Generalized shifts: unpredictability and undecidability in dynamical systems

Cristopher Moore

Department of Physics, Cornell University, Ithaca, NY 14853, USA

Abstract. A class of shift-like dynamical systems is presented that displays a wide variety of behaviours. Three examples are presented along with some general definitions and results.

   A correspondence with Turing machines allows us to discuss issues of predictability and complexity. These systems possess a type of unpredictability qualitatively stronger than that which has been previously discussed in the study of low-dimensional chaos, and many simple questions about their dynamics are undecidable.

   We discuss the complexity of various sets they generate, including periodic points, basins of attraction, and time series. Finally, we show that they can be embedded in smooth maps in R^2, or smooth flows in R^3.

1. Introduction

In the dynamical systems literature, 'chaos' is usually exemplified by the shift map,

$$\sigma : \Sigma \to \Sigma : (a)_i \mapsto (a)_{i+1} \tag{1}$$

where $\Sigma = A^Z$ is the set of bi-infinite sequences of some finite alphabet A. The most famous occurrence of this is in Smale's 'horseshoe' map [1, 2]: whenever a map folds some open set into itself in a way like figure 1(a), the set of non-wandering points is a square Cantor set, in which each point corresponds to a sequence or 'itinerary' in $\{0, 1\}^Z$ describing which half (upper or lower) of the square the point is in at each iteration of the map forward or backward in time. This is a homeomorphism, and the action of the map then induces the shift on two symbols.

   We can portray the horseshoe as a map of the unit onto itself, as in figure 1(b); figure 1(c) shows the 'baker's map' [3], which is also conjugate to the shift on two symbols. In this case, the left and right halves of a point's sequence are simply the binary expansions of its x and y coordinates respectively.

   As a straightforward generalization of the shift, we could consider a map in
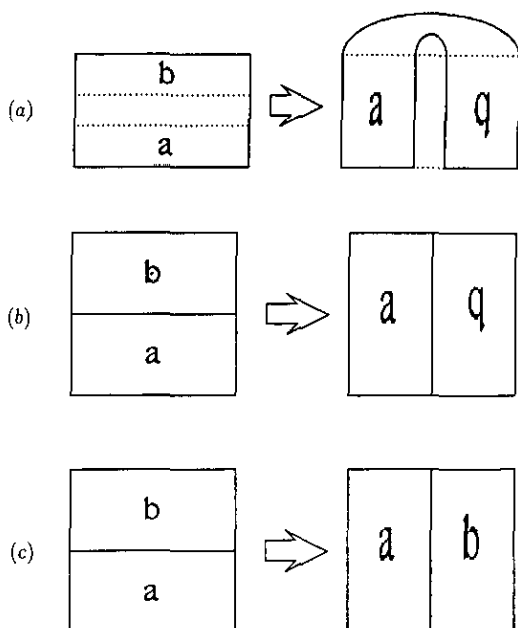
Figure 1. (*a*) The Smale horseshoe. (*b*) The Smale horseshoe, abbreviated as a map of the unit square. (*c*) The 'baker's map', similarly portrayed.

which we shift left or right by a variable amount, depending on a finite number of places in the sequence:

$$\Phi : a \mapsto \sigma^{F(a)} a$$

where $a \in \Sigma$ and $F : \Sigma \to Z$ has a finite *domain of dependence* (DoD), the set of integers i such that $F(a)$ depends on a_i. We could further generalize, and allow a finite number of cells in the sequence to change:

$$\Phi : a \mapsto \sigma^{F(a)} (a \oplus G(a)) \tag{2}$$

where
  (1) $F : A^Z \to \mathbb{Z}$ and $G : A^Z \to (A \cup \{\phi\})^Z$ both have a finite DoD,
  (2) every sequence in the image of G has a value of ϕ except for a finite number of cells, and
  (3) for $a \in A^Z$ and $g \in (A \cup \{\phi\})^Z$, $a \oplus g$ (read '*a* replace *g*') is defined as follows:

$$(a \oplus g)_i = \begin{cases} g_i & \text{if } g_i \neq \phi \\ a_i & \text{if } g_i = \phi. \end{cases}$$

(i.e. G specifies what modifications to perform on the sequence.) Such a system Φ will be called a *generalized shift* (on A) and is determined by the functions F and G. (Note that since F and G depend on a finite number of cells in a, they take a finite number of different values: in particular, F is bounded above and below.) Here ϕ is just an auxiliary character for the notation.

In other words, the dynamics works as follows: depending on a finite number of cells in the sequence, we change some of them according to G, and then shift left or right by an amount F. Unlike the standard shift, we do not obviously have invariant subshifts, since the sequence is being modified as well as shifted; in this paper we will look just at the 'full shift' A^Z.

The set of cells that are modified by G (integers i s.t. $g_i \neq \phi$ for some g in Im G) will be referred to as the *domain of effect* (DoE). It turns out that it plays many of the same roles as the DoD.

This class of systems is reasonable to study because it is large enough to display many types of behaviour, and small enough for the following lemma to hold.

Lemma 0. Any generalized shift on n symbols is conjugate to a piecewise linear map of the square Cantor set into itself, by the homeomorphism induced by the horseshoe or baker's map; this map will have a finite number of linear components

$$k \leqslant n^{|\text{DoD} \cup \text{DoE}| + \max |F|}$$

each of which is measure-preserving.

Proof. Let $D = \text{DoD} \cup \text{DoE}$, and partition the Cantor set into $n^{|D|}$ subsets, characterized by their sequences' values on D. Each of these is connected, as long as the origin is contained in D. Each of these is first mapped onto another one by G, and then shifted by F. When it is shifted, one coordinate is stretched by $n^{|F|}$; then to fold it back into the square, it must be cut into at most $n^{|F|}$ pieces. Then the number of components $k \leqslant (n^{|D|})(n^{\max |F|})$ as stated. QED. (The homeomorphism induced by the baker's map is probably the most convenient, as it produces maps which preserve the directions of the x and y axes.)

Conversely, it is possible to show [4] that any C^2 area-preserving diffeomorphism of R^2 which preserves this Cantor set is homeomorphic to a generalized shift.

We also have the following reduction lemma.

Lemma 1. Any generalized shift Φ on a finite alphabet A is conjugate to some Φ' on the two-symbol alphabet $\{0, 1\}$.

Proof. We simply map letters of A onto sufficiently large blocks of 0s and 1s: since $|A| \leqslant 2^m$ for some m, construct some function $E : A \to \{0, 1\}^m$ and let

$$\mathscr{E} : \{0, 1\}^Z \to A^Z : a_i \mapsto E^{-1}(a_{mi} \ldots a_{m(i+1)-1}).$$

Then let

$$F'(a) = mF(\mathscr{E}a)$$

and

$$G'(a) = \mathscr{E}^{-1}G(\mathscr{E}a).$$

So we can take $A = \{0, 1\}$ with no loss of generality. QED.

In the following sections, we first present three examples of GSs with various sorts of behaviour, and do what we can to describe them. We then give an introduction to Turing machines and the languages they produce, and show that GSs and Turing machines are essentially equivalent. We use this equivalence to show the undecidability of many simple dynamical questions about GSs, such as questions about their periodic points, basins of attraction, ergodicity, etc. We then discuss the qualitative nature of their unpredictability and show how it is different, and stronger, than the standard kind of 'chaos'. Finally we show that these systems can

be suspended in smooth maps and flows, and we close with a discussion of whether such systems might be found in nature.

In the examples below we will take $A = \{0, 1\}$, and the DoD for both F and G as well as the DoE will typically be the set of integers between $-r$ and r for some small r. (Unless the DoD is kept as small as possible, things get out of hand.) Throughout, a 'language' is simply a set of sequences; see [8] for an introduction to various sorts of languages.
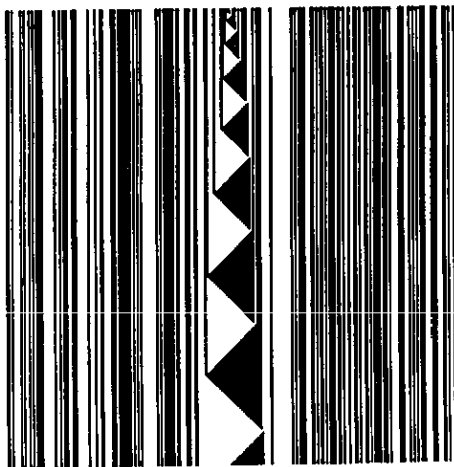
2. Examples

We now present and study three examples. Throughout, Φ is the dynamics defined above, $n = |A|$ is the number of symbols in the alphabet, and w is the width of the DoD, i.e. the size of the smallest continuous set of integers that contains the DoD ($= 2r + 1$ in these cases.) We use the product measure on the space of sequences Σ, and the standard metric

$$d(x, y) = \sum_{i \in \mathbb{Z}} (1 - \delta(x_i, y_i)) n^{-|i|}.$$

2.1. Example 1

Let F and G depend only on a_0, and let G only modify a_0 (DoD = DoE = $\{0\}$.) Then let $F(0) = +1$, $F(1) = -1$, $G(0) = 1$, and $G(1) = 0$; i.e. 'if the cell at the origin is a 0, change it to a 1 and shift left; if it is a 1, change it to a 0 and shift right'.

There are two fixed points, $\ldots 111.1000 \ldots$ and $\ldots 111.000 \ldots$, where a_0 is immediately to the right of the decimal point; we will denote these $(1).1(0)$ and $(1).(0)$ respectively. In fact, it is easy to show that these are the only periodic points, and that as fixed points of the conjugate map in the plane they are hyperbolic. Figure 2 shows the evolution of a random initial sequence; it shows that the effect of the map is to make the origin bounce back and forth between the nearest boundaries between 0s and 1s, pushing them back one space each time. Thus the

Figure 2. A simulation of example 1 from random initial conditions. Black = 0, white = 1. Instead of shifting the sequence and keeping the DoD and DoE in the centre, we shift a 'pointer' that marks where the origin is and keep background sequence constant. As the evolution progresses, we pass closer and closer to the two fixed points, $(1).1(0)$ (moving left) and $(1).(0)$ (moving right.)

generic behavior alternates between the neighbourhoods of the two fixed points, coming closer (according to the standard metric on sequences) and staying longer each time.

This is a comparatively 'unchaotic' kind of behaviour; the time $B(d)$ (B for 'busy') it takes to shift a total distance of d goes as

$$B(d) \propto d^2$$

like a random walk. This corresponds to a sub-exponential divergence of close initial conditions; normally

$$B(d) = d/v \approx (\ln n / \lambda)d$$

where v is the speed of the shift (cells per unit time), n is the number of symbols in the alphabet, and λ is the Liapunov exponent. Here λ approaches zero in the long-time limit.

This behaviour is easy to explain qualitatively. If we use the Smale homeomorphism from sequences to points in the Cantor set, we get a non-invertible map of the unit square into itself as shown in figure 3. Clearly every point in the square is 'hyperbolic' in that either the x or y direction is getting stretched, and the other compressed (corresponding to shifting left or right); but the evolution of almost every point involves switching which eigenvector is which, so that the expansion and contraction is not uniform. We see that one half of the square is getting squished, and the other half squashed; moreover, almost every point spends an equal amount of time in these two domains. Thus any eigenvector gets halved as often as it gets doubled—this explains why things take a long time to diverge so that $B(d)$ is supra-linear. (In terms of the sequence, we shift left and right equally often, so we do not get far very quickly.) The exact form of $B(d)$ depends on the statistical distribution of the squishings and squashings; in this example it is quadratic in d, while in the next we will see that $B(d)$ can be exponential.

Since we do not have a uniform hyperbolic structure, we do not have a Markov partition [7]: local stable and unstable manifolds are getting mapped onto each other, and so are ill-defined as $t \to \infty$. In particular, if we record the value of a_0 (or of $F(a)$), the set of possible sequences produced by the dynamics is not a subshift of finite type, or even a regular or context-free language [8]: it is the set

$$\{0^{s_1}1^{s_2} \ldots 0^{s_{2k-1}}1^{s_{2k}} \ldots\}$$

where $s_{i+1} > s_i$ for all $i > 0$. This simply describes the fact that the map oscillates between the two nearest boundaries, pushing them back or annihilating them on
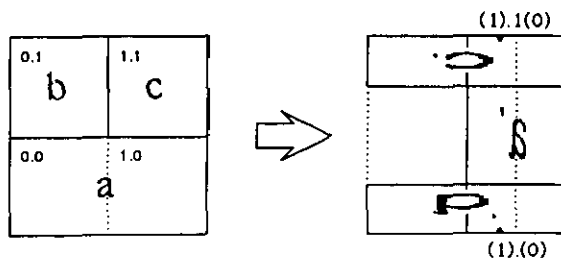

Figure 3. The map of the plane corresponding to example 1, showing the two fixed points.

each bounce, so that each oscillation takes longer than the last. (We discuss this language in the section on time series below.)

This example is in a rather restricted subclass of generalized shifts:

Definition. Let $\langle F \rangle$ be the semigroup of integers generated by the image of F in $\mathbb{Z}$. Then we call a generalized shift *stagnant* if

$$\forall i \in \langle F \rangle, \ i \neq 0 : \sigma^i(\text{DoE}) \cap \text{DoD} = \varnothing$$

For instance, Φ is stagnant if $\text{DoD} = \text{DoE} = \{0\}$, as in this case. We call these systems 'stagnant' since information cannot really move around: F and G never get to depend on the cells that they modified before, except to modify the same ones again. Each cell evolves independently of its neighbours, $x \mapsto G(x)$ (although the time at which it does so may be complicated) so the system cannot do anything very interesting†.
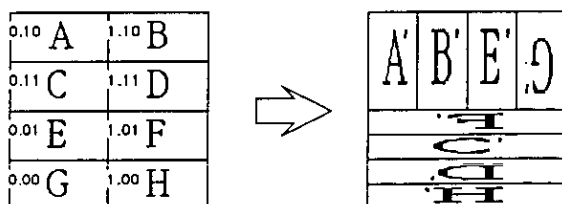
2.2. Example 2

Let $\text{DoD} = \text{DoE} = \{-1, 0, +1\}$, with F and G as follows:

$a_{-1}a_0a_{+1}$	F	G
0.00	+1	0.11
0.01	+1	1.11
0.10	+1	0.01
0.11	-1	0.11
1.00	-1	0.00
1.01	-1	0.10
1.10	+1	1.01
1.11	-1	0.01

This map corresponds to the map of the square shown in figure 4, and so, unlike example 1, it is 1–1 (even though neither F nor G is.) We have the following lemma.

Lemma 2. If Φ is a generalized shift, the following are equivalent:
   (1) Φ is 1–1;
   (2) Φ is onto;
   (3) Φ is measure-preserving as a map of the unit square.

Figure 4. The map of the plane corresponding to example 2.

† It seems likely that we can calculate the average $B(d)$ analytically for systems with $\text{DoD} = \text{DoE} = \{0\}$ by mapping them onto a type of random walk.

Proof. From lemma 0, we can construct a map $\bar{\Phi}$, conjugate to Φ, of the Cantor set into itself with a finite number of components, each of which is measure-preserving. Then $\bar{\Phi}$ and Φ are 1–1, onto, and measure-preserving iff none of these components overlap. QED.

I know of no simple criterion in terms of F and G for Φ being 1–1, although it is certainly decidable by constructing $\bar{\Phi}$.

This map has three fixed points: $(0).1(0)$, $(1).01(0)$, and $(1).(0)$. In studying fixed or periodic points the following definition is useful.

Definition. For a point $x \in \Sigma$, the *shift number* (for time t) is

$$S_t(x) = \sum_{0 \leqslant i < t} F(\Phi^i x)$$

the 'net amount of shifting' that takes place during t iterations of Φ on x. If x is periodic with period T, we define

$$S(x) = S_T(x).$$

The shift numbers $S(x)$ of the fixed points are $+1$, -1, and -1 respectively. $S(x)$ tells us about the behaviour of the map near the periodic orbit.

Lemma 3. Let x be periodic with period T. Then the eigenvalues of $D\Phi^T$ near the orbit of x are $n^{S(x)}$ and $n^{-S(x)}$.

Proof. Let $dx \in (A \cup \{\phi\})^Z$, and define $|dx|$ as

$$|dx| = \sum_{i \,:\, dx_i \neq \phi} n^{-|i|}.$$

Then $d(x, x \oplus dx) \leqslant |dx|$ where d is the usual metric on Σ. Then if

$$|dx| < N^{-(r+t \max |F|)}$$

it is easy to see that

$$\Phi^t(x \oplus dx) = \Phi^t x \oplus \sigma^{S_t(x)} \, dx.$$

In other words, $S_t(x)$ measures the amount of shifting that the sequence undergoes far away from the DoD and DoE; cells that are never brought within r of the origin are simply shifted, not modified, and the dynamics never depends on them. Specifically, if x is periodic with period T,

$$\Phi^T(x \oplus dx) = x \oplus \sigma^{S(x)} \, dx.$$

Then perturbations to the left or right are shifted away from or toward the origin by $S(x)$, so the eigenvalues are $n^{-S(x)}$ and $n^{S(x)}$ respectively. QED.

For example, the eigenvalues of the periodic points of this map are 2 and 1/2.

Theorem 4. The set of periodic points is of non-zero measure iff there exist periodic x with $S(x) = 0$.

Proof. First we prove that there can only be a countable number of periodic points x with $S(x) \neq 0$. As in lemma 3, we have that cells sufficiently far from the origin are

simply shifted by $S(x)$ over the course of the orbit, i.e.

$$(\Phi^T(x))_i = x_{i+S(x)} \text{ for } |i| > r + T \max |F|.$$

But if $S(x) \neq 0$, this means that x must have a periodic head and tail; i.e. x must be of the form AbC where b is a finite word of length $\leq w + T \max |F|$, A and C are semi-infinite sequences, and A and C are periodic with wavelength $S(x)$ (or a factor of $S(x)$. There are only a countable number of such 'rational' sequences.

   Now suppose there is an x with $S(x) = 0$. Only a finite set of cells, of width $m \leq w + T \max |F|$, ever influences the dynamics. Therefore if y coincides with x in that region, i.e. $d(y, x) < n^{-m}$, y will also be periodic with the same period. To put it differently, there is a finite word b of length $l \leq m$ such that all sequences of the form AbC are periodic with period T. But this is an open 'cylinder' set P of periodic points, with measure

$$\mu(P) = n^{-1} \geq n^{-m}.$$

QED.

   So periodic points with $S(x) \neq 0$ are isolated, while an x with $S(x) = 0$ is associated with a block of periodic points of non-zero measure.

   To calculate the total measure of all of P's images under Φ (all the periodic orbits associated with x), we need the following.

Theorem 5. Suppose Φ is a 1-1 generalized shift, and let P be an open set of periodic points x of T, with $S(x) = 0$ for all $x \in P$. Suppose further that F and G are constant on P and on all of P's images. Then all of P's images are disjoint, so that the total measure of the orbit of P is $T\mu(P)$.

Proof. Clearly $\Phi^t P = P$. Suppose $P \cap \Phi^i P \neq \varnothing$ for some $i < T$, $i > 0$. Now since F is constant on all of P's images, S_t is also, for any t; but P and $\Phi^i P$ overlap, so $S_i(P) = S_i(\Phi^i P)$. But then we have

$$S_{2i}(P) = S_i(P) + S_i(\Phi^i P) = 2S_i(P).$$

In fact, since $\Phi^{ni} P \cap \Phi^{(n+1)i} P \neq \varnothing$ for any n, we have by induction

$$S_{ni}(P) = nS_i(P) \text{ for any } n.$$

But $S_T(P) = 0$, so $S_i(P)$ is bounded by $T \max |F|$; so $S_i P$ must be 0.
   Now let $x \in P$. We have $S_i(x) = 0$, so

$$\exists y \in (A \cup \{\phi\})^Z : \forall x \in P : \Phi^i(x) = x \oplus y$$

where $y = G(P)$ (since G is constant on P). From the definition of '$\oplus$', it is easy to see that
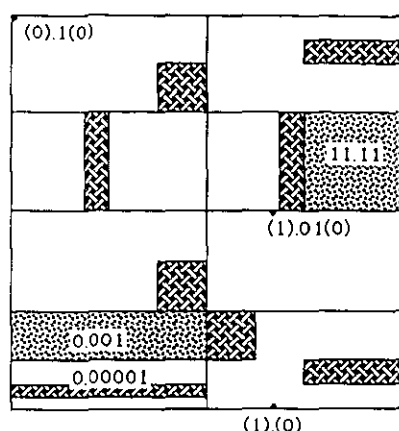
$$\Phi^i(x) = x \oplus y = x \oplus y \oplus y = \Phi^i(x \oplus y).$$

But Φ is 1–1, so

$$x = x \otimes y = \Phi^i(x).$$

Then x is of period $i < T$ and we have a contradiction. So all the images are disjoint. QED.

Figure 5. The period-2 and period-δ cylinders of example 2, and its three fixed points.

This example does in fact have periodic cylinders: in fact, any sequence of the form $X0.0^{2i}1Y$ for $i \geqslant 1$ is periodic. Here are the first few: (the first two are shown in figure 5 in the unit square, along with the isolated periodic points).

cylinder	period T_i	T_i/T_{i-1}	$\sum_{j \leqslant i} \mu(\text{orbit}_j)$
0.001	2		0.125
0.00001	8	4	0.25
0.0^61	28	3.5	0.359 375
0.0^81	96	3.43	0.453 125
$0.0^{10}1$	328	3.417	0.533 203
$0.0^{12}1$	112 0	3.414 6	0.601 563
$0.0^{14}1$	382 4	3.414 29	0.659 912
$0.0^{16}1$	130 56	3.414 22	0.709 717
$0.0^{18}1$	445 76	3.414 216	0.752 228
$0.0^{20}1$	152 192	3.414 213 9	0.788 513
$0.0^{22}1$	519 616	3.414 213 6	0.819 485
$0.0^{24}1$	177 4080	3.414 213 57	0.845 921
These converge to ∞		$2 + \sqrt{2}$	1

$(\mu(\text{orbit}_i) = T\mu(P_i) = Tn^{-d_i} = T2^{-(2i+1)}.)$

We see that the time to traverse a distance d grows as

$$T \propto \kappa^d$$

with $\kappa^2 = 2 + \sqrt{2}$ in this case. This value of κ is so simple that there must be an explanation for it. In fact, looking at the system in figure 6, we see that the generic behaviour is a fractal build-up to the left of a sort of 'scaffolding'. Using the labelling in that figure, we get the following recurrence relations on the 'busy times' (ignoring terms linear in i):

$$a_i = b_{i-1} + a_{i-1}$$

$$b_i = a_i + 2b_{i-1}$$

where a_i and b_i approximate $B(2i)$ and $B(2i + 1)$ respectively. If we write

$$a_i = \alpha b_{i-1} \qquad b_i = \beta a_i$$
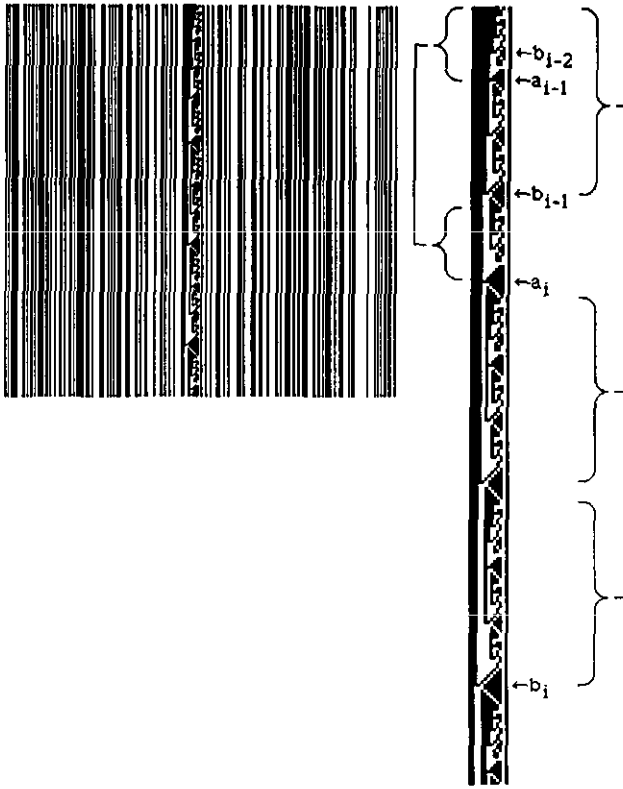
Figure 6. Example 2, showing the fractal build-up to the left. A portion is magnified and labelled, showing the recursion relations in the text.

we obtain $\alpha = \sqrt{2}$ and $\beta = 1 + \sqrt{2}$; then

$$\kappa = \sqrt{\alpha\beta} = \sqrt{2 + \sqrt{2}}.$$

As we will see in the next example, this nice scaling is not always available. In general, it is not clear that κ is algebraic or even computable, even when it exists.

How can we estimate κ in general? Since F and G are constant on P and all its images (since they each lie entirely inside one of the map's eight components) we know that all its images are cylinder sets of length d. We also know that the DoD and DoE are always inside the specified area of the cylinder. But there are only n^d cylinder sets distinct under the shift, and each one can only be in $d - 2r$ positions and still contain the DoD. Then the system can only be in one of $(d - 2r)n^d$ different states before leaving or falling into a periodic orbit. Since the period cannot exceed this number, we have

$$T(d) \leq (d - 2r)n^d \qquad \text{so} \qquad \kappa \leq n. \qquad (3)$$

This also serves as an upper limit to the 'busy function', $B(d)$.

Amusingly, although almost every point is periodic, we have the following for

the expectation of the period:

$$\tilde{T} = \sum_{d=d_i} T\mu(\text{orbit}_d) = \sum_{d} T^2\mu(P_d) \approx \sum_{d} \left(\frac{\kappa^2}{n}\right)^d = \infty \qquad (4)$$

whenever, as in this case,

$$\kappa^2 \geqslant n.$$

In the absence of a stronger limit for κ than (3), we can expect this phenomenon to be fairly common.

Why is the set of periodic points of measure 1? Watching this system's behaviour, we see that 'barriers' exist: if we imagine the position of the origin as a pointer moving back and forth on the sequence (as the figures are drawn), there are certain environments which it cannot penetrate from certain directions. For instance, a 1 surrounded by 0s cannot be penetrated from the left; it also cannot be penetrated from the right under certain conditions; see figure 7.

We make the following definitions.

Definition. A set S is *bounded-recurrent* under a map Φ if, for any $x \in S$, the set $\{i : \Phi^i x \in S\}$, the set of times it falls back in S, has bounded gaps; i.e., x may leave S, but only for an amount of time bounded by some finite t.

Definition. A *strong barrier on the left* (*right*) is a finite word b such that the set of sequences $\{AbC\}$, where A and C are semi-infinite sequences and b appears to the left (right) of the DoD, is bounded-recurrent under Φ.

Figure 7. The first four periodic cylinders of example 2, with periods of 2, 8, 28 and 96. We see the operation of the barrier 01, which is always effective on the right and is effective of the left if the nearest one to the right of the DoD is an even number of cells away.

Definition. A *weak barrier on the left (right) with domain D*, where $D \subset A^{Z^+}$ is a set of semi-infinite sequences of non-zero measure, is a finite word b such that the set of sequences AbC where b appears to the left (right) of the DoD and $C \in D$ ($A \in D$) is bounded-recurrent under Φ. (A strong barrier is a weak barrier with $d = A^{Z^+}$.)

Then we have the following.

Theorem 6. Let Φ be a generalized shift. Suppose that Φ has, on both the left and the right, either a strong barrier or a weak barrier b with domain d such that for every $d.C$ ($A.d$) where d is a finite word extending to the left (right) of the DoD and $C(A)$ is a semi-infinite sequence, there exists a finite word u of bounded length such that $ubd.C \in D$ ($A.dbu \in D$). Then almost every point in A^Z is periodic under Φ.

Proof. Let $x \in A^Z$. Let the two barriers be b_L and b_R. We will treat the one on the left.

With probability 1, any finite word appears an infinite number of times in x to the left of the origin: therefore x can be written $A_1 b_L C_1$, $A_2 b_L C_2$, etc., where b_L appears farther to the left each time: i.e., $A_{i+1} \subset A_i$ and $C_{i+1} \supset C_i$.

If b_L is strong, its first appearance suffices to bound the motion. If b_L is weak with some domain D, we need $C_i \in D$ for some i. But by hypothesis, even if $C_i \neq D$, there is a finite probability that one of the next occurrences of b_1 will be $|u|$ cells to the left of C_i and will in fact be followed by u; i.e. that there will be some j with $C_j = u b_L C_i$ and so $C_j \in D$. The probability of this is $n^{-(|u|+|b_L|)}$. If this probability fails, the same argument applies to C_j, and so on. Since the length of u is bounded by some m, the probability of all of these failing is

$$P \leqslant \lim_{k \to \infty} (1 - n^{-(m+b_L|)})^k = 0$$

where the occurrences of b_L are labelled by k. So with probability 1,

$$\exists i : C_i \in D.$$

The same argument holds for b_R.

So almost all points are bounded on the left and right: if the bounds of the gaps in the bounded recurrence are t_L and t_R, the motion has to stay within an area of width $d + (t_L + t_R) \max |F|$ where d is the distance between b_L and b_R. Then it must have a finite period bounded by equation (3).   QED.

The conditions of the theorem are in fact satisfied by this system: 01 is a strong barrier on the right, and a weak barrier on the left. Its domain can be studied in figure 7. If $1^n.00C$ is not in D, $0011^n.00C$ is ($u = 0$); $1^n.01C$ is not in D, but $011^n.01C$ is ($u = \varepsilon$, the empty word). Almost any configuration will evolve to one of these two subsets; these correspond to the two left-moving fixed points, (1).(0) and (1).01(0), moving toward the barrier.

Note that not all domains of non-zero measure satisfy the condition of the theorem; for instance, if D is an open set specified by the values of the sequence at the DoD, nothing can be added on the left to bring a sequence into D if it is not already.

2.3. Example 3

As in example 2, but as follows:

$a_{-1}a_0a_{+1}$	F	G
000	−1	011
001	+1	101
010	+1	111
011	−1	000
100	+1	001
101	−1	010
110	+1	011
111	−1	001

This map is also 1–1, and corresponds to the map of the square shown in figure 8. The simulations shown in figure 9 show highly complex behaviour, including intermittent periodicity in the neighbourhood of various periodic points. These show up in the 'moving pointer' diagrams as propagating structures reminiscent of 'gliders' in cellular automata [9].

The first few periodic orbits (observed empirically) are:

x	T	$S(x)$
(0)1.1(0)	1 (fixed)	+1
(1).01(0)	1 (fixed)	−1
(0)10.101(0)	7	−1
(0)10.100(1)	7	−1
(10)0.1000(10)	16	2
$(10^n)0.10^{n+2}(10^n)$	$15+n$	$n+1$ (for all $n \geqslant 1$)

Some of these are visible in figure 9.

The points in the infinite family given above, starting with the period-16 point, are not 'fundamental' in that they are actually generated in some sense from the fixed point (0)1.1(0). Close inspection of figure 9 reveals that the period-17 point that occurs briefly in fact consists of the fixed point propagating to the right until it collides with a 1 (note that its normal environment is a sea of 0s), going through a complicated transition, and then reappearing, delayed by 13 steps. The fact that this particular fixed point can make a transition back to itself after colliding with a certain obstacle allows us to write down an infinite number of periodic points, in
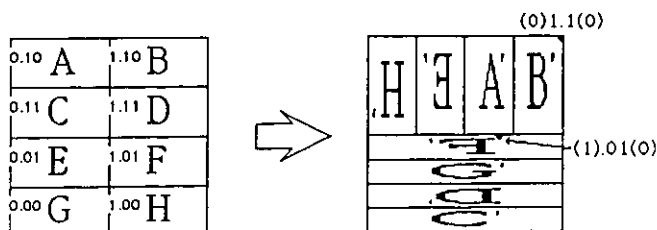

Figure 8. The map of the plane corresponding to example 3, showing the two fixed points.
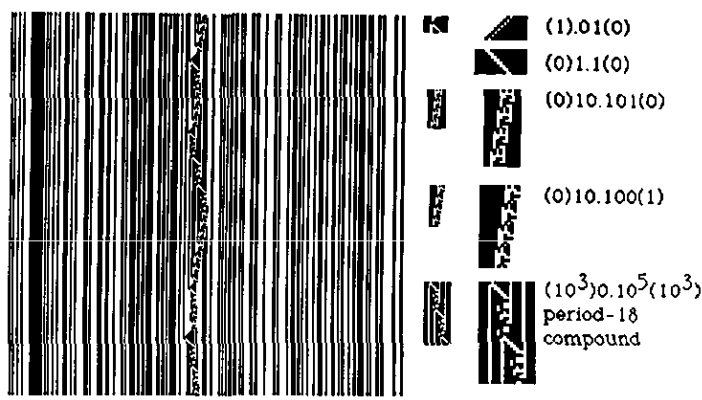
Figure 9. A simulation of example 3 with random initial conditions. The first few 'elementary' periodic points are shown at right.

which we allow it to propagate arbitrarily far in between collisions. We could even use any periodic sequence of gap sizes in between the obstacles. The period-7 points can also be used to construct more complicated examples. Several questions arise here.

(1) How can we formalize this notion of 'fundamental' versus 'compound' periodic points? Is there a well defined notion of which periodic point a given sequence is in the 'vicinity' of, perhaps based on the simplest one agreeing with x on the DoD for a full period?

(2) Is the set of periodic points dense, so that any sequence is at least temporarily involved in periodic motion of some period?

(3) When do such interactions exist that allow families of compound points to be constructed, so that an infinite number of periodic points exist?

(4) Is the number of fundamental points finite, so that the set of periodic points can be described in a finite way, using them as a sort of 'basis'?

Even for this particular Φ, these questions (2), (3) and (4) are still open; (2), as we will see in the next section, is known to be undecidable. It seems likely that (3) and (4) would be as well, if they could be stated in a well defined way.

Empirically, this system has no barriers, and no invariant proper subsets. In fact, it seems to be ergodic. No proof of any of these statements is presently available.

We have calculated the busy function $B(d)$, averaged and maximized over all initial conditions for d up to 9:

d	$\bar{B}(d)$	$\bar{B}(d)/\bar{B}(d-1)$	$B_{max}(d)$	$B_{max}(d)/B_{max}(d-1)$
1	1		1	
2	4.5	4.5	16	16
3	11.0	2.4	33	2.0625
4	18.55	1.686	74	2.2424
5	29.22	1.5752	133	1.7973
6	46.33	1.5856	244	1.8346
7	70.94	1.5312	443	1.8156
8	103.26	1.4556	634	1.4311
9	142.61	1.3810	883	1.3927

It is not clear whether or not

$$B(d) \propto \kappa^d$$

for some κ, as it was in example 2. R Durrett (private communication) conjectures that all ergodic generalized shifts are essentially diffusive, i.e.

$$B(d) \propto d^2 \qquad \text{for large } d.$$

These systems are extremely difficult to model statistically; they may be thought of as a 'random walk in an interacting environment' (RWIE), but except for 'stagnant' systems like example 1 it is impossible to take sensible averages. We will see in the next section that statistical quantities are either non-existent or uncomputable in general.

3. Turing machines

3.1. Review

A Turing machine [10] is a dynamical system in its own right, but it is rarely studied from that point of view. It is visualized as a finite-state machine moving left and right on a tape, on which is written a string of symbols in some finite alphabet (we can again assume this is $\{0, 1\}$ without loss of generality). At each time step the machine reads the symbol at its present position on the tape and, depending on that symbol and on its internal state,
   (1) changes its internal state,
   (2) changes the symbol on the tape at that point, and
   (3) moves one space left or right to its next position.
   If we have some finite set of states S, then, we can describe the Turing machine with three functions:

$$F_1 : S \times \{0, 1\} \to S$$
$$F_2 : S \times \{0, 1\} \to \{0, 1\}$$
$$F_3 : S \times \{0, 1\} \to \{+1, -1\}.$$

Then as a dynamics on the internal state $s \in S$, the tape $T \in \{0, 1\}^{\mathbb{Z}}$, and the machine's position $i \in \mathbb{Z}$ on the tape, we have

$$s \mapsto F_1(s, T_i)$$
$$T_i \mapsto F_2(s, T_i)$$
$$i \mapsto i + F_3(s, T_i).$$

The set

$$\mathcal{I} = S \times \{0, 1\}^{\mathbb{Z}} \times \mathbb{Z} = \{(s, T, i)\}$$

is called the set of 'instantaneous descriptions', and we will refer to the quadruple

$$(s, F_1, F_2, F_3)$$

as a *Turing machine M*.

The significance of these systems lies in their ability to perform computations: Turing showed that they are capable of performing any finitely describable calculation, at least according to any method of description anyone has come up with. To be more precise, for any finite program P in any existing computer language, be it FORTRAN, Pascal or whatever, there exists a Turing machine $M = (S, F_1, F_2, F_3)$ such that its evolution corresponds to the running of that program, perhaps with some ratio between instructions in P and time steps in M. In fact, mathematicians regard a given calculation as possible ('computable') if a Turing machine can perform it.

This property of a class of systems, usually called 'computational universality', allowed Turing to prove some very powerful results about what can or cannot be predicted about a Turing machine's behaviour. Most important is the 'halting problem'. Suppose we label some of the states in S as terminal states, representing final states of the machine announcing that its work is complete. Then an obvious question to ask is:

> Given a Turing machine and an initial instantaneous description, can we predict whether it will arrive at a terminal state in a finite amount of time, or run forever?

To put this more sharply, we define the function

$$H : \{M\} \times \mathcal{I} \to \{\text{halts, does not halt}\}$$

to encode the question 'does the machine M halt when given the initial instantaneous description $I \in \mathcal{I}$?' and ask

> Is there an algorithm to evaluate H that is guaranteed to run in a finite amount of time?

Turing's answer is

> NO.

In other words,

$$H : \{M\} \times \mathcal{I} \to \{\text{halts, does not halt}\} \text{ is not } recursive. \qquad (5)$$

Here we are using the following definition.

Definition [13, 14]. An algorithm is *recursive* if there is a Turing machine that performs it that is guaranteed to halt in a finite amount of time. A function is *recursive* if it can be calculated by some recursive algorithm. (If the function is real, we require a recursive algorithm which accepts an integer n as input and produces the first n digits of the result.)

We can also define a set A as *recursive* if there is a recursive algorithm to tell whether a given x is in A or not: i.e., if we define the 'characteristic function' χ_A as

$$\chi_A(x) = \begin{cases} 0 & \text{if } x \notin A \\ 1 & \text{if } x \in A \end{cases}$$

then χ_A is recursive.

A little reflection shows that the unsolvability of the halting problem is not surprising. I can arrange a program to search through the integers to counterexamples to Fermat's Last Theorem, so that it will halt if and only if such a

counterexample exists. Then solving the halting problem would be equivalent to solving the conjecture itself. If there were a general algorithm for the halting problem, mathematics would be very easy.

Furthermore, it is possible to construct 'universal' Turing machines M which can simulate any other Turing machine N, by accepting its description $N = (S, F_1, F_2, F_3)$ in code on its initial tape. The behaviour of these universal machines includes in some sense the behaviour of all Turing machines, including itself! Since specifying their initial data amounts to specifying an N as well, we can say

$$\exists M \text{ such that } H_M : \mathcal{I}\{\text{halts, does not halt}\} \text{ is not recursive.} \qquad (6)$$

Note the distinction between (5) and (6): (5) says that there is no algorithm that works for all M to predict the evolution of a given initial state, while (6) says that there are specific M which are unpredictable, even with an algorithm specific to M. (There are certainly many M for which a specific algorithm does exist.) A universal machine with seven internal states and an alphabet of four symbols [10] is shown in figure 10.

So some Turing machines are predictable, while some are not. As it turns out, this in itself is undecidable! In other words,

$$\{M : H_M \text{ is recursive}\} \text{ is not recursive.} \qquad (7)$$

Many problems related to the basic halting problem are also undecidable, such as [13]: Does M halt for all inputs? For no inputs? For an infinite number of inputs? For a set of inputs of non-zero measure? Do two Turing machines M and N halt for the same inputs? In addition, there is no recursive function $f(M, x)$ that gives an upper bound on the number of steps x takes to halt under the Turing machine M (if it does halt.)

The most powerful form of the halting problem is Rice's theorem.

Definition. A *partial recursive function* is a function that can be calculated by a Turing machine, with no restrictions on whether or not it halts. In other words, suppose we have a Turing machine M: we put it in some initial state with some input I on the tape, and let it run. It defines a function F_M as follows: if M halts, the sequence on the tape is the output $F_M(I)$, and if it never halts then F_M is undefined for that input. Then:

Rice's Theorem. Let C be a class of partial recursive functions. Then C is not recursive unless it is the empty set, or the set of all partial recursive functions.

In other words, any property that partial recursive functions can have, like being 1–1, or onto, or having an infinite domain or range, or being recursive, is undecidable unless it is trivial (i.e. true for all partial recursive functions or for none.) This proves all the undecidabilities mentioned above in one swell foop.

Corresponding to partial recursive functions, we have a wider class of sets containing the recursive sets.

Definition. A set is *recursively enumerable* (RE) if there is a recursive algorithm (i.e. Turing machine) which prints out a growing list of elements in the set, and which will eventually print out any given element. In other words, a set A is RE iff it is a union of countably many finite sets A_i such that there is a recursive procedure for producing each one.
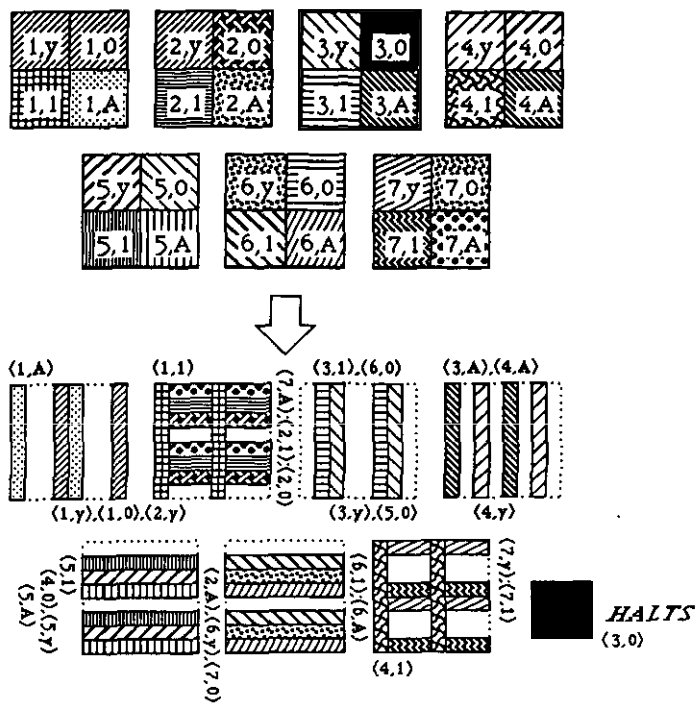
Figure 10. a map of the plane conjugate to Minsky's universal Turing machine with internal states $S = \{1, \ldots, 7\}$ and tape alphabet $a = \{y, 0, 1, A\}$ and the following transitions:

	1	2	3	4	5	6	7
y	0L	0L/1	yL	yL	yR	yR	0R
0	0L	yR	HALT	yR/5	yL/3	AL/3	yR/6
1	1L/2	AR	AL	1L/7	AR	AR	1R
A	1L	yR/6	1L/4	1L	1R	1R	0R/2

where L and R indicate shifting left and right, and the internal state stays the same unless a new one is indicated (reprinted from [7]).

It then turns out that the domains, ranges, images and inverse images of partial recursive functions are RE sets. Thus any non-trivial property of RE sets is undecidable by Rice's theorem. It also turns out that 'type 0 languages', the languages producible by a finite generative grammar [8], are identical to RE sets.

Now in terms of dynamical systems, these questions concern things like basins of attraction: for instance, $B = \{x : H_M(x) = \text{'halts'}\}$ is the set of initial conditions that end up in the 'halt' state, corresponding to a completed program. These basins are arbitrary RE sets; (5) and (6) show that these basins are, in general, non-recursive; i.e., there is no algorithm that will tell us, in a finite time, whether or not a point is in them. Statement (7) shows that there is no algorithm to tell us whether or not there is an algorithm! We see from Rice's theorem that questions like 'does B have measure greater than μ' or 'is B dense' (in the space of sequences) are also undecidable. We will show later on that this represents a kind of chaos qualitatively different from that which is usually studied in dynamical systems.

We also find that statistical questions about Turing machines are extremely hard to answer. The most dramatic case is Chaitin's number Ω [11], the probability that a given universal Turing machine will halt, given random input—this is simply the measure of the basin of attraction to the 'halt' state. Chaitin has shown that this number is uncomputable because it could act as an 'oracle' for the halting problem; knowing the first few thousand or ten thousand digits would suffice to answer most unsolved questions in mathematics! Attempts to average over all possible Turing machines with a given number of internal states and tape symbols can succeed only up to the smallest universal machine, after which such averages will contain Ω and hence be uncomputable. Thus it is very difficult to speak coherently about a 'generic' or 'random' Turing machine.

The reader may be curious about generalizations of Turing machines: being able to move more than one space at a time, having more than one tape or tape 'head', moving on a two- or more-dimensional lattice, etc. In fact, the Turing machine has *remarkable closure properties: all of these seemingly more powerful versions can be* simulated by a single ordinary Turing machine. This is another manifestation of the Turing machine's universality; it may take a long time, but it is capable of any computation whatsoever.

3.2. *The map from Turing machines to generalized shifts*

We now have the following map from Turing machines to generalized shifts.

Theorem 7. For any Turing machine M, there is a generalized shift $\Phi = (F, G)$ conjugate to M by a map $\mathscr{C}$.

Proof. We wish to absorb the internal states S into the tape so that we will have a dynamics on the tape alone. Let our alphabet $A' = S \cup A$ where S is M's set of internal states and A is M's tape alphabet. Then a state of the Turing machine can be mapped onto a sequence $a \in A'^{\mathbb{Z}}$ in the following way:

$$\mathscr{C} : \mathscr{S} \to A'^{\mathbb{Z}} : (s, (t_i), i) \mapsto a = \ldots t_{i-1}.st_it_{i+1}\ldots.$$

Then we let the DoD = DoE be $\{-1, 0, 1\}$, and define F and G as follows:

$$F(a) = F_3(s, t_i)$$
$$G(a) = \begin{cases} (\phi).t's'(\phi) & \text{if } F(a) = 1 \\ (\phi)s'.t_{i-1}t'(\phi) & \text{if } F(a) = -1 \end{cases}$$

where $s' = F_1(s, t_i)$ and $t' = F_2(s, t_i)$.
Then a little inspection shows that

$$M = \mathscr{C}^{-1}\Phi\mathscr{C}.$$

as we require. In fact, S and A do not even have to be disjoint for this construction to work; so the number of symbols we need in the generalized shift is

$$|A'| = \max(|S|, |A|).$$

We can then reduce this Φ to a generalized shift on two symbols if we wish by lemma 1. QED.

It should be clear that a reversible Turing machine becomes a 1–1 generalized shift under this construction. Charles Bennett [12] has shown how to construct for any Turing machine, a reversible one which performs the same computations. Therefore 1–1 generalized shifts are just as powerful as generalized shifts in general.

We also have the following.

Theorem 8. Any generalized shift Φ on an alphabet A can be simulated by a Turing machine with a number of internal states

$$|S| \leq (|\mathrm{Im}\, F| + 1)\frac{n^w - 1}{n - 1} + 2(\max |F| - 1)$$

and a number of time steps per time step of Φ

$$\tau/t \leq 2(w - 1) + \max |F|$$

where $n = |A|$ and $w = |\mathrm{DoD}|$.

Proof. This is done by constructing a Turing machine which scans the DoD, ascertains which of the n^w possible states it is in, replaces them according to G, and then moves to its next position according to F. Let the Turing machine have the following set of internal states, where all $a_i \in A$:

$$
\begin{aligned}
S = \{&r, r_{a_1}, r_{a_1 a_2}, \ldots, r_{a_1 a_2 \ldots a_{w-1}} \mid \forall a_1, a_2, \ldots, a_{w-1}\} \\
&\cup \{w_{a_1}, w_{a_1 a_2}, \ldots, w_{a_1 a_2 \ldots a_{w-1}} \mid \forall a_1, a_2, \ldots, a_{w-1}\} \times (\mathrm{Im}\, F) \\
&\cup \{R_2, R_3, \ldots, R_{\max F}, L_2, L_3, \ldots, L_{\max(-F)}\}.
\end{aligned}
$$

Then let these states have the following transitions:

state s	$F_1(s, a)$	$F_2(s, a)$	$F_3(s, a)$
r	r_a	a	$+1$
r_{a_1}	$r_{a_1 a}$	a	$+1$
$r_{a_1 a_2}$	$r_{a_1 a_2 a}$	a	$+1$
$\vdots$			
$r_{a_1 \ldots a_{w-1}}$	$(w_{G_1 \ldots G_{w-1}}, F(a_1 \ldots a_{w-1}a))$	$G_w(a_1 \ldots a_{w-1}a)$	-1
$\vdots$			
$(w_{g_1 g_2 g_3}, f)$	$(w_{g_1 g_2}, f)$	g_3	-1
$(w_{g_1 g_2}, f)$	(w_{g_1}, f)	g_2	-1
$(w_{g_1}, \pm 1)$	r	g_1	± 1
$(w_{g_1}, f \geq 2)$	R_f	g_1	$+1$
$(w_{g_1}, f \leq -2)$	$L_{|f|}$	g_1	-1
$R_n (n > 2)$	R_{n-1}	a	$+1$
R_2	r	a	$+1$
$L_n (n > 2)$	L_{n-1}	a	-1
L_2	r	a	-1

These states work as follows: starting in the state r, the machine moves from left to right, reading the value of the tape on the DoD. It then goes to the state $w_{G_1 \ldots G_{w-1}}$ where G is the word the DoD is to be replaced with, and writes it onto the tape moving from right to left. It then moves F spaces left or right to its new position using the L and R states (if it has more than one step to go) and restarts the loop.

Then the evolution Φ will be carried out in the stated amount of time.   QED.

3.3. Undecidable properties of generalized shifts

Theorems 7 and 8 establish that generalized shifts and Turing machines are identical in their computational power, since each class can be simulated by the other. This allows us to prove the following theorem about the complexity of dynamical sets associated with GSs: first we have the following.

Definition. A class $\mathscr{C}$ of sets of sequences is *block-arbitrary* if for any RE set S there exists some $A \in \mathscr{C}$ and some cylinder set $V \subset \Sigma$ such that $A \cap V$ is homeomorphic to S.

In other words, the set of sets in $\mathscr{C}$ is not completely arbitrary, but for any RE set we can find a member of $\mathscr{C}$ homeomorphic to it inside some block. We will use the following homeomorphism: if V specifies the cells $a_{-n} \ldots a_m$, define

$$H : \Sigma \cap V \to \Sigma:$$

$$a_i \mapsto a_{i-n}(i < 0)$$

$$a_i \mapsto a_{i+m+1}(i \geq 0).$$

This removes the specified area, and glues the two pieces of the sequence together. In particular, if we let V be the area corresponding to the Turing machine's internal state as constructed in theorem 7, then H removes it and leaves us with the tape sequence.

Now we can prove the following.

Theorem 9. The following sets associated with a given generalized shift Φ are block-arbitrary.

(1) For any open cylinder set A (except the whole space), the set

$$B_A = \{x \mid \exists t > 0 : \Phi^t x \in A\}.$$

(2) For any periodic point p, the set of x that converge to p.

(3) The set of periodic points (in fact, $S(x)$ is an arbitrary partial recursive function into the integers).

(4) The set of barriers.

Proof. Let R be an arbitrary RE set (or type 0 language). Then there is some Turing machine M which halts on the set R. We will then show that for each of the sets mentioned above there is a generalized shift Φ which has R as that set.

(1) Consider a cylinder set A; it corresponds to specifying a certain finite word w in the sequence. Map M to a generalized shift in such a way that the sequence does not enter A during the running of the program (this is easy—expand the DoD to include w, and use values of the sequence other than w to encode the internal states s.) Then use G to add the following transition: simply stamp the word w on the sequence when we reach the halt state. Thus Φ falls into A if and only if M halts.

(2) Similarly to (1), but have the halt state lead into a periodic loop where G lays down a periodic pattern on the sequence and F shifts steadily in one direction or the other. (There are techniques to have the Turing machine 'watch itself' for periodic behaviour, so we can arrange that M itself never falls into a loop by mistake.)

(3) Expand the tape alphabet of M from T to $T \times T$. Let the initial state have the same sequence in each component, and let M work on the second component, calculating some partial recursive function S from sequences into the integers. If it halts with a result, it can then go back to where the original word is written on the first component of the tape, and re-copy it S spaces to the right on the tape, erasing the original; it can then go back and redo the computation. Thus a sequence x will periodic if and only if it halts, and it will have a shift number of $S(x)$ if it does.

(4) Similarly to (3), but instead let the machine simply restore the original (on both components) and move to the right (left) of it. Then a word is a left (right) barrier if and only if M halts on it.

If we look closely, we find that we have proven that as sets of tape configurations of a Turing machine, these sets are arbitrary; but we have to specify some initial machine state to carry out these programs. Thus when this state is absorbed onto the tape to form a GS as constructed above, we will be specifying some cylinder set V. This is why, in the GS, these sets are block-arbitrary rather than completely arbitrary; we use the homeomorphism from the GS sequence to the tape configuration defined above.   QED.

In any case, it is clear that these sets are arbitrarily complicated, since we can always find a block of the Cantor set on which they are homeomorphic to arbitrary RE sets.

More specific classes can be identified; for instance, if Φ has left and right barriers, then a given pair of barriers is associated with a type 1 (context-sensitive) language [8]: since the motion of the pointer is bounded, the equivalent Turing machine becomes a 'linear bounded automaton', or LBA.

In any case, we have the following corollaries.

Theorem 10. The following questions about a generalized shift Φ are undecidable.

(1) Given a point x and an open set A, will x fall into A, i.e. is $x \in B_A$? Is B_A dense? Of measure 1? Of measure greater than μ for a given μ?

(2) Given a point x and a periodic point p, will x converge to p? Will a dense set of points converge to p?

(3) Is the set of periodic points on a given cylinder infinite? Dense? Of measure 1?

(4) Does Φ have barriers?

(5) Does Φ have sensitive dependence, i.e. is Φ chaotic?

(6) What is the form of the busy function $B(d)$ for large d? Does it exist for all d?

Proof. (1) through (4) follow directly from theorem 9 and Rice's theorem; (5) follows from theorem 11 (stated below), which states that Φ has sensitive dependence iff it has no periodic x with $S(x) = 0$, and the fact that the set of such x is an arbitrary RE set since S is an arbitrary partial recursive function. (6) is then intimately related to (4) and (5); for instance, unless Φ has sensitive dependence $B(d)$ is undefined for greater than a certain d.   QED.

These undecidabilities can be thought of as properties of one part of the phase space of a GS, or about the return map to V.

We add a note here about the difficulty of establishing undecidability for certain global properties, e.g. the set of periodic points as whole. We would like to remove

the phrase 'on a given cylinder' from question (3) above, but to do that we have to think much harder about the actual dynamics of the Turing machine, as opposed to the task it carries out. For instance, imagine a Turing machine which generates the binary expansion of π on the tape. Clearly it is not intended to fall into a periodic loop, so it has no periodic points that start with the right initial machine state (i.e. in the cylinder V). However, some of its intermediate states will involve things like keeping track of how many digits it has produced, performing additions and multiplications, moving markers on the tape, etc. It is not at all clear that we can assure ourselves that these intermediate tasks cannot fall into loops; in fact it seems quite likely that they can. Thus periodic orbits could arise from sub-tasks we do not really care about; controlling the dynamics of these sub-tasks would be a highly non-trivial programming problem. (Showing that global ergodicity is undecidable suffers from the same difficulties.)

3.4. Recursive enumerability and finite-time behaviour

To clarify the nature of these sets, it is nice to show directly that they are recursively enumerable. For the set of periodic points, we simply consider points in order of increasing period: $t = 1$ (fixed points), $t = 2$, etc. For each of these finite periods t, only a finite number of cells $m \leqslant w + t \max |F|$ are relevant to the dynamics, as in theorem 4. Then we need only consider points of the form AbC, where $|b| = m$ and A and C are each periodic with period $S(x) \leqslant t \max |F|$; but there are a finite number of such points x, each of which we can check individually by iterating Φ t times and seeing if they're the same. So we can find all the fixed points, all the period-2 points, etc.

For a basin of attraction of some open set A, we simply iterate backwards for one step, two steps, etc.; at each stage there are a finite number of pre-images bounded by p^t where $p \leqslant n^w$ is the maximum index of the map. Then we can find all x such that $\Phi(x) \in A$, $\Phi^2(x) \in A$, etc.

So we see that the finite-time behaviour of these systems is recursive; but the complexity grows with each time step so that the long-time behaviour is not recursive, but only recursively enumerable. Then questions about the long-time behaviour are undecidable. This is similar to the behaviour of cellular automata [15, 18].

4. Discussion of unpredictability

At this point, many readers may not be clear on the nature of these systems' unpredictability, or may doubt that anything different from the usual 'chaos' is being claimed. In fact it is radically different, in a well defined sense.

The shift map σ is chaotic in that two initially close initial conditions diverge exponentially, so that errors in the description or preparation of the system are magnified as we follow its evolution into the future. This means that initial data with an error of size δ serve to accurately predict the system only for a time

$$t \approx \log(D/\delta)/\lambda \tag{8}$$

where D is the size of the system (≈ 1 for the shift) and λ is the Liapunov exponent, the log of the largest eigenvalue of the map. (For the standard shift on n symbols, λ is simply $\log n$.) We may call this property *sensitive dependence* on *initial conditions* [7].

However, if the initial data for the shift were known exactly in terms of its symbolic description, prediction would be no problem: to know the value of a particular cell t time steps in the future, simply look up the cell t spaces to its right. In a certain sense this is an immediate process; no complicated calculations are required. It is the equivalent of a closed-form solution $x(t)$ to a differential equation: to get x at time t, just plug t into the formula. In this sense the shift map is very simple and 'unchaotic'.

Now consider a generalized shift or Turing machine. We have the following theorem.

Theorem 11. A generalized shift Φ has sensitive dependence iff it has no periodic cylinders, i.e. periodic points x with $S(x) = 0$.

Proof. We have shown in equation (3) that if the busy function $B(d)$ exceeds a certain maximum, the system has fallen into a periodic orbit with $S(x) = 0$. Therefore, if no such orbits exist, $B(d)$ exists and is finite for all d and all initial x. Then if between two close initial conditions there is a discrepancy d cells away from the origin, it will wander into the DoD in a finite time $t = B(d)$, whereupon the difference between the two will be of order 1.

Specifically, if the initial distance between two sequences is δ, they will almost always diverge completely in an amount of time

$$t \approx B(-\log_n \delta) \tag{9}$$

since the size of an error d cells away from the origin is n^{-d} according to the standard metric.   QED.

Here we have a more general form of (8); the divergence is subexponential since $B(d)$ is supra-linear (recall that $B(d) = d$ for the shift) but the qualitative features are the same. So these systems often share this sense of unpredictability with the shift map.

However, these systems also have a much stronger form of unpredictability; even if the initial conditions are known *exactly*, the long-time behaviour is undecidable. You can always simulate the evolution step-by-step to see what will happen, but you can never 'jump ahead'; there is no short-cut. In other words, sensitive dependence means you need more and more information to go into the future; this 'algorithmic chaos' or 'complexity' means you need to do more and more complicated computations.

For instance, theorems 9 and 10 establish that even if the symbolic description of a point is known exactly, it is undecidable whether or not it is periodic, whether or not it lies in a certain basin of attraction, etc. Sets of periodic points and attractors then become non-recursive sets, with a kind of structure much more complex than fractals or multifractals. Even qualitative properties like sensitive dependence (and presumably ergodicity) become undecidable, so that no finite classification of generalized shifts according to these properties can every exist. (These questions can be completely answered for certain subclasses of generalized shifts, but only if we

restrict ourselves so much that they are incapable of computation: e.g. the stagnant ones.)

In physical terms, the behaviour of these maps is hard to pin down because they have no well-defined scaling behaviour. This means that at any scale in time or in resolution of the phase space, there is always more qualitative behavior just around the corner; no finite experiment is sufficient to divine the nature of the basins of attraction or of the set of periodic points. This again corresponds to the fact that these sets are recursively enumerable but not recursive; you can produce longer and longer lists of points or orbits, but no regularity will ever emerge and allow you to caputre the entire set in one swell foop.

These properties clearly put these systems at a more 'chaotic' level than systems like the horseshoe map or baker's transformation, even though they can be represented as piecewise linear maps of the plane with a finite number of components and with everything rational.

If the reader still doubts, she may consult figure 10, where we exhibit a map of the plane that is equivalent to a universal Turing machine designed by Minsky [10]; it has a basin of attraction B consisting of those finite programs that eventually halt! This is clearly not a set with a simple description, like a fractal or multifractal. It would not be hard to construct a point, with finitely described, rational coordinates, which searches for counterexamples to Fermat's Last Theorem as it evolves. Is it in B or not? Also consider the fact that if you knew what fraction Ω of initial points ended up in the 'halt' area, you would be able to answer any finitely stateable existence problem of mathematics.

5. Time series

In theorem 9, we showed that periodic points and basins of attraction are arbitrary recursively enumerable sets, since they correspond to the set of initial configurations on which some Turing machine will halt. Their properties are thus the properties of the beginning and end of a computation, i.e. the domain and range of some partial recursive function. It is this kind of property that Rice's theorem refers to, and identifies as undecidable.

However, many properties dealing with the Turing machine itself, as opposed to the computation it eventually performs, are decidable. (Recall from above the difficulty with proving undecidability for various global properties that involved the Turing machine's internal workings.) In fact, the *time series* generated by a Turing machine (for instance, its internal state, or the value of the tape at the current position of the 'head') are described by a somewhat simpler kind of language, the *context-sensitive* languages [8]. Among other things, they are recursive, i.e. there is an algorithm to check a string for membership.

We can immediately see how such an algorithm would work; the Turing machine can only move a distance t in time t, so to check whether it can produce a certain time series we just have to simulate n^t different initial conditions for t steps and see if it does or not. Thus the time it takes to check whether a series of length t is producible or not is bounded by

$$\tau < tn^t$$

although there are clearly more efficient methods than just simulating every possible initial condition all the way through.

We can also characterize the set of time series as the intersection of two *context-free* languages, which we now define [8].

Definition. A *Push-Down Automaton* (PDA) is rather like a Turing machine, in that it consists of a 'box' with a finite number of states and a 'tape' it can use as memory; but the tape is only a one-sided 'stack' and the box is only allowed access to the end of it. It can read the top character, and can add or remove a finite number of characters. The other difference is that its dynamics on the set of instantaneous descriptions

$$\mathcal{I} = S \times A^{Z+}$$

is not deterministic, but is controlled by characters on an 'input tape'. The set of words on this input tape that cause the PDA to begin and end with an empty stack is then a *context-free* language.

With coding tricks like those used in the previous section, we can absorb the box onto the stack, and produce a (non-deterministic) dynamics called a *generalized one-sided shift* (GOS). Like a GS, depending on the values of the sequence on some finite DoD it may be shifted left or right, and a finite number of cells may be changed or added at the end. The itinerary defined by the sequence's values on the DoD is then a context-free language. (See [5] for a discussion of GOSs and one-dimensional maps that are conjugate to them.)

Now suppose we would like to know the time series produced by some GS Φ, partioned by the DoD. We give a construction of two GOSs, Φ_L and Φ_R, such that the intersection of their two languages is the time series language of Φ. We do this as follows: each GOS keeps track of one side of the sequence. When the DoD moves to the right (say), Φ_R shifts its one-sided sequence to the left, removing cells, and Φ_L shifts its one-sided sequence to the right, adding more cells to simulate the introduction of new cells that were previously to the right of the DoD. Then the intersection of the allowed itineraries of Φ_L and Φ_R is just the set of allowed itineraries of Φ.

As an example, we find two context-free languages whose intersection is the language generated by example 1 above. The two GOSs are as follows:

$$\Phi_R : .0a \mapsto .a, \ .1a \mapsto .x0a$$
$$\Phi_L : .1a \mapsto .a, \ .0a \mapsto .x1a.$$

(Here a is a one-sided sequence, and $x \in \{0, 1\}$ is a new digit.) This shows how 0s turn to 1s and shift left, and 1s turn to 0s and shift right. These two GOSs are identical to each other if we switch the 1s and 0s.

Now we ask what itineraries Φ_R (say) generates. If we start with $.1a$, new digits will be brought in until a 0 appears; suppose n 1s appear first. Then the itinerary consists of 1s for n steps, and we are left with $.0^{n+1}a$. Now the sequence will shift right for m steps, during which time the itinerary consists of 0s, until the first 1 in a is brought to the DoD. But clearly $m \geq n + 1$, or $m > n$. So we have a cycle of 1s followed by a larger number of 0s. So the language generated by Φ_R is

$$\{0^{s_1}1^{s_2}0^{s_3} \ldots \mid s_i < s_{i+1}, \ i \text{ even}\}.$$

By the symmetry between Φ_L and Φ_R, we known that Φ_L produces the same language, but with the 0s and 1s reversed; i.e.

$$\{0^{s_1}1^{2_2}0^{s_3} \ldots \mid s_i < s_{i+1}, \ i \text{ odd}\}.$$

The intersection between the two is then

$$\{0^{s_1}1^{s_2}0^{s_3} \ldots \mid s_i < s_{i+1}\}$$

the same language that was discussed above.

6. Smooth maps and flows

So far our maps have been defined on blocks of the Cantor set, with gaps in between on which they are undefined. We can close these gaps as we did in the figures, but then they become discontinuities between blocks. We add a few comments here about 'filling in' these gaps and making the map smooth, and embedding the resulting map in a flow on some manifold.

Theorem 12. For any invertible GOS Φ, there exists a C^∞ diffeomorphism f on R^2 whose action on a Cantor set is conjugate to Φ.

Proof. First, any GOS Φ is conjugate to one Φ' whose DoD is entirely to the right of the origin. This is because we can shift the DoD to the right by conjugating with the shift map:

$$\Phi' = \sigma^{-r}\Phi\sigma^r.$$

Then, using Φ', lemma 0 gives us a piecewise linear map $\tilde{\Phi}$ on the Cantor set in the unit square. But since the DoD is to the right of the origin, the only dependence of F and G is on the y coordinate; so the map's components are all horizontal blocks. Call them the b_i.

Now the images of the b_i with $F > 0$ are single blocks of the Cantor set, squished horizontally; but those with $F < 0$ are squashed vertically and broken into 2^{-F} blocks by the reverse shift. We need to extend the map smoothly across the vertical gaps in the b_i so that their images are simply connected.

We can do this most easily if we use the horseshoe map as our conjugacy in lemma 0. It can be extended to a diffeomorphism in a disk, as shown in figure 11 (here we have turned it on its side, to shift to the right). Then any iterates of horizontal blocks are disjoint if they are disjoint on the Cantor set; i.e. if they do not overlap as blocks, the gap images connecting them will not overlap either. This is proved as follows: consider two iterates $\sigma^{-p}b_i$ and $\sigma^{-q}b_j$, with (say $p \leq q$. Then iterate the diffeormorphism backwards (shift to the left) until you have b_i and $\sigma^{-(q-p)}b_j$. These are clearly disjoint if they are disjoint as blocks, since b_i has no gap image to overlap b_j's gap images.

So we have successfully smoothed the map across the vertical gaps in the b_i; now the only gaps left are those between the b_i, which are horizontal strips. We can extend the map to these as shown in figure 12, by stretching them around until they connect the top and bottom edges of the blocks' images. The blocks' images themselves do not overlap, since the GOS is invertible; and the gaps' images can always be drawn without any overlap, for the following reason. Label the blocks' images B_i, $i = 1, \ldots, k$ from top to bottom, and label the gaps' images G_i, $i = 1, \ldots, k - 1$. Then draw the gaps' images in order, connecting B_1 to B_2, B_2 to B_3, etc. At each stage we have to connect the bottom edge of B_i to the top edge of

C Moore

(a)

(b)
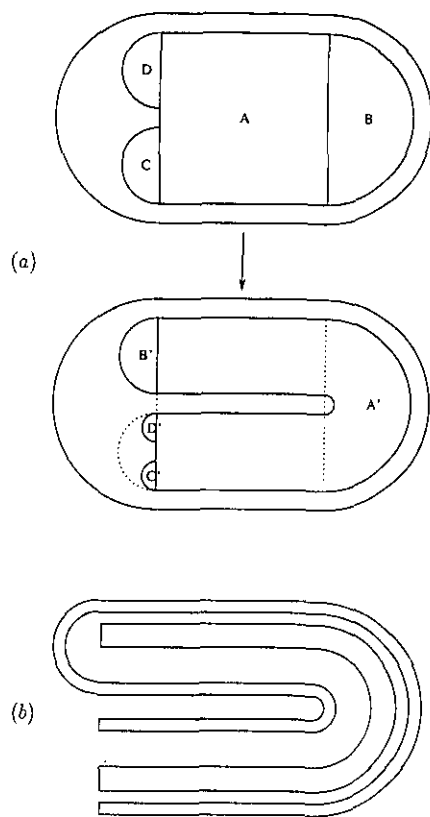
Figure 11. (a) The horseshoe embedded in a diffeomorphism of a disk. (b) Two disjoint images of horizontal blocks (one after one iteration, the other after two).
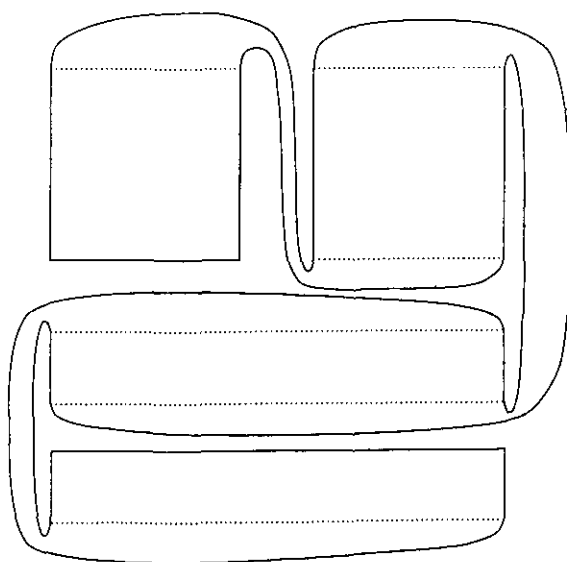
Figure 12. Stretching the map to connect the blocks.

B_{i+1}, without crossing the already defined image

$$A_i = \left(\bigcup_{j=1}^{i} B_j \right) \cup \left(\bigcup_{j=1}^{i-1} G_j \right)$$

or any of the other B_j. But A_i and all the B_j are homeomorphic to disks; so there is no difficulty in finding a path through the remaining space (the complement of their union) to connect the next gap. Clearly this can be done as smoothly as we want, up to C^∞.

Now what we have is a smooth map f from the unit square $[0, 1]^2$ into R^2, since, like the horseshoe, some of the gaps will have to fall outside the square. We can extend this to a neighbourhood of the square by thickening the image, until it gives us a diffeomorphism of a disk D into itself; we can extend this outside with any map that matches it smoothly on the boundary, to all of R^2 or just to a diffeomorphism of a larger disk D' *onto* itself. QED.

We can turn this into a flow in R^3:

Corollary. Since the map we just constructed is homotopic to the identity, there exists a smooth flow in a torus $D' \times S^1$ which has f as its Poincaré section; this can be extended to a flow in R^3.

For non-invertible GSs, we need one more dimension.

Theorem 13. For any GS Φ, there is a 4-manifold M and a smooth flow v on M such that the Poincaré map of the attractor is conjugate to Φ.

Proof. First we need to smooth the map. If the GS is not invertible, we lose nothing by stretching the gaps over the blocks, since the blocks overlap anyway. Thus we can smooth the map using some algorithm like the following:

First of all, make the map continuous by linearly interpolating in the gaps. In a gap between two blocks, the map f is defined along two opposite edges; in between we can write

$$f(x, y) = xf(1, y) + (1 - x)f(0, y)$$

where x and y are affine variables defined to vary between 0 and 1 in the gap. In the corner gaps between four squares, the map is defined in the corners: then write

$$f(x, y) = xyf(1, 1) + x(1 - y)f(1, 0) + (1 - x)yf(0, 1) + (1 - x)(1 - y)f(0, 0).$$

Then we have extended f to a continuous function throughout the square.

Now extend f to a smooth function by smoothing the above construction. If we extend the blocks some distance into the gaps first, we can cap the discontinuities in the first derivative with some integrated sigmoid function, which can be as smooth as we like (up to C^∞); see figure 13.

Thus we have obtained a smooth non-invertible map f in R^2. Now define a branched 3-manifold M' on $R^2 \times [0, 1]$ by identifying

$$((x, y), 0) \sim (f(x, y), 1).$$

Define a semiflow on this manifold by $dz/dt = 1$ where z is the third coordinate; clearly f is the Poincaré section of this flow at $z = 0$.
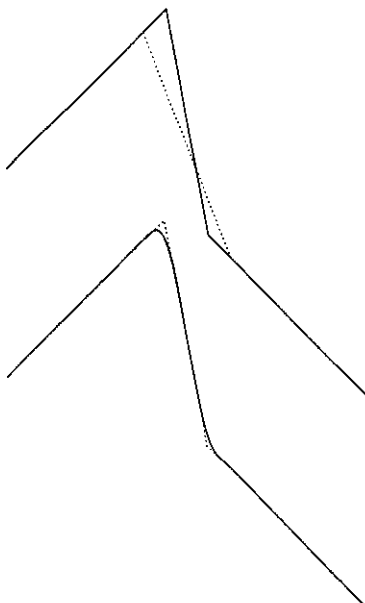
Figure 13. Smoothing the gaps by linear interpolation, then extending the blocks somewhat into the gaps, then smoothing with a sigmoid function.

We can then thicken this branched manifold by some amount ε in a fourth direction, and contract along an invariant foliation in that direction to cause the branches of M' to approach each other exponentially. This gives us a smooth flow v on some 4-manifold M, in which M' is an collapsing of the attractor along the stable foliation.   QED.

Increasing the dimension by 2 in this way is just like the Lorenz flow: a non-invertible 1D map is the Poincaré map of a flow on a branched 2-manifold, which approximates the attractor of a flow in R^3. (Finally, if we wish to embed M' in R^n for some n, we can do it with $n \leqslant 9$ according to standard embedding theorems [19], although the construction hardly seems convoluted enough to require a dimensionality higher than 4 or 5.)

In any case, extending the map through the gaps will re-inject them into the square, thus increasing the non-wandering set in a complicated way.

We should mention again here that invertible GSs are capable of all the same computations that non-invertible GSs are [12]; for instance, the universal Turing machine shown in figure 10 could, with a considerable amount of work, be encoded as an invertible GS. So:

Corollary. There exist smooth flows in R^3 that are conjugate to universal computers.

7. Conclusion

It would be nice if we could establish that some physical systems do in fact possess this kind of unpredictability. Several comments are in order here.

Firstly, dynamical systems with many degrees of freedom that are computationally powerful, such as neural nets or cellular automata [16], have already been

discussed in physics†. Some cellular automata have been shown to have long-time behaviour not characterized by regular or unambiguous context-free languages [17], and several computationally universal (i.e., having power equivalent to a universal Turing machine) cellular automata are known, such as the 'game of life' [21] as well as some one-dimensional examples [22]. Various types of neural networks are known to be computationally universal [10], and neural networks do occur in nature; one has written this paper, using another one built by some more. Some other physical systems known to be computationally universal are gases of hard spheres [23] and electrical circuits. It is fair to say that no one knows whether hydrodynamics, e.g. the Navier–Stokes equation, is capable of computation.

These systems, on the other hand, require only two dimensions, or three for a flow; they encode an entire computer into two coordinates, by using the symbol addresses of a point in a Cantor set as the 'tape' of a Turing machine. The main consequence of this is that these systems are extremely fragile, since any perturbation will destroy an infinite number of cells on the tape. They are not hyperbolic since initial conditions diverge sub-exponentially‡, and so they are not structurally stable; whether their codimension in some sense is finite or infinite remains to be explored. (Certainly those with periodic blocks have codimension infinity, since some iterate of the map is the identity on an open set.) However, many non-hyperbolic systems occur in Nature. Finally, it is worth mentioning that the 3D flows constructed above can be expressed as billiards, or particle motion in a smooth (C^∞) 3D potential [25].

Chart. Comparing the two types of chaos discussed.

Type of chaos	Sensitive dependence	Algorithmic complexity
Systems with many degrees of freedom	the weather	neural nets
Kind of unpredictability	errors grow	undecidability
Unpredictable if you have of the initial conditions	imperfect knowledge	perfect knowledge
Low-dimensional systems	$\geqslant$1D maps $\geqslant$2D flows (non-invertible)	$\geqslant$2D maps $\geqslant$3D flows
Maps on sequences	shift	generalized shift, Turing machine
Languages produced	regular (type 3)	unrestricted (type 0)
Sets produced (e.g. basins of attraction)	fractals	non-recursive sets (recursively enumerable)
Statistics	well behaved	uncomputable
Kind of understanding	can finitely describe and classify	not much—have to work hard on each system

† In fact, we could use the Smale homeomorphism to construct maps of the plane conjugate to one-dimensional cellular automata; however, these would be nowhere differentiable, except for CAs equivalent to shifts [20].

‡ It is possible to show directly using Turing machine methods that, if a GS has an overall linear drift to one side (e.g. if the DoD drifts to the right) then it cannot do anything useful or interesting, i.e. it can only produce regular languages. But this corresponds to the hyperbolic case: so hyperbolic GSs have a Markov partition [6].

In addition to these issues, other directions for further work include:

(1) Formalizing the questions about periodic points raised at the end of section 2.3.

(2) Discussing the formation of the generalized shift structure through a series of bifurcations. This is likely to be quite difficult; it is already known that an infinite number of non-universal routes to chaos exist in two-dimensional maps [24]. If the set of periodic points is non-recursive, the bifurcations are likely to be also.

(3) Finding simple finite-dimensional physical systems with a generalized shift dynamics, such as billards or optical systems.

We close with a chart comparing the two types of chaos we have discussed.

Acknowledgments

References

[1] Smale S 1963 Diffeomorphisms with many periodic points *Differential and Combinatorial Topology* ed S S Cairns (Princeton, NJ: Princeton University Press) pp 63–80
[2] Smale S 1967 Differentiable dynamical systems *Bull. Am. Math. Soc.* **73** 473–87.
[3] Cornfeld L P, Fomin S V and Sinai G Ya 1982 *Ergodic Theory* (Berlin: Springer)
[4] Moore C 1989 C^2 diffeomorphisms of the plane with a preserved Cantor set unpublished.
[5] Moore C 1991 Generalized one-sided shifts and maps of the interval *Nonlinearity.* submitted
[6] Moore C and Hartmanis J 1989 Hyperbolicity of generalized shifts unpublished
[7] Guckenheimer J and Holmes P 1983 *Nonlinear Oscillations, Dynamical Systems and Bifurcations of Vector Fields* (Berlin-Springer)
[8] Salomaa A 1973 *Formal Languages* (New York: Academic)
[9] Gardner M 1983 *Wheels, Life, and Other Mathematical Amusements* (San Francisco: Freeman)
[10] Minsky M 1967 *Computation: Finite and Infinite Machines* (Englewood Cliffs, NJ: Prentice-Hall)
[11] Chaitin G 1975 *J. Assoc. Comput. Mach.* **22** 329; 1977 *IBM J. Res. Develop.* **21** 350, 496
[12] Bennett C H 1973 Logical Reversibility of Computation *IBM J. Res. Develop.* **17**
[13] Rogers H Jr. 1967 *Theory of Recursive Functions and Effective Computability* (New York: McGraw-Hill)
[14] Davis M 1958 *Computability and Unsolvability* (New York: McGraw-Hill)
[15] Wolfram S 1984 Computation theory of cellular automata *Commun. Math. Phys.* **96**
[16] Wolfram S 1985 Undecidability and intractability in theoretical physics *Phys. Rev. Lett.* **54** 735–8
[17] Nordahl M G 1989 Formal languages and finite cellular automata *Complex Systems* **3** 63–78
[18] Nordahl M G 1988 Discrete dynamical systems *Preprint* Chalmers Institute of Technology
[19] Takens F 1980 Detecting strange attractors in turbulence *Dynamical Systems and Turbulence, Warwick 1980.* (Lecture Notes in Mathematics) (Berlin: Springer)
[20] MacKay R S private communication
[21] Berlekamp E R, Conway J H and Guy R K 1982 *Winning Ways for your Mathematical Plays* (New York: Academic)
[22] Albert J and Culik K II 1987 A simple cellular automaton and its one-way and totalistic versions *Complex Systems* **1** 1–16
[23] Margolus N 1984 *Physica* **10D** 81
[24] Holmes P J and Whitley D C 1984 Bifurcations of one- and two-dimensional maps *Phil. Trans. R. Soc.* A **311** 43–102
[25] Moore C 1990 Unpredictability and undecidability in dynamical systems *Phys. Rev. Lett.* **64** 2354