the same function. There are indications of a significant correspondence [23]. As the parameter is increased, regular periodic (class 2) cellular automaton behaviour can exhibit period doubling. Then as the parameter is further increased, chaotic (class 3) behaviour can occur. Class 4 seems to appear as an intermediate phenomenon.

Problem 8

What are the scaling properties of cellular automata?

Scaling transformations change the number of sites in a cellular automaton. Under such transformations, one cellular automaton rule may simulate another one. For example, if each site with value 0 is replaced by a pair of sites 00, and each 1 is replaced by 01, a new cellular automaton rule is obtained [2]. In some cases, this rule may have the same k and r as the original rule; in other cases it may not. The inverse transformation, in which 00 is replaced by 0, and 01 by 1, may be considered as a “blocking transformation” analogous to a block spin transformation (e.g., [25]), and yields a cellular automaton with fewer degrees of freedom. However, the transformation may be applied only to those special configurations in which just 00 and 01 site value pairs occur.

One may develop a network that shows the results of blocking transformations on rules of a particular kind, say with $k = 2$ and $r = 1$ [4, 26]. Some rules are found to be invariant under blocking transformations. Examples are the additive rules numbers 90 and 150 with $k = 2$ and $r = 1$. Patterns generated by these rules are thus scale invariant.

ant, so that they have the same form when viewed with different magnifications. If the initial configuration consists of a simple seed, say a single nonzero site, then regular scale-invariant patterns are obtained. These fractal patterns [27] have the property that pieces of them, when magnified, are indistinguishable from the whole pattern. (The fractal dimensions of the patterns are related to the parameters of the blocking transformations.) When the initial state is disordered, the patterns generated are instead statistically scale invariant, in the sense that their statistical properties are invariant under blocking transformations. So, for example, the pattern obtained by considering every site in the cellular automaton may have the same statistical properties as the pattern obtained by considering only every other site on every other time step.

Blocking transformations typically apply only to configurations that contain specific blocks in a given cellular automaton. So for example, different simple initial seeds in a cellular automaton may lead to rather different behaviour if they contain blocks that allow for different blocking transformations. Under certain blocking transformations, many of the $k = 2, r = 1$ cellular automata simulate the additive rules 90 or 150, which are invariant under blocking transformations. An initial state containing a single nonzero site is often one for which this simulation occurs, so that the pattern to which it leads is self-similar, just as for rule 90 or rule 150. With more complicated initial states, however, patterns with different forms may be obtained.

Starting from a disordered initial state, in which all possible sequences of site values occur with equal probabilities, the irreversible evolution of many cellular automata leads to states in which only particular sequences actually occur. If these sequences correspond to those for which some blocking transformation applies, then the overall behaviour of the cellular automaton will be given by the result of this blocking transformation. In a typical case, a cellular automaton rule supports a number of "phases". Each phase consists of sequences to which some blocking transformation applies, and under which the cellular automaton behaves just like one with a different rule. So for example [28], in the $k = 2, r = 1$ rule number 18, sequences containing only 00 and 01, or only 00 and 10, constitute two phases with behaviour just like the additive rule 90. An arbitrary disordered state consists of a series of small domains, each in one of these phases, separated by "domain walls", consisting of 11 blocks. These domain walls execute approximately random walks with time, and annihilate in pairs, leaving larger and larger domains in a pure phase [28]. In two and higher dimensional cellular automata, the domains may have complicated geometrical structures [12]. The domain walls often behave as if they have a surface tension. When the surface tension is positive, the domains tend to become spherical. When the surface tension is negative, the domains take on a highly-convoluted labyrinthine form.

It seems that one may in general define a quantity analogous to free energy, or essentially pressure, for each possible phase in a cellular automaton. Domains containing phases with higher pressures typically expand linearly with time through domains with lower pressures, sometimes following biased random walks. The

walls between domains with equal pressures typically execute unbiased random walks. After a long time, the phases with the highest pressure (or lowest free energy) dominate the behaviour of the cellular automaton, and thus determine the form of the limiting set of configurations. One may speculate that the phases that survive in this limit should be fixed points of the blocking transformation, and thus should exhibit some form of scale invariance. This is evident in some cases, where there are phases that behave say like rule 90. It is not clear how general the phenomenon is. If, however, it were widespread, then the overall large time behaviour of cellular automata would be dominated by fixed points of the blocking transformations, much as critical phenomena in spin systems are dominated by fixed points of the renormalization group or block spin transformation. Then there would be a universality in the properties of the many different cellular automata attracted to a particular fixed point rule. (So far the only fixed points of the blocking transformation that have been found are additive rules, but one suspects that not all fixed point rules must in fact be additive.) The spatial measure entropies for the different cellular automata would for example presumably then be related by simple rational factors.

One rule whose scaling properties remain unclear is the $k = 2, r = 1$ rule number 22. This rule simulates rule 90 under the blocking transformation $0000 \rightarrow 0, 0001 \rightarrow 1$, and its rotated equivalents. But the simulation is not an attractive one: starting from a disordered initial state, domains of these phases do not grow. It may be possible to describe the configurations obtained as domains of phases corresponding to some other blocking transformation. A generalization of blocking transformations may be required. One may consider a blocking transformation as a translation from one formal language to another. In simple cases, such a translation may be achieved with a finite automaton that reads symbols sequentially from the “input” configuration, and writes symbols into the “output” configuration according to the internal state that it reaches. Blocking transformations that consist of simple substitutions correspond to very simple finite automata of this kind. More complicated finite automata may be necessary to describe phases in cellular automata such as rule number 22. In general, the irreversible nature of most cellular automata implies that only a subset of possible configurations are generated with time. As a consequence, only certain neighbourhoods of site values may appear, so that some of the elements of the cellular automaton rule are never used, and a different rule would give identical results.

The description of cellular automaton configurations in terms of domains of different phases is related to a description in terms of “elementary excitations”. Just as for a spin system, one may consider decomposing a cellular automaton configuration into a “ground state” part, together with “phonons” or excitations. The excitations may for example correspond to domain walls. Or they could be persistent structures in class 4 cellular automata. But if their interactions are comparatively simple, then they can be used to provide an overall description of the cellular automaton behaviour, and perhaps allow for example a computation of entropies.

Problem 9

What is the correspondence between cellular automata and continuous systems?

Cellular automata are discrete in several respects. First, they consist of a discrete spatial lattice of sites. Second, they evolve in discrete time steps. And finally, each site has only a finite discrete set of possible values.

The first two forms of discreteness are addressed in the numerical analysis of approximate solutions to, say, differential equations. It is known that so long as a “stable” discretization is used, the exact continuum results are approximated more and more closely as the number of sites and the number of time steps is increased. It is possible to devise cellular automaton rules that provide approximations to partial differential equations in this way. In the simplest cases, however, the approximations are of the Jacobi, rather than the Gauss-Seidel kind, in that the algorithm for calculating new site values uses the old values of all the neighbours, rather than the new values of some of them. This can lead to slow convergence and instabilities in some cases.

The third form of discreteness in cellular automata is not so familiar from numerical analysis. It is an extreme form of round-off, in which each “number” can have only a few possible values (rather than the usual say 2^{16} or 2^{32}). It is not clear what aspects of, say, differential equations are preserved in such an approximation. However, preliminary studies in a few cases suggest that the overall structure of solutions to the equations are remarkably insensitive to such approximations. If the cellular automaton approximates for example a continuous field, then the value of the field at a particular point could correspond roughly to the density of say nonzero sites around that point: the values of individual field points would be represented in a distributed manner, just as they often are in actual physical systems. Explicit examples of cellular automaton approximations to partial differential equations of physical importance would be valuable.

There are some aspects of nonlinear differential equations that may well have rather direct analogues in cellular automata. For example, the persistent propagating structures found in class 4 cellular automata may well be related to solitons in nonlinear differential equations, at least in their solitary persistence, if not in their interactions. Similarly, the overall topological forms of some of the patterns generated by two and higher dimensional cellular automata [29] may correspond to those generated say by reaction-diffusion equations [30]. Moreover, many highly-nonlinear partial differential equations give solutions that exhibit discrete or cellular structure on some characteristic length scale (e.g., [31]). The interactions between components in the cellular structure cannot readily be described by a direct discretization of the original differential equation, but a cellular automaton model for them can be constructed.

Continuum descriptions may be given of many of the large-scale structures that occur in cellular automata. For example, the motion of domain walls between phases may be described by diffusion-like differential equations. A very direct continuum approximation to a cellular automaton is provided by a mean field theory, in which

only the average density of sites, and not their individual values, is considered [2]. Presumably in the limit of large spatial dimensionality, this approximation should become accurate. But in one or two dimensions, it is usually quite inadequate, and gives largely misleading results. Large-scale phenomena in cellular automata occur as collective effects involving many individual sites, and the particular rules that relate the values of these sites are significant.

Problem 10

What is the correspondence between cellular automata and stochastic systems?

Cellular automata satisfy deterministic rules. But their initial states can have a random form. And the patterns they generate can have many of the properties of statistical randomness. As a consequence, the behaviour of cellular automata may have a close correspondence with the behaviour of systems usually described by basic rules that involve noise or probabilities. So for example domain walls in cellular automata execute essentially random walks, even though the evolution of the cellular automaton as a whole is entirely deterministic. Similarly, one can construct a cellular automaton that mimics say an Ising spin system with a fixed total energy (microcanonical ensemble) [32]. Apparently random behaviour occurs as a consequence of randomly-chosen initial conditions, just as in many systems governed by the deterministic laws of classical physics.

Even models that involve explicit randomness are in practice simulated in computer experiments using pseudorandom sequences generated by some definite algorithm. These sequences are not unlike the sequences of site values produced by many cellular automata. In fact, the linear feedback shift registers often used in practice to produce pseudorandom sequences are exactly equivalent to certain additive cellular automata (cf. [33]). Empirical evidence suggests that the properties of many supposedly stochastic models are quite insensitive to the detailed form of the randomness used in their simulation. It should be possible to find entirely deterministic forms for such models, based say on cellular automata. One expects in general that just as with algorithms say for primality testing the fundamental capabilities of stochastic and deterministic models should be equivalent.

Problem 11

How are cellular automata affected by noise and other imperfections?

Many mathematical approaches to the analysis of cellular automata make essential use of their simple deterministic structure. One must find out to what extent results for the overall behaviour of cellular automata are changed when imperfections are introduced into them. The imperfections can be of several kinds. First, the cellular automaton rules can have a probabilistic element (e.g., [17, 34, 35]). Then for

example each site may be updated at each time step according to one rule with probability p , and according to another rule with probability $1 - p$. A second class of imperfections modifies the homogeneous cellular automaton lattice. One may for example take different sites to follow different rules. Or one may take the connections that specify the rules on the lattice to be different at different sites. In an ordinary cellular automaton, the values of all the sites are updated simultaneously, using the previous values of the sites in their neighbourhoods. One may consider the effect of deviations from this synchronization, allowing different sites to be updated at different times [36]. Finally, each site is usually taken to have a discrete set of possible values. One could instead allow the sites to have a continuum of values, but take the rules to be continuous functions with sharp thresholds.

Several classes of models can be considered as imperfect cellular automata. Directed percolation is directly analogous to certain cellular automata in the presence of noise [35]. The patterns generated with time by noisy cellular automata also correspond to the equilibrium configurations of spin systems at finite temperature [35]. And if inhomogeneities are introduced into the cellular automata, they give spin glass configurations. When nonlocal connections and asynchronous updates are introduced, models analogous to Boolean or neural networks are obtained (e.g., [37]).

Even an arbitrarily small imperfection in a cellular automaton can have a large effect at arbitrarily large times. However, small imperfections very often do not affect the overall behaviour of a cellular automaton. There is often a critical magnitude of imperfection at which essentially a phase transition occurs, and the behaviour of the cellular automaton changes suddenly. One can presumably find such transitions as a function of noise and other imperfections in many different cellular automata (cf. [34, 35]). Often the transitions should be associated with critical exponents; one expects that several universality classes may be identified. Note that even one-dimensional cellular automata can exhibit phase transitions at nonzero values of imperfection parameters if imperfections are introduced in such a way that for example certain initial states still evolve as they would without the imperfections.

Given a pattern generated by a cellular automaton with imperfections, as might be obtained in a physical experiment, one may consider how the basic cellular automaton rule could be deduced. One could lay down a definite grid, and then accumulate histograms of the new site values obtained with all neighbourhoods, and thereby deduce the cellular automaton rule (it will not necessarily be unique, since certain neighbourhoods may never appear) [13]. This procedure accounts for imperfections due to noise, but not for imperfections such as deformations of the lattice. It appears that an iterative optimization approach must be used to treat such imperfections.

Problem 12

Is regular language complexity generically non-decreasing with time in one-dimensional cellular automata?

The sets of configurations generated by cellular automaton evolution, starting say from all possible initial states, can be considered as formal languages. Each configuration corresponds to a word in the language, formed from a sequence of symbols representing site values, according to a definite set of grammatical rules. For one-dimensional cellular automata, it can be shown that the set of configurations generated after any finite number of time steps forms a regular formal language [7]. Thus the configurations correspond to the possible paths through a finite directed graph, whose arcs are labelled by the values that occur at each site. There is an algorithm to find the graph with the minimal number of nodes that represents a particular regular language [8, 38], in such a way that each word in the language corresponds to a unique path through the graph (deterministic finite automaton). This minimal graph provides a complete canonical description of the set generated by the cellular automaton evolution. From its properties such as topological entropy may be deduced. The entropy is in fact given by the logarithm of the largest eigenvalue of the adjacency matrix for the graph, which is an algebraic integer.

One characteristic of a regular language is the total size or number of nodes Ξ in its minimal graph. This quantity can be considered as a measure of the complexity of the regular language. The larger it is, the more complicated a subset of the space of possible symbol sequences the language corresponds to. Ξ gives in a sense the size of the shortest description of this subset, at least in terms of regular languages. The value of Ξ is in general bounded above by $2^{k^{2t}} - 1$. The empirical studies done so far suggest that for class 1 and 2 cellular automata, Ξ in fact becomes constant after a few time steps, or increases at most as a polynomial with t . For most class 3 and 4 cellular automata, however, Ξ appears to increase rapidly with time, though it usually stays far below the upper bound. There are a few cases where Ξ decreases slightly at a particular time step, but in general it seems that Ξ is usually non-decreasing with time. If this is indeed a general result, it gives a quantitative form to the qualitative statement that complexity seems to increase with time. It could be a principle for self-organizing systems analogous in generality but complementary in content to the law of entropy increase in thermodynamic systems.

If the non-decrease of Ξ is indeed a general result, then it should have a simple proof that depends on few of the properties of the system considered. A crucial property of cellular automata may be irreversibility, which leads to a progressive contraction in the set of configurations generated. As a consequence of this contraction, the set generated at each time step must correspond to a different regular language. But there are only a limited number of regular languages with complexities less than any particular value, and so the complexity of the language generated must increase, albeit slowly, with time. To find a complete bound, one must study

the structure of the space of possible regular languages. It is clear that the number of regular languages of complexity Ξ is less than the number of labelled directed graphs with Ξ nodes, $2^{k\Xi^2}$. The minimal graph for a regular language must have a trivial automorphism group; but the number of graphs with a given automorphism group does not appear to be known (e.g., [39]). Beyond the total number of regular languages, one may consider the network that represents the containment of regular languages, divided into zones of different Ξ . One suspects that this network is close to a tree, with a number of nodes increasing perhaps exponentially with depth Ξ .

Problem 13

What limit sets can cellular automata produce?

Not all possible sets of configurations can be produced as limit sets of cellular automata. For the number of distinct cellular automaton rules, while infinite, is countable. Yet the number of possible sets of configurations is uncountable.

At each step in the evolution of an irreversible cellular automaton, a new set of configurations is excluded. The limit set consists of those configurations that are never excluded. The set of all excluded configurations is recursively enumerable, since each of its elements is found by a finite computation. Thus the limit sets for cellular automata are always the complements of recursively enumerable (co-r.e.) sets, and are therefore countable in number. Nevertheless, not every co-r.e. set is the limit set for a cellular automaton: one additional condition is that they must be translationally invariant. Thus for example, cellular automaton limit sets must contain either one configuration, or an infinite number of distinct configurations, and cannot consist of some other finite number of configurations [40]. Not every possible real number value of dimension or entropy can be realized by cellular automata; but the set that is realized presumably includes some values that are non-computable.

After any finite number of time steps, the set of configurations generated by a one-dimensional cellular automaton forms a regular formal language. For some cellular automata (essentially those in classes 1 and 2), the limit set is also a regular language. But in other cases, the limit set probably corresponds to a more complicated formal language. Explicit examples are known in which context-free and context-sensitive languages are obtained as limit sets [40]. In addition, cellular automata that are capable of universal computation can generate limit sets that are not recursive [40]. The generic behaviour is however not known: some more examples would be valuable.

When the limit set forms a regular language, the simplest description of it, in terms of a regular grammar or graph, can be found by a finite algorithm. The size Ξ of this description can be used as a measure of the complexity of the set. However, for languages more complicated than regular ones, there is in general no finite algorithm to find the simplest grammar (e.g., [8]). The size of such a minimal grammar is thus formally non-computable. One may test a sequence of grammars, but the languages

to which they lead cannot in general be enumerated by a computation of any bounded length.

Minimum grammar size is thus not a useful measure of complexity for complicated cellular automaton limit sets. Some other measure must be found. And in terms of this measure, one should be able to determine how the complexity of the behaviour of a cellular automaton, as revealed by the structure of its limit set, depends on the complexity of its local rule, or the values of k and r .

One may wonder what features of the local rule for a cellular automaton determine its global properties, and the structure of its limit set. Some simple observations may be made. For example, unless the local rule contains elements that give value 1 with neighbourhoods such as 001, no information can propagate in the cellular automaton, and class 1 or 2 behaviour must occur. But in general one expects that the problem is undecidable: the only way to determine many of the limiting properties of a cellular automaton is probably by explicit simulation of its evolution, for an infinite time.

As a practical matter, one may ask whether cellular automaton rules may be constructed to yield particular limit sets (cf. [41]), so that their evolution serves to filter out the components that appear in these limit sets. It is probably possible to construct cellular automata that yield any of some class of regular languages as limit sets. But one suspects that a construction for more complicated limit sets can be carried out only in very special cases.

Problem 14

What are the connections between the computational and statistical characteristics of cellular automata?

The rate of information transmission is one attribute of cellular automata that potentially affects both computational and statistical properties. On the statistical side, the rate of information transmission gives the Lyapunov exponent for the cellular automaton evolution. Class 1 and 2 cellular automata have zero Lyapunov exponents, so that information almost always remains localized, and the value of a particular site at any time can almost always be determined from the initial values of a bounded neighbourhood of initial sites. As a consequence, the limit sets for one-dimensional such cellular automata correspond to regular languages. The configurations can thus be generated by an essentially Markovian process, in which there are no long-range correlations between different parts.

Class 3 and 4 cellular automata have positive Lyapunov exponents, so that a small initial change expands with time. The value of a particular site after many time steps thus depends in general on an ever-increasing region in the initial state. The limit sets for such cellular automata can thus involve long-range correlations, and need not correspond to regular languages. If class 4 cellular automata are generically capable of universal computation, then their limit sets should be unrestricted, in general non-recursive, formal languages. Some arguments can be given that

class 3 cellular automata should yield limit sets that correspond to context-sensitive languages. In general, one suspects that dynamical systems that exhibit chaotic behaviour characterized by positive Lyapunov exponents should yield limit sets that are more complicated than regular languages.

When the limit set for a cellular automaton is a regular language, its spatial entropy can be computed, and is given by the logarithm of an algebraic integer. If the limit set is a context-free language, then it seems that the entropy is always the logarithm of some algebraic number. But for context-sensitive and more complicated languages, the entropy is in general non-computable. It may thus be common to find class 3 and 4 cellular automata for which the entropy of their limit sets is non-computable.

The computational structure of sets generated in the evolution of two and higher dimensional cellular automata can be very complicated even after a finite number of time steps. In particular, while in one-dimensional cellular automata the set of configurations that can be generated at any finite time forms a regular formal language, this set can be non-recursive in two-dimensional cellular automata [12, 42]. The essential origin of this difference is that there is an iterative procedure to find the possible predecessors of arbitrarily long sequences in one-dimensional cellular automata, but no such procedure exists for two-dimensional cellular automata. In fact, even the problem of finding configurations that evolve periodically in time in a two-dimensional cellular automaton appears to be equivalent to the domino tiling problem, which is known to be formally undecidable [43]. Nevertheless, it seems likely that only two-dimensional cellular automata in which information transmission can occur throughout the plane, as revealed by positive Lyapunov exponents in all directions, exhibit such complications, and give non-recursive sets at finite times.

The grammar for a formal language specifies which sequences occur in the language, but not how often they occur. It does not for example distinguish sequences that occur with zero probability from those that occur with positive probability. However, it is the probable, rather than the possible, behaviour of cellular automata that is most significant in determining their statistical properties, such as Lyapunov exponents and measure entropies. There are class 1 and 2 cellular automata in which a set of states of measure zero yields class 3 behaviour: this is irrelevant in the Lyapunov exponent or the measure entropy, but affects the topological entropy, and the structure of the grammar for the limit set. One should construct formal languages that include probabilities for configurations. A suitable approach may be to consider stochastic automata, closely related to standard Markov chains.

Problem 15

How random are the sequences generated by cellular automata?

The spatial sequences obtained after a finite number of steps in the evolution of a one-dimensional cellular automaton starting from all possible initial states are known to form a regular formal language. But no such characterization is known

for the temporal sequences generated by cellular automata. At least for cellular automata capable of universal computation, these sequences can be non-recursive. But the generic behaviour is not known, and no non-trivial examples have yet been given.

One question is to what extent the initial state of a cellular automaton can be reconstructed from a knowledge of the time series of values of a few sites. An essentially equivalent question is how wide a patch of sites need to be considered to compute the invariant entropy of the cellular automaton mapping. When the mapping is surjective and expansive (so that roughly information transmission occurs at a positive rate), only a finite width is required (e.g., [44]). Nevertheless, the transformation necessary to find the initial state from the temporal sequence may be very complicated. In particular, there may be effectively no better method than to try all exponentially many possible initial states. Temporal sequences in cellular automata are thus candidates for use in pseudorandom number generation and in cryptography [20].

The patterns generated by some cellular automata evolving from initial states consisting of simple seeds have a simple form. They may be asymptotically homogeneous, or may correspond to regular fractals. But many cellular automata yield complicated patterns even starting from an initial state as simple as a single nonzero site. Some examples are shown in Fig. 2. It is remarkable that such complicated and intricate patterns can be generated in such a simple system.

Often the temporal sequences that appear in these patterns have a seemingly random form, and satisfy many statistical tests for randomness. There is empirical evidence that in many cases the sequence of values taken on say by the centre site in the pattern contains all possible subsequences with equal frequencies, so that the whole sequence effectively has maximal measure entropy. A simple example of this phenomenon occurs in the $k = 2, r = 1$ rule number 30 ($a_i^{(t+1)} = a_{i-1}^{(t)} \oplus \max(a_i^{(t)}, a_{i+1}^{(t)})$).

Systems that exhibit chaotic behaviour usually start from initial conditions that contain an infinite amount of information, either in the form of an infinite sequence of cellular automaton site values, or the infinite sequence of digits in a real number. Their irregular behaviour with time can then be viewed as a progressive excavation of the initial conditions. The chaotic behaviour seen in Fig. 2 is however of another kind: it occurs as a consequence of the dynamics of the system, even though the initial conditions are simple. It may well be that this kind of chaos is central to physical phenomena such as fluid turbulence.

It is important to investigate the mathematical bases for such behaviour. The closest analogies seem to lie in number theory. The integers generated for example by repeated application of a linear congruence transformation form a pseudorandom sequence (e.g., [45]), often used in practical applications. The linearity of this system makes it amenable to a rather complete number theoretical analysis, which provides formulae for computing the n th integer in the sequence directly from the original seed, with working out all the intermediates. It seems likely that such analyses, and

the resulting short cuts, are not possible in most nonlinear cellular automata. The randomness produced in these systems may be more like the randomness say of the digits of π . In some cases it is in fact possible to cast essentially number theoretical problems in terms of questions about patterns generated by cellular automata. One example concerns the sequence of leading binary digits in the fractional parts of successive powers of $3/2$ [46]. There is empirical evidence that all possible blocks of digits occur in this sequence, so that in a sense it has maximal entropy. The sequence corresponds to the time series of values of the central site in the pattern generated by a particular cellular automaton from a simple initial state.

Problem 16

How common are computational universality and undecidability in cellular automata?

If a system is capable of universal computation, then with appropriate initial conditions, its evolution can carry out any finite computational process. A computationally universal system can thus mimic the behaviour of any other system, and so can in a sense exhibit the most complicated possible behaviour.

Several specific cellular automata are known to be capable of universal computation. The two-dimensional nearest-neighbour cellular automaton with two possible values at each site known as the “Game of Life” has been proved computation universal [47]. The proof was carried out by showing that the cellular automaton could support structures that correspond to all the components of an idealized digital electronic computer, and that these components could be connected so to implement any algorithm. Some one-dimensional nearest-neighbour cellular automata with $k = 18$ have been shown to be computationally equivalent to the simplest known universal Turing machines, and are thus capable of universal computation [48].

One speculates that cellular automata identified on statistical grounds as class 4 are in fact generically capable of universal computation. This would imply that there exist one-dimensional computationally universal cellular automata in cases as simple as $k = 2, r = 2$ or $k = 3, r = 1$. But it remains to prove the computational universality of any particular such rule. Several methods could be used for such a proof. One is to identify a set of persistent structures in the cellular automaton that could act as the components of a digital computer, or like combinations of symbols and internal states for a Turing machine. Structures that remain fixed, propagate, and interact in various ways have been found. A structure that can act as a “clock”, producing an infinite sequence of “signals”, has not yet been found in such cellular automata. Another method of proving universality would be a direct demonstration that this cellular automaton rule could simulate any other cellular automaton rule with an appropriate encoding of initial states. Blocking transformations may provide the necessary encodings: so one must find out whether a particular cellular automaton rule is connected to all others in the simulation networks constructed from blocking transformations.

If class 4 cellular automata are indeed capable of universal computation, then the capability for universal computation is quite common among one-dimensional cellular automata. Class 4 behaviour is however much rarer in two dimensional cellular automata—the “Game of Life” is almost the only known example (cf. [12]).

There may well be cellular automata whose behaviour is usually computationally simple, but which with very special initial states can perform arbitrary computations. It is certainly possible to construct cellular automata in which universal computation occurs only with initial states in which say every other site has value zero (cf. [49]), a condition that occurs in disordered states with probability zero. Such phenomena may be common in class 3 cellular automata.

Any predictions about the behaviour of a cellular automaton must be made by performing some computation. But if the cellular automaton is capable of universal computation, then this computation must in general reduce to a direct simulation of the cellular automaton evolution. So questions about the infinite time limiting behaviour of cellular automata may require infinite computations, and therefore be formally undecidable.

For example, one may consider the question of whether the patterns generated from particular finite initial seeds ever die out in the evolution of the cellular automaton. One may simulate the evolution explicitly to find out whether a pattern dies out after say a thousand time steps; but to determine its ultimate fate in general requires a computation of unbounded length. The question is therefore formally undecidable.

The set of finite configurations that evolve to the null configuration after a fixed finite time can be specified by a regular formal language (cf. [50]). But there is no such finite specification for the set of finite configurations that evolve after any time to the null configuration. Even the fraction of configurations in this set is in general a non-computable number.

A similar problem is to determine whether a particular finite sequence of site values occurs in any configurations in the limit set for a cellular automaton. Again this problem is in general undecidable [40]. An explicit finite calculation can show that a sequence is forbidden after say three time steps. But a particular sequence may only be forbidden after some arbitrarily large number of time steps. In a one-dimensional cellular automaton, the length $L^{(t)}$ of the shortest sequence newly excluded at a given time step in the evolution is bounded by $L^{(t)} \geq L^{(t-1)} - 2r$. In most actual examples $L^{(t)}$ seems to increase monotonically with time, so that the exclusion of a particular finite sequence must occur before some predictable finite time. But in some cases $L^{(t)}$ is not monotonic, and the occurrence of particular sequences may be undecidable.

The capability for universal computation can be used to establish the undecidability of questions about the behaviour of a system. But undecidability can occur even in systems not capable of full universal computation. For example, one may arrange to disable all computations that give results of a certain form. In this way, the system fails to be able to perform arbitrary computations. Nevertheless, there may be

undecidable questions about the class of computations that it still can perform. These may well occur in cellular automata. Proofs of undecidability usually use a diagonal argument based essentially on universal computation. To establish undecidability in a system not itself capable of universal computation, one must usually find another system that is capable of universal computation, and show that a reduction of its capabilities does not affect undecidability.

Rice's theorem states that almost all questions about an arbitrary recursively-enumerable set are undecidable (e.g., [8]). However, it may be that natural or simple questions, which can be stated in say a few logical symbols, are usually decidable. So for example the halting of all simple initial seeds in a particular cellular automaton might be easy to determine, and it might only be very large and specially-chosen initial seeds whose halting was difficult to determine. There are certainly examples in which the halting problem appears to be difficult to answer even for simple seeds. One must establish in general not only whether there are any undecidable propositions about the behaviour of a particular cellular automaton, but whether simple propositions about it are in fact undecidable.

Problem 17

What is the nature of the infinite size limit for cellular automata?

Statistical averages in many systems converge to definite values when the infinite size or thermodynamic limit is taken. Several complications can however arise in cellular automata.

Different seeds can lead to very different behaviour in class 4 cellular automata. Some may die out; others may yield periodic patterns; still others may produce propagating structures. Propagating structures usually involve at least five or ten sites, and appear only with seeds of such a size. One expects that when larger seeds are used, new kinds of structures can begin to occur. For example, there may be structures that periodically generate propagating patterns, giving an asymptotically infinite number of nonzero sites. If the cellular automaton is capable of universal computation, then it should support structures with arbitrarily complicated behaviour. So for example there may be self-reproducing structures, which replicate even in the presence of a disordered background. Any such structure present in an initial state would yield offspring that could eventually dominate the behaviour of the system. In a given class 4 cellular automaton, the simplest self-reproducing structure may have a size of say 100 sites. The density at which the structure would occur in a disordered state is then k^{-100} . So in practical simulations, there is an overwhelming probability that no such structure would ever be seen. But if configurations of size much larger than k^{100} were considered, such a structure would occur in almost every case. And after a long time, the behaviour of the system would almost always be dominated by the self-reproducing structures. Statistical results obtained with smaller configurations would then be misleading. And as the idealized limit of infinite size is

taken, more and more complicated phenomena may occur, and statistical quantities have no simple limits.

Since a finite description in terms of regular formal languages can be given for the set of configurations generated at any finite time in the evolution of a one-dimensional cellular automaton, definite infinite size limits for statistical quantities presumably exist in this case. With time the limits may however become more complicated, and be reached more slowly. One expects that most statistical quantities will continue to show simple behaviour for class 3 cellular automata. But for class 4 cellular automata, in which different structure appears to be manifest on every different scale, the limits may become progressively more complicated, and may not exist at infinite times.

Two-dimensional cellular automata exhibit complicated infinite size limits even after a finite number of time steps. The sets of configurations that they generate can be non-recursive in the infinite size limit [12, 42], and some statistical quantities may have no limits as a consequence.

It is in general undecidable how large the smallest structure with some property such as self-reproduction can be in a particular cellular automaton. In some cases, the cellular automaton rule may be specially constructed to allow such structures. But for simple rules, one is reduced to an essentially experimental search for the structures. In several class 4 one-dimensional cellular automata with $k = 2$, all configurations of less than 21 sites have been tested, and all those up to about 30 sites are probably accessible with special-purpose computer hardware [51]. In the Game of Life, a number of complex structures were found through extensive experimentation. Further examples, particularly in one-dimensional cellular automata, would be valuable. One may imagine that each capability such as self-reproduction has a logical description of some length. Then the size of the smallest configuration that has the capability may be related in some way to this length. Obviously particular cellular automata may have special properties with respect to particular capabilities, but the result may hold as some average over all possible capabilities. If so, the very large number of particles in the universe could be essential for very complex physical and biological phenomena to occur.

For direct simulation and other practical purposes one is often concerned with cellular automata of finite size. When an infinite size limit exists, the local properties deduced from studies of finite cellular automata are likely to correspond directly with the infinite size case. But for global properties the correspondence is less clear. For the rather special case of finite cellular automata with additive rules, algebraic methods provide a complete description of the state transition diagram [33]. There are typically about $k^{N/2}$ cycles, each of length about $k^{N/2}$ steps. The cycles are reached after transients of length less than N . In the limit $N \rightarrow \infty$, the system exhibits chaotic behaviour, but the mapping is surjective, so that all configurations are generated. Presumably in this limit there are an infinite number of infinite cycles, perhaps each characterized by a particular form of some invariant algebraic function. In general, some cellular automata that show chaotic behaviour in the infinite size limit

exhibit exponentially long cycles at small finite sizes. Others exhibit exponentially long transients. Some show neither. The general connections between the structure of finite state transition diagrams, and the behaviour of cellular automata in the infinite size limit remain to be established.

Problem 18

How common is computational irreducibility in cellular automata?

One way to find out the behaviour of a cellular automaton is to simulate each step in its evolution explicitly. The question is how often there are better ways.

Cellular automaton evolution can be considered as a computation. A procedure can short cut this evolution only if it involves a more sophisticated computation. But there are cellular automata capable of universal computation that can perform arbitrarily sophisticated computations. So at least in these cases no short cut procedure can in general be found. The cellular automaton evolution corresponds to an irreducible computation, whose outcome can be found effectively only by carrying it out explicitly.

A number of complications arise in giving a precise definition of such computational irreducibility. In general one should compare the number of steps in the evolution of a system such as a cellular automaton with the number of steps required to reproduce the evolution using another computational system. However, by making the computational system more complicated, it is always possible to reduce the number of steps required by an arbitrary constant factor, or even an arbitrary function. For example, if a computer can apply the square of a cellular automaton mapping at each step, then it can always simulate T steps of cellular automaton evolution in $T/2$ steps.

Nevertheless, no amount of additional complication in the computer can allow it to find in a finite time the outcome of an infinite number of steps in the evolution of a cellular automata that is for example capable of universal computation. As a consequence, there are undecidable propositions about the ultimate behaviour of the cellular automaton. The occurrence of such undecidable propositions may be viewed as a consequence of computational irreducibility. But to give a complete definition of computational irreducibility for finite time processes, one must in some way exclude arbitrary complication in the computer used for predictions.

One approach is to consider finite cellular automata and to use methods from computational complexity theory. A cellular automaton with N sites can evolve for a time up to k^N before retracing its steps. The computation corresponding to this evolution is performed in a bounded space, and is therefore in the class *PSPACE* (e.g., [8]), but it can take a time exponential in N . However if the computation were reducible, then it could be possible to find the outcome of the evolution in a time polynomial in N , or in other words to reduce the problem to one in the class *P*. It is believed that *PSPACE* \neq *P*, so that there exist problems that can be solved in polynomial space that cannot be solved in polynomial time. Determining the

outcome of the evolution of some cellular automata may be a problem of this kind (cf. [52]).

Conventional computational complexity theory concerns computations in finite systems. It may well be that the definition of computational irreducibility for cellular automata can be sharpened in the infinite size limit.

The evolution of class 1 and 2 cellular automata yielding periodic configurations is computationally reducible. But one suspects that the evolution of most class 3 and 4 cellular automata is computationally irreducible. In fact, it may well be in general that most systems that show apparently complex or chaotic behaviour are computationally irreducible.

Even if the detailed behaviour of a system can effectively be found only by direct simulation, it could be that many of its overall properties can be found by more efficient procedures. It is this possibility that makes investigations of cellular automata worthwhile even when computational irreducibility is present. But what should be done is to find a characterization of those properties whose behaviour can be found by efficient methods, and those for which computational irreducibility makes explicit simulation the only possible approach, and precludes a simple description.

Problem 19

How common are computationally intractable problems about cellular automata?

Questions concerning the finite time behaviour of finite cellular automata can always be answered by finite computations. But as the phenomenon of computational irreducibility suggests, there may be questions for which the computations are necessarily very long. One may consider for example the question of whether a particular sequence of X site values can occur after T time steps in the evolution of a one-dimensional cellular automaton, starting from any initial state. Then one may ask whether there exists any algorithm that can determine the answer in a time given by some polynomial in X and T . The question can certainly be answered by testing all k^{X+2rT} sequences of initial site values that determine the length X sequence, but this procedure requires a time that grows exponentially with X and T . Nevertheless, if an initial sequence could be guessed, then it could be tested in a time polynomial in X and T . As a consequence, the problem is in the class NP . Now if $P \neq NP$, then there may be no polynomial time algorithm for the problem, and the best method of solution may essentially be to try all the exponentially many possible cases explicitly, so that the problem rapidly becomes intractable. In the infinite time limit, the analogous problem is in general undecidable.

Just as undecidability in a system can be proved by establishing a capability for universal computation, so, assuming $P \neq NP$, computational intractability can be proved through NP -completeness. A problem is NP -complete if specific instances of it correspond to arbitrary problems in the class NP [8, 53]. This can be shown by establishing equivalence to a known NP -complete problem. Thus for example it has

been possible to give a specific example of a cellular automaton in which the problem of determining whether particular sequences can occur after T time steps is equivalent to the NP -complete problem of finding a set of truth values for variables so that a particular logical expression is satisfied [54]. How widespread NP -completeness is in problems concerning cellular automata has yet to be established. But one suspects that it is common in many class 3 and 4 systems.

Problem 20

What higher-level descriptions of information processing in cellular automata can be given?

Cellular automaton evolution can in principle carry out arbitrary information processing. An important problem of theory and practice is to find a way of organizing this information processing. In specific cases one can devise cellular automaton rules that allow particular computations to be carried out (e.g., [55]). Or one can identify within a cellular automaton structures that can interact so as to mimic the components of conventional digital computers. But all these approaches are strongly based on analogues with conventional serial-processing computers. Information processing in cellular automata occurs however in a fundamentally distributed and parallel fashion, and one must invent a new framework to make use of it. Such a framework would likely be valuable in studying the many physical systems in which information processing is also distributed.

One approach is statistical in nature. It consists in devising and describing attractors for the global evolution of cellular automata. All initial configurations in a particular basin of attraction may be thought of as instances of some pattern, so that their evolution towards the same attractor may be considered as a recognition of the pattern. This approach is probably effective when the basins of attraction are local in space, as in image processing (e.g., [56]). But the construction of attractors for more general problems is likely to be very difficult. An attempt in this direction might be made by considering basins of attraction as sets of sequences corresponding to particular formal languages (cf. [50]).

Another approach is to use symbolic representations for various attributes or components of cellular automaton configurations. But the structures used in conventional computer languages are largely inappropriate. The definite organization of computer memory into named areas, stacks, and so on, is not suitable for cellular automata in which processing elements are not distinguished from memory elements. Rather perhaps data could be represented by an object like a graph, on which transformations can be performed in parallel. But the simple organizing principles that are required still remain to be found. It seems likely that a radically new approach is needed [57].

Acknowledgements

I have benefitted from discussions about cellular automata with many people, too numerous to list here. For recent discussions I am particularly grateful to: C. Bennett, M. Feigenbaum, E. Fredkin, D. Hillis, L. Hurd, L. Kadanoff, D. Lind, O. Martin, J. Milnor, N. Packard, D. Ruelle, R. Shaw, and K. Steiglitz.

References

1. Wolfram, S., *Nature* **311**, 419 (1984).
2. Wolfram, S., *Rev. Mod. Phys.* **55**, 601 (1983).
3. Farmer, D., Toffoli, T. and Wolfram, S. (editors), "Cellular automata: proceedings of an interdisciplinary workshop", *Physica* **10D** numbers 1 and 2 (1984), North-Holland Publishing Co. (1984).
4. Wolfram, S., "Cellular automata", Los Alamos Science (fall 1983).
5. Wolfram, S., *Physica* **10D**, 1 (1984).
6. Guckenheimer, J. and Holmes, P., *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, Springer-Verlag (1983).
7. Wolfram, S., *Comm. Math. Phys.* **96**, 15 (1984).
8. Hopcroft, J. E. and Ullman, J. D., *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley (1979).
9. Wolfram, S., "Computer software in science and mathematics", *Scientific American* (September 1984).
10. Packard, N., "Complexity of growing patterns in cellular automata", Institute for Advanced Study preprint (October 1983).
11. Milnor, J., "Entropy of cellular automaton-maps", Institute for Advanced Study preprint (May 1984).
12. Packard, N. and Wolfram, S., "Two-dimensional cellular automata", *J. Stat. Phys.* **38**, 901 (1985).
13. Packard, N., Private communication.
14. Ahmed, N. and Rao, K. R., *Orthogonal Transforms for Digital Signal Processing*, Springer-Verlag (1975).
15. Franks, J. and Fried, D., Private communications.
16. Margolus, N., *Physica* **10D**, 81 (1984).
17. Vichniac, G., *Physica* **10D**, 96 (1984).
18. Pomeau, Y., *J. Phys. A17*, L415 (1984).
19. Hedlund, G., Private communication.
20. Milnor, J. and Wolfram, S., In preparation.
21. Milnor, J., "Notes on surjective cellular automaton-maps", Institute for Advanced Study preprint (June 1984).
22. Chaitin, G., "Towards a mathematical definition of life", in *The Maximum Entropy Formalism*, R. D. Levine and M. Tribus (eds.), MIT press (1979).
23. Crutchfield, J. and Packard, N., Private communication.
24. Collet, P. and Eckmann, J.-P., *Iterated Maps on the Interval as Dynamical Systems*, Birkhauser (1980).
25. Amit, D., *Field Theory, the Renormalization Group, and Critical Phenomena*, McGraw-Hill (1978).
26. Milnor, J., Unpublished.
27. Mandelbrot, B., *The Fractal Geometry of Nature*, Freeman (1982).

28. Grassberger, P., *Physica* **10D**, 52 (1984).
29. Greenberg, J. M., Hassard, B. D., and Hastings, S. P., *Bull. Amer. Math. Soc.* **84**, 1296 (1975); Madore, B. and Freedman, W., *Science* **222**, 615 (1983).
30. Winfree, A. and Winfree, E., "Organizing centers in a cellular excitable medium", to be published.
31. Packard, N., "Cellular automaton models for dendritic growth", Institute for Advanced Study preprint, in preparation.
32. Bennett, C., Neuberger, H., Pomeau, Y. and Vichniac, G., Private communications.
33. Martin, O., Odlyzko, A. and Wolfram, S., *Comm. Math. Phys.* **93**, 219 (1984).
34. Grassberger, P., Krause, F. and von der Twer, T., "A new type of kinetic critical phenomena", University of Wuppertal preprint WU B 83-22 (October 1983).
35. Domany, E. and Kinzel, W., *Phys. Rev. Lett.* **53**, 311 (1984).
36. Ingerson, T. E. and Buvel, R. L., *Physica* **10D**, 59 (1984).
37. Kauffman, S., *Physica* **10D**, 145 (1984).
38. Hopcroft, J., "An $n \log n$ algorithm for minimizing states in a finite automaton", in Proc. Int. Symp. on the Theory of Machines and Computations, Academic Press (1971).
39. Harary, F., Graph Theory, Chapter 15, Addison-Wesley (1972).
40. Hurd, L., "Formal language characterizations of cellular automaton limit sets", to be published.
41. Hasslacher, B., Private communication.
42. Yaku, T., *J. Comput. System Sci.* **7**, 481 (1973); Golze, U., "Differences between 1- and 2-dimensional cell spaces", in A. Lindenmayer and G. Rozenberg (eds.), *Automata, Languages, Development*, North-Holland (1976).
43. Berger, R., *Mem. Amer. Math. Soc.*, no. 66 (1966); Robinson, R., *Inventiones Math.* **12**, 177 (1971).
44. Walters, P., *An Introduction to Ergodic Theory*, Springer (1982).
45. Knuth, D., *Seminumerical Algorithms*, 2nd. ed., Addison-Wesley (1981).
46. Furstenberg, H. and Lind, D., Private communication.
47. Berlekamp, E. R., Conway, J. H. and Guy, R. K., *Winning Ways for Your Mathematical Plays*, vol. 2, chap. 25, Academic Press (1982); Gardner, M., *Wheels, Life and Other Mathematical Amusements*, Freeman (1983).
48. Smith, A. R., *J. ACM* **18**, 339 (1971).
49. Banks, E. R., "Information processing and transmission in cellular automata", MIT project MAC report no. TR-81 (1971).
50. Smith, A. R., *J. Comput. Sys. Sci.* **6**, 233 (1972); Sommerhalder, R. and van Westrenen, S. C., *Acta Inform.* **19**, 397 (1983).
51. Steiglitz, K., Private communication.
52. Bennett, C. H., "On the logical "depth" of sequences and their reducibilities to random sequences", *Info. & Control*, to be published.
53. Garey, M. R. and Johnson, D. S., *Computers and Intractability: a Guide to the Theory of NP-completeness*, Freeman (1979).
54. Sewelson, V., Private communication.
55. Cole, S. N., *IEEE Trans. Comput.* **C-18**, 349 (1969).
56. Preston, K. et al., *Proc. IEEE* **67**, 826 (1979).
57. Hillis, D. and Wolfram, S., Work in progress.