Part II

Chaos rules

QUANDRY: all these cycles, but what to do with them? What you have now is a topologically invariant road map of the state space, with the chaotic region pinned down by a rigid skeleton, a tree of *cycles* (periodic orbits) of increasing lengths and self-similar structure. In chapter 18 we shall turn this topological dynamics into a multiplicative operation on the state space partitions by means of transition matrices of chapter 17, the simplest examples of evolution operators. This will enable us to *count* the distinct orbits, and in the process touch upon all the main themes of this book, going the whole distance from diagnosing chaotic dynamics to computing zeta functions.

1. Partition the state space and describe all allowed ways of getting from 'here' to 'there' by means of transition graphs (transition matrices). These generate the totality of admissible itineraries (chapter 17)
2. Learn to count (chapter 18)
3. Learn how to measure what's important (chapter 19)
4. Learn how to evolve the measure, compute averages (chapter 20)
5. Learn what a 'Fourier transform' is for a nonlinear world (chapter 21),
6. and how the short-time / long-time duality is encoded by spectral determinant expression for its spectrum in terms of periodic orbits (chapter 22)
7. Learn how to use short period cycles to describe chaotic world at times much beyond the Lyapunov time (chapter 23)
8. What is all this hard work good for? Deterministic diffusion and foundations of 'far for equilibrium' statistical mechanics (chapter 24)

Chapter 17

Walkabout: Transition graphs

> I think I'll go on a walkabout
> find out what it's all about [...] take a ride to the other side
> —Red Hot Chili Peppers, 'Walkabout'

In chapters 14 and 15 we learned that invariant manifolds partition the state space in invariant way, and how to name distinct orbits. We have established and related the *temporally* and *spatially* ordered topological dynamics for a class of 'stretch & fold' dynamical systems, and discussed pruning of inadmissible trajectories.

Here we shall use these results to generate the totality of admissible itineraries. This task will be particularly easy for repellers with complete Smale horseshoes and for subshifts of finite type, for which the admissible itineraries are generated by finite transition matrices, and the topological dynamics can be visualized by means of finite transition graphs. We shall then turn topological dynamics into a linear multiplicative operation on the state space partitions by means of transition matrices, the simplest examples of 'evolution operators.' They will enable us – in chapter 18 – to *count* the distinct orbits.

17.1 Matrix representations of topological dynamics

The allowed transitions between the regions of a partition $\{\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_m\}$ are encoded in the $[m\times m]$-dimensional transition matrix whose elements take values

$$T_{ij} = \begin{cases} 1 & \text{if the transition } \mathcal{M}_j \to \mathcal{M}_i \text{ is possible} \\ 0 & \text{otherwise}\,. \end{cases} \tag{17.1}$$

The transition matrix is an explicit linear representation of topological dynamics. If the partition is a dynamically invariant partition constructed from sta-

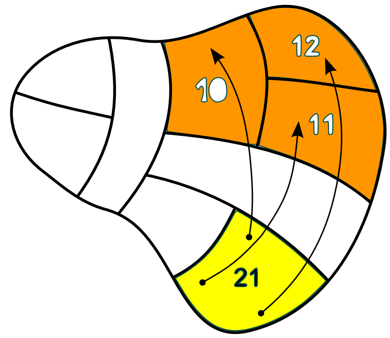


Figure 17.1: Points from the region $\mathcal{M}_{21}$ reach regions $\{\mathcal{M}_{10}, \mathcal{M}_{11}, \mathcal{M}_{12}\}$, and no other regions, in one time step. Labeling exemplifies the 'shift map' of example 14.6 and (14.13).

ble/unstable manifolds, it encodes the topological dynamics as an invariant law of motion, with the allowed transitions at any instant independent of the trajectory history, requiring no memory.

A non-negative matrix whose columns conserve probability, $_i L_{ij} = 1$, is called *Markov*, *probability* or *stochastic* matrix.

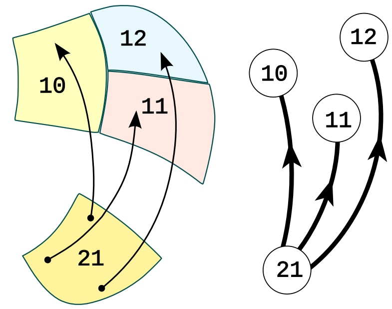


Figure 17.2: Topological dynamics: shrink each state space partition region figure 17.1 to a *node*, and indicate the possibility of reaching a region by a *directed link*. The links stand for transition matrix elements $T_{10,21} = T_{11,21} = T_{12,21} = 1$; remaining $T_{ij,21} = 0$.

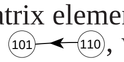
A subshift (14.15) of finite type is a *topological dynamical system* (Σ, σ), where the shift σ acts on the space of all admissible itineraries (s_k)

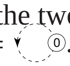
$$\Sigma = \{(s_k)_{k \in \mathbb{Z}} \;:\; T_{s_{k+1}s_k} = 1 \quad \text{for all } k\}\,, \qquad s_k \in \{a, b, c, \cdots, z\}\,. \tag{17.4}$$

The task of generating the totality of admissible itineraries is particularly easy for subshifts of finite type, for which the admissible itineraries are generated by finite transition matrices, and the topological dynamics can be visualized by means of finite transition graphs.

17.2 Transition graphs: wander from node to node

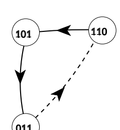
Let us abstract from a state space partition such as figure 17.1 its topological essence: indicate a partition region $\mathcal{M}_a$ by a *node*, and indicate the possibility of reaching the region $\mathcal{M}_b$, $L_{ba} \neq 0$ by a *directed link*, as in figure 17.2. Do this for all nodes. The result is a *transition graph*.

A *transition graph* consists of a set of *nodes* (or *vertices*, o r *states*), one for each letter in the alphabet $\mathcal{A} = \{a, b, c, \cdots, z\}$, connected by a set of directed *links* (*edges*, *arcs*, *arrows*). A directed link starts out from node j and terminates at node i whenever the matrix element (17.3) takes value $L_{ij} \neq 0$. A link connects two nodes, or originates and terminates on the same node. For example, if a partition includes regions labeled $\{\cdots, \mathcal{M}_{101}, \mathcal{M}_{110}, \cdots\}$, the transition matrix element connecting the two is drawn as $L_{101,110} =$ 

, whereas $L_{0,0} =$ 

. Here a dotted link indicates that the shift $\sigma(x_{011\cdots}) = x_{11\cdots}$ involves symbol 0, and a full one a shift $\sigma(x_{110\cdots}) = x_{10\cdots}$ that involves 1. A $j \to \cdots \to k$ *walk* (*path*, *itinerary*) traverses a connected set of directed links, starting at node j and ending at node k. A *loop* (*periodic orbit*, *cycle*) is a walk that ends at the starting node (which can be any node along the loop), for example

$$t_{011} = L_{110,011} L_{011,101} L_{101,110} = \tag{17.5}$$



Our convention for ordering indices is that the successive steps in a visitation sequence $j \to i \to k$ are generated by matrix multiplication from the left, $T_{kj} = \sum T_{ki} T_{ij}$.

A transition graph compactly describes the ways in which the state space regions map into each other, accounts for finite memory effects in dynamics, and generates the totality of admissible trajectories as the set of all possible walks along its links. Construction of a good transition graph is, like combinatorics, unexplainable (check page 245). The only way to learn is by some diagrammatic gymnastics, so we recommend that you work your way through the examples, exercises in lieu of plethora of baffling definitions.

example 17.1 p. 314
example 17.2 p. 315
example 17.3 p. 315

The complete unrestricted symbolic dynamics is too simple to be illuminating, so we turn next to the simplest example of pruned symbolic dynamics, the finite subshift obtained by prohibition of repeats of one of the symbols, let us say _11_. This situation arises, for example, in a billiard, and in studies of the circle maps, where this kind of symbolic dynamics describes "golden mean" rotations.

exercise 18.6
exercise 18.8

example 17.4 p. 315
example 17.5 p. 316

In the complete N-ary symbolic dynamics case (see example 17.2) the choice of the next symbol requires no memory of the previous ones. However, any further refinement of the state space partition requires finite memory.

example 17.6 p. 316

For M-step memory the only nonvanishing matrix elements are of the form $T_{s_1 s_2 \ldots s_{M+1}, s_0 s_1 \ldots s_M}$, $s_{M+1} \in \{0, 1\}$. This is a sparse matrix, as the only non vanishing entries in the $a = s_0 s_1 \ldots s_M$ column of T_{ba} are in the rows $b = s_1 \ldots s_M 0$ and $b = s_1 \ldots s_M 1$. If we increase the number of remembered steps, the transition matrix grows large quickly, as the N-ary dynamics with M-step memory requires an $[N^{M+1} \times N^{M+1}]$ matrix. Since the matrix is very sparse, it pays to find a compact representation for T. Such a representation is afforded by transition graphs, which are not only compact, but also give us an intuitive picture of the topological dynamics.

exercise 18.1

17.3 Transition graphs: stroll from link to link

(P. Cvitanović and Matjaž Gomilšek)

What do finite graphs have to do with infinitely long trajectories? To understand the main idea, let us construct an infinite rooted tree graph that explicitly enumerates all possible itineraries. In this construction the nodes are unlabeled, and the links labeled (or colored, or dotted in different ways), signifying different kinds of transitions.

A *tree graph*

contains no loops, i.e., it is not possible to return to any of its nodes by a walk along a sequence of distinct links. A *rooted tree graph* is a directed graph (its links are directed, $j \to i$), obtained from an undirected tree graph by picking a distinguished node, called the *root*, and orienting all links in the tree so that they point away from the root.

In a rooted tree graph, all nodes have exactly one parent (in-degree = 1), except for the root, which is the single "parentless" node (in-degree = 0), with all links pointing away from it. An *external node* (*leaf*) is a "childless" node, with in-degree ≥ 1, out-degree = 0.

example 17.7
p. 316

We illustrate how trees are related to transition graphs by first working out the simplest example of pruned symbolic dynamics, the finite subshift obtained by prohibition of repeats of one of the symbols, let us say _00_ . As we shall see, for finite grammars a rooted tree (and, by extension, but less obviously, the associated transition graph) is the precise statement of what is meant topologically by a "self-similar" fractal; supplemented by scaling information, such a rooted tree generates a self-similar fractal.

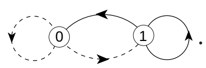
example 17.8
p. 317

17.4 Examples

Example 17.1 ***Full binary shift.*** *Consider a full shift on two-state partition* $\mathcal{A} = \{0, 1\}$*, with no pruning restrictions. The transition matrix and the corresponding transi-*

tion graph are

$$T = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \qquad (17.6)$$



Dotted links correspond to shifts originating in region 0*, and the full ones to shifts originating in* 1*. The admissible itineraries are generated as walks on this transition graph. (continued in example 17.7)*

click to return: p. 310

Example 17.2 ***Complete*** N***-ary dynamics:*** *If all transition matrix entries equal unity (one can reach any region from any other region in one step),*

$$T_c = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix}, \qquad (17.7)$$

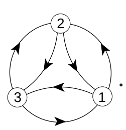
the symbolic dynamics is called complete*, or a* full shift. *The corresponding transition graph is obvious, but a bit tedious to draw for arbitrary* N*.*

click to return: p. 310

Example 17.4 ***Pruning rules for a 3-disk alphabet:*** *As the disks are convex, there can be no two consecutive reflections off the same disk, hence the covering symbolic dynamics consists of all sequences which include no symbol repetitions* 11*,* 22*,* 33*. This is a finite set of finite length* pruning rules*, hence, the dynamics is a subshift of finite type (see (14.16) for definition), with the transition matrix / graph given by*

exercise 18.1

$$T = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} = \qquad (17.9)$$



click to return: p. 310

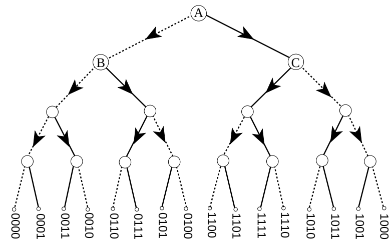
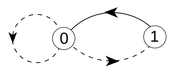


Figure 17.5: The self-similarity of the complete binary symbolic dynamics represented by a rooted binary tree: trees originating in nodes $B, C, \cdots$ (actually - any node) are the same as the tree originating in the root node A. Level $m = 4$ partition is labeled by 16 binary strings, coded by dotted (0) and full (1) links read down the tree, starting from A. See also figure 14.12.

Example 17.5 ***'Golden mean' pruning.*** *Consider a subshift on two-state partition $\mathcal{A} = \{0, 1\}$, with the simplest grammar $\mathcal{G}$ possible, a single pruned block $b = \_11\_$ (consecutive repeat of symbol 1 is inadmissible): the state $\mathcal{M}_0$ maps both onto $\mathcal{M}_0$ and $\mathcal{M}_1$, but the state $\mathcal{M}_1$ maps only onto $\mathcal{M}_0$. The transition matrix and the corresponding transition graph are*

$$T = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \qquad (17.10)$$



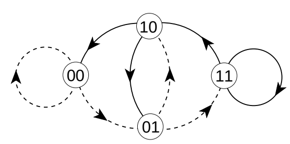
Admissible itineraries correspond to walks on this finite transition graph. (continued in example 17.8)

click to return: p. 310

Example 17.6 ***Finite memory transition graphs.*** *For the binary labeled repeller with complete binary symbolic dynamics, we might chose to partition the state space into four regions $\{\mathcal{M}_{00}, \mathcal{M}_{01}, \mathcal{M}_{10}, \mathcal{M}_{11}\}$, a 1-step refinement of the initial partition $\{\mathcal{M}_0, \mathcal{M}_1\}$. Such partitions are drawn in figure 15.3, as well as figure 1.9. Topologically f acts as a left shift (15.7), and its action on the rectangle $[.01]$ is to move the decimal point to the right, to $[0.1]$, forget the past, $[.1]$, and land in either of the two rectangles $\{[.10], [.11]\}$. Filling in the matrix elements for the other three initial states we obtain the 1-step memory transition matrix/graph acting on the 4-regions partition*

exercise 14.7

$$T = \begin{bmatrix} T_{00,00} & 0 & T_{00,10} & 0 \\ T_{01,00} & 0 & T_{01,10} & 0 \\ 0 & T_{10,01} & 0 & T_{10,11} \\ 0 & T_{11,01} & 0 & T_{11,11} \end{bmatrix} = \qquad (17.11)$$



(continued in example 18.7)

click to return: p. 310

Example 17.7 ***Complete binary topological dynamics.*** *Mark a dot '·' on a piece of paper. That will be the root of our tree. Draw two short directed lines out of the dot, end each with a dot. The full line will signify that the first symbol in an itinerary is '1,' and the dotted line will signifying '0.' Repeat the procedure for each of the two new dots, and then for the four dots, and so on. The result is the binary tree of figure 17.5. Starting at the top node, the tree enumerates exhaustively all distinct finite itineraries of lengths $n = 1, 2, 3, \cdots$*

$$\{0, 1\} \quad \{00, 01, 10, 11\}$$
$$\{000, 001, 010, 011, 100, 101, 111, 110\} \cdots .$$

The $n = 4$ nodes in figure 17.5 correspond to the 16 distinct binary strings of length 4, and so on. By habit we have drawn the tree as the alternating binary tree of figure 14.12, but that has no significance as far as enumeration of itineraries is concerned

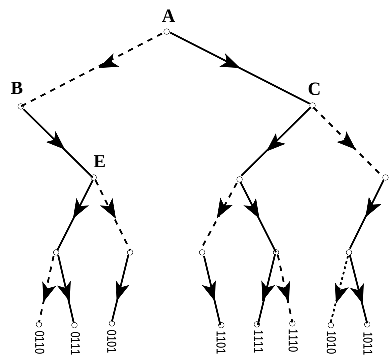
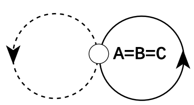


Figure 17.6: The self-similarity of the _00_ pruned binary tree: trees originating from nodes C and E are the same as the entire tree.

- a binary tree with labels in the natural order, as increasing binary 'decimals' would serve just as well.

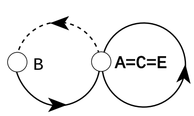
The trouble with an infinite tree is that it does not fit on a piece of paper. On the other hand, we are not doing much - at each node we are turning either left or right. Hence all nodes are equivalent. In other words, the tree is self-similar; the trees originating in nodes B and C are themselves copies of the entire tree. The result of identifying $B = A$, $C = A$ is a single node, 2-link transition graph with adjacency matrix (17.2)

$$A = \begin{bmatrix} 2 \end{bmatrix} =$$  . (17.12)

An itinerary generated by the binary tree figure 17.5, no matter how long, corresponds to a walk on this graph. This is the most compact encoding of the complete binary symbolic dynamics. Any number of more complicated transition graphs such as the 2-node (17.6) and the 4-node (17.11) graphs generate all itineraries as well, and might sometimes be preferable.

exercise 18.6
exercise 18.5
click to return: p. 311

Example 17.8 ***'Golden mean' pruning.*** *(a link-to-link version of example 17.5) Now the admissible itineraries are enumerated by the pruned binary tree of figure 17.6. Identification of nodes $A = C = E$ leads to the finite 2-node, 3-links transition graph*

$$T = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} =$$  . (17.13)

As 0 is always followed by 1, the walks on this graph generate only the admissible itineraries. This is the same graph as the 2-node graph (17.10), with full and dotted lines interchanged. (continued in example 18.4)

click to return: p. 311

Exercises

17.1. **Time reversibility.** Hamiltonian flows are time reversible. Does that mean that their transition graphs are symmetric in all node $\rightarrow$ node links, their transition matrices are adjacency matrices, symmetric and diagonalizable, and that they have only real eigenvalues?

17.2. **Alphabet {0,1}, prune** $\_1000\_$, $\_00100\_$, $\_01100\_$. This example is motivated by the pruning front description of the symbolic dynamics for the Hénon-type mapsremark 15.3.

step 1. $\_1000\_$ prunes all cycles with a $\_000\_$ subsequence with the exception of the fixed point $\overline{0}$; hence we factor out $(1 - t_0)$ explicitly, and prune $\_000\_$ from the rest. This means that x_0 is an isolated fixed point - no cycle stays in its vicinity for more than 2 iterations. In the notation of sect. 17.3.1, the alphabet is $\{1, 2, 3; \overline{0}\}$, and the remaining pruning rules have to be rewritten in terms of symbols 2=10, 3=100:

step 2. alphabet $\{1, 2, 3; \overline{0}\}$, prune $\_33\_$, $\_213\_$, $\_313\_$. This means that the 3-cycle $\overline{3} = \overline{100}$ is pruned and no long cycles stay close enough to it for a single $\_100\_$ repeat. Prohibition of $\_33\_$ is implemented by dropping the symbol "3" and extending the alphabet by the allowed blocks 13, 23:

step 3. alphabet $\{1, 2, \underline{13}, \underline{23}; \overline{0}\}$, prune $\_2\underline{13}\_$, $\_\underline{23}\,\underline{13}\_$, $\_\underline{13}\,\underline{13}\_$, where $\underline{13} = 13$, $\underline{23} = 23$ are now used as single letters. Pruning of the repetitions $\_\underline{13}\,\underline{13}\_$ (the 4-cycle $\overline{13} = \overline{1100}$ is pruned) yields the

result: alphabet $\{1, 2, \underline{23}, \underline{113}; \overline{0}\}$, unrestricted 4-ary dynamics. The other remaining possible blocks $\_2\underline{13}\_$, $\_\underline{2313}\_$ are forbidden by the rules of step 3.

References

[17.1] D.A. Lind and B. Marcus, *An introduction to symbolic dynamics and coding* (Cambridge Univ. Press, Cambridge 1995).

[17.2] A. Salomaa, *Formal languages* (Academic Press, San Diego1973).

[17.3] J.E. Hopcroft and J.D. Ullman, *Introduction to automata theory, languages, and computation* (Addison-Wesley, Reading MA 1979).

[17.4] D.M. Cvetković, M. Doob and H. Sachs, *Spectra of graphs* (Academic Press, New York 1980).

[17.5] C.J. Puccia and R. Levins, *Qualitative modeling of complex systems: An introduction to loop analysis and time averaging* (Harvard Univ. Press, Cambridge MA 1986).

[17.6] E.D. Sontag, *Mathematical control theory: Deterministic finite dimensional systems* (Springer, New York 1998).

[17.7] J. Bang-Jensen and G. Gutin, *Digraphs: Theory, algorithms and applications* (Springer, London 2002).

[17.8] P. Grassberger, "On the symbolic dynamics of the one-humped map of the interval" *Z. Naturforsch. A* **43**, 671 (1988).

[17.9] P. Grassberger, R. Badii and A. Politi, "Scaling laws for invariant measures on hyperbolic and nonhyperbolic attractors," *J. Stat. Phys.* **51**, 135 (1988).

[17.10] S. Isola and A. Politi, "Universal encoding for unimodal maps," *J. Stat. Phys.* **61**, 259 (1990).

[17.11] A. Boyarski and M. Skarowsky, *Trans. Amer. Math. Soc.* **225**, 243 (1979); A. Boyarski, *J. Stat. Phys.* **50**, 213 (1988).

[17.12] C.S. Hsu, M.C. Kim, *Phys. Rev. A* **31**, 3253 (1985); N. Balmforth, E.A. Spiegel, C. Tresser, *Phys. Rev. Lett.* **72**, 80 (1994).