SHIFT CLASSIFICATION

RACHEL LEE

ABSTRACT. The aim of this paper is to provide a brief survey of the field of shifts and then dive into the shift equivalence problem that remains unresolved to this day. We will assume little to no prior knowledge and will begin by introducing basic terminology and notation regarding shifts. We will then introduce alternate presentations of shifts that will become useful in our discussion of shift equivalence. Finally, we will discuss the progress that has been made so far in the classification of shifts.

Contents

1. Introduction

We often see shifts in the field of symbolic dynamics, in which they can be used as a tool to simplify complex dynamical systems. In essence, we are discretizing space and assigning symbols to them, then analyzing sequences of these symbols. Consider a game of billards, where we lay a labeled grid over the table. We can observe certain patterns in the symbol sequences that would result from jotting down the path of a cue ball. For one, given a symbol in the sequence, the next symbol would be one of finitely many possibilities, since the cue ball can only travel to a nearby square. The symbol for the grid on the other side of the table could never appear next. As we begin to see, we can get a sense of a dynamical system by coming up with rules for which symbol sequences can occur.

Although in real-world examples such as the billards table it is obvious to us how it differs from, say, a flowing river, mathematically defining these distinctions is a difficult task. In fact, a thorough classification of shifts has yet to be achieved. However, we have seen many important results in the mean time. As we will see, shifts can be presented in many different ways. There are intuitive ways of translating shifts into graphs and matrices and vice versa. Notably, the work around shift classification has dealt with these alternative presentations, as they are well-studied structures in their own right and interesting to work with.

Date: DEADLINES: Draft AUGUST 14 and Final version AUGUST 28, 2021.

2. Preliminaries

Let $\mathcal{A}$ denote a finite set of symbols, or an *alphabet.* The full $\mathcal{A}$-*shift* is the set of all biinfinite sequences of elements of $\mathcal{A}$ paired with a shift map σ that sends the full shift into itself, such that $(\sigma x)_i = x_{i+1}$. We can denote the full $\mathcal{A}$ shift as the pair $(\mathcal{A}^{\mathbb{Z}}, \sigma)$, where $\mathcal{A}^{\mathbb{Z}}$ denotes the set of all biinfinite sequences over $\mathcal{A}$. An element of $\mathcal{A}^{\mathbb{Z}}$ often has the following form, where the symbols are indexed by $\mathbb{Z}$.

$$\dots x_{-2}x_{-1}x_0x_1x_2\dots$$

It can be helpful to think of an alphabet as the possible states of a system or discretizations of space. Likewise, an element of X can be thought of as a time series in which $\{x\}_{i<0}$ represents the past, $\{x\}_{i>0}$ is the future, and x_0 is the current state. The shift map acts as a clock that moves future to present and present to past.

A *word* is a finite sequence of symbols. We define a *subshift* or *shift space* as a subset of a full shift that is closed under σ. It is a fact that every shift space can be uniquely characterized by a set of "forbidden" words, and it is sometimes defined as such. For a set of words $\mathcal{F}$, the shift with elements that do not contain words in $\mathcal{F}$ is denoted $X_{\mathcal{F}}$. In other words, $X_{\mathcal{F}}$ is the shift space with $\mathcal{F}$ as its forbidden words. Just as we can define a shift space by which words are not allowed, we will denote the *language* of a shift space X, or its allowed blocks, as $\mathcal{B}(X)$. It may be useful to look at only certain words of a language, say, ones of a certain length. Thus, we use $\mathcal{B}_n(X)$ to denote the set of allowed n-blocks of X.

Example 2.1. *The* ***golden mean shift*** *is the shift space of all binary sequences without two consecutive ones. In other words, it is the subshift with forbidden words* $\mathcal{F} = \{11\}$.

Example 2.2. *The* ***even shift*** *is the subshift consisting of binary sequences such that between any two ones, there is an even number of zeroes. So, we have that* $\mathcal{F} = \{10^{2n+1}1 \mid n \in \mathbb{N}\}$.

To be clear, a shift space is a pair $(X_{\mathcal{F}}, \sigma_X)$, with σ_X being the shift map restricted to $X_{\mathcal{F}}$, although the notation $X_{\mathcal{F}}$ often extends to the shift space as a whole.

Definition 2.3. A *shift of finite type*, or *SFT*, is a shift space whose set of forbidden words is finite.

The golden mean shift is an SFT, to give an example. SFT's are regarded as the class of shifts we know most about and will be the focus of much of our discussion.

In the study of shifts, we naturally get to maps between them. A *homomorphism* from one shift space to another is a continuous map f such that $f \circ \sigma = \sigma \circ f$. We define distance in this case by the shared central block of coordinates between two points. It follows that if x and y share a block of central coordinates, then σx and σy do as well, showing that the shift map and its inverse are continuous. A surjective homomorphism is known as a *quotient map*, and an injective one is known as an *embedding.* A homomorphism that is a bijection is termed a *topological conjugacy.* Note that a conjugacy is an isomorphism in the world of symbolic dynamics. It is helpful to mention now another well-studied category of shifts called *sofic shifts*, which are quotients of SFT's.

Now, we will discuss some properties of SFT's.

Theorem 2.4. *A shift space X is an SFT if and only if for $uv, vw \in \mathcal{B}(X)$. where $|v| \geq N$, for an N-step shift X, then $uvw \in \mathcal{B}(X)$.*

Proof. $\Longrightarrow$ Suppose X is an N-step SFT with a set of forbidden words $\mathcal{F}$, each of which has length $N+1$. Consider words $uv, vw \in \mathcal{B}(X)$ such that $|v| \geq N$. We know that for some $x, x' \in X$, $x_{[-in]} = uv$ and $x'_{[1,j]} = vw$, where $x_{[1,n]} = v = x'_{[1,n]}$. Since $n \geq M$, we know that no word in $\mathcal{F}$ is in $x_{(-\infty,0]}vx'_{[n+1,\infty)}$, and thus, it belongs in X. So, it follows that $uvw \in \mathcal{B}(X)$.

$\Longleftarrow$ Now, suppose that for a shift space X, whenever $uv, vw \in \mathcal{B}(X)$ and $|v| \geq M$, then $uvw \in \mathcal{B}(X)$. Let $\mathcal{F}$ be the set of $(N+1)$-blocks that don't belong to $\mathcal{B}_{N+1}(X)$. We will show that $\mathcal{F}$ makes up the set of forbidden words of X. Let Y denote the SFT characterized by $\mathcal{F}$. It is clear that $X \subset Y$. Now, we will consider some $x \in Y$. We have that $x_{[0,N]}, x_{[1,N+1]} \in \mathcal{B}(X)$. We can further deduce that $x_{[0,M+1]} \in \mathcal{B}(X)$, and even further that $x_{[0,M+2]} \in \mathcal{B}(X)$. From here, we can keep extending the word in either direction and see that $x_{[-i,j]} \in \mathcal{B}(X)$, for $i, j \geq 0$. So, we can conclude that $x \in X$. □

Maps between shift spaces can always be made into a *sliding block code*, as we will see later.

Definition 2.5. A *sliding block code* is a map $f : X \to Y$, where X and Y are shift spaces, such that the coordinates of its image are defined by $(f(x))_i = F(x_{i-m} \dots x_{i+a})$, where F takes a sequence of symbols in X and produces a symbol in Y. We call m and a the *memory* and *anticipation* of a code, respectively.

The shift map σ of a full shift is a 1-block sliding block code or, more simply, a 1-block code.

Theorem 2.6. *A conjugacy of an SFT is itself an SFT.*

Proof. Let Y be an SFT and X a shift space that is conjugate to Y. We will find an integer $N \geq 1$ that satisfies Theorem 2.3. Let f be a block code that induces a conjugacy from X to Y, and let f^{-1} be the block code that induces its inverse. Since we can increase the window size, assume that the conjugacy has memory and anticipation k.

Since Y is an SFT, we know that there exists some M such that the desired property holds. We will show that $M = N + 4k$ satisfies the property for X. Consider blocks $uv, vw \in \mathcal{B}(X)$ such that $|v| \geq M$. We can take $s, t \in \mathcal{B}_{2k}(X)$ such that $suv, vwt \in \mathcal{B}(X)$. Observe that $f^{-1} \circ f(suvwt) = uvmw$.

Let $u', w' \in \mathcal{B}(Y)$ such that $f(suv) = u'f(v)$ and $f(vwt) = f(v)w'$ and $|f(v)| \geq M$. So, we know that $u'f(v)w' \in \mathcal{B}(Y)$. Thus, we get that $f^{-1}(u'f(v)w') \in \mathcal{B}(X)$, which allows us to conclude that $f^{-1}(f(suvwt)) = uvm \in \mathcal{B}(X)$, completing our proof. □

Theorem 2.7. *Every homomorphism between shift spaces is given by a sliding block code.*

Proof. Let $f : X \to Y$ be a homomorphism between shift spaces X and Y. Suppose f is a homomorphism. Let us use $\mathcal{A}$ and $\mathcal{B}$ to denote the alphabets of X and Y, respectively. Let us define the set of sequences of Y centered around a symbol b in $\mathcal{B}$.

$$C_0(b) = \{y \in Y \mid y_0 = b\}$$

We know that for all b, $C_0(b)$ are disjoint and compact in X. Thus, we can deduce that for all b, their inverse images $C_0^{-1}(b)$ are also disjoint and compact in X. So, there exists $\delta > 0$ such that $dist(x, y) \geq \delta$, where x and y belong to $C_0^{-1}(b_1)$ and $C_0^{-1}(b_2)$, respectively, for some $b_1 \neq b_2$. Let us choose n such that $2^{-n} < \delta$. Then, for any $x, x' \in X$ such that $x_{[-n,n]} = x'_{[-n,n]}$, we know that $x, x' \in C_0^{-1}(b)$, for some b. We can now define a $(2N+1)$-block map g such that $g(w) = f(x)_0$, for $x \in X$. We can easily check that $f = g_\infty^{[-N,N]}$. Therefore, we know that f is a sliding block code. □

We see that sliding block codes transform sequences by expanding and contracting them. Using a similar idea, we can transform shift spaces by changing the underlying alphabet. This is called passing the shift space to a *higher block presentation*. Like the name suggests, we do this by replacing the alphabet with $n - blocks$, or words of length n, of the original alphabet. Let X be a shift space over the alphabet $\mathcal{A}$. Let $\mathcal{A}_X^{[N]}$ denote the set of all allowed N-blocks of X. Naturally, we can construct a full shift from this new alphabet, which we will denote $(\mathcal{A}_X^{[N]})^{\mathbb{Z}}$. We can define a sliding block code $\beta_N \colon X \to (\mathcal{A}_X^{[N]})^{\mathbb{Z}}$ that replaces the ith coordinate of a sequence in X with a block of N coordinates centered at i, which is a symbol in the new, higher presentation alphabet. More formally,

$$\Big(\beta_N(x)\Big)_{[i]} = x_{[i,i+N-1]}$$

The N-block presentation of X is denoted $X^{[N]}$ and is the image of X over β_N. A 3-block presentation looks like the following.

$$\ldots [x_{-2}x_{-1}x_0][x_{-1}x_0x_1][x_0x_1x_2] \ldots$$

We can observe that in the N-block presentation of a shift space, the blocks “overlap”, or the original coordinates are repeated from one new symbol to the next. Thus, we can deduce both X and its image by taking just one coordinate from each new symbol.

Example 2.8. *Consider the 2-block presentation of the golden mean shift. We have that* $\mathcal{A}_X^{[2]} = \{a = 00, b = 01, c = 10\}$ *and* $\mathcal{F} = \{ac, ba, bb, cc\}$.

3. Matrix Presentations of Shifts

Since a shift space is entirely defined by its forbidden words, and likewise by its allowed words, we can represent them using structures like graphs and matrices. In this section, we will be assuming an understanding of basic definitions regarding graphs. A shift space can be easily translated into a directed graph using the symbols of its alphabet as vertices, where there is an edge from a vertex u to a vertex v if and only if uv is an allowed word.

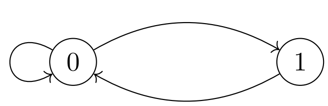

Figure 1. Graph representing the golden mean shift

The *vertex shift* of a graph is the space of biinfinite walks along the graph, recording the vertices traversed. We can think of the set of forbidden words of a vertex shift as the ordered pairs of vertices with zero entries in the corresponding adjacency matrix of the graph.

Theorem 3.1. *Every SFT is conjugate to a vertex shift.*

Proof. Let $X_{\mathcal{F}}$ be a shift of finite type with forbidden words $\mathcal{F}$. Without loss of generality, let $|w| = n + 1$, for all words $W \in \mathcal{F}$, since we can assume all words in the forbidden word list are of the same length. We can then pass $X_{\mathcal{F}}$ to a higher block presentation and assume $n = 1$. Let this new alphabet be the vertices of a graph G. Notice that the new set of forbidden words contains words of length 2, so G has an edge from i to j if and only if ij is not a forbidden word in the higher block presentation. The resulting vertex shift of G has the same alphabet and forbidden word list as $X_{\mathcal{F}}$. Thus, they are conjugate. $\square$

Now, let A be the adjacency matrix of a directed graph, allowing for multiple edges. Again, we can construct a shift space of biinfinite walks along the graph, this time recording the edges traversed. This is called the *edge shift* defined by A and is denoted S_A. Edge shifts are a much more compact and useful way of presenting shift spaces. For example, edge shifts often yield smaller matrices with integer entries that allow for matrix operations that cannot be used with the zero-one matrix resulting from a vertex shift presentation.

Example 3.2. *Consider the adjacency matrix of the graph of the golden mean shift.*

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

We can label the edges of Figure 1 to get the S_A. We have $\mathcal{A} = \{a, b, c\}$ and $\mathcal{F} = \{ac, ba, bb, gg\}$. Note that the resulting edge shift is based on the 2-shift presentation of the golden mean shift.

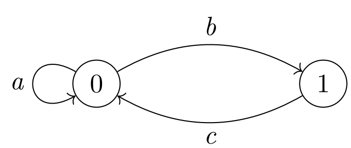

FIGURE 2. Edge shift of the recoded golden mean shift

We have that S_A is an SFT, with its forbidden words being the set of edge pairs without matching terminal and initial vertices. Furthermore, every SFT is conjugate to an edge shift, since the two-block presentation of an SFT is an edge shift.

Now, we can transform graphs representing shifts through state splittings and amalgamations. Essentially, in a state splitting, we divide the edges of a vertex between itself and additional new vertices. Consider a graph G with vertices V and edges E. Here, the terms vertex and state are synonymous. For each state $v \in V$, we will have partitions $E_v^1, E_v^2, \ldots, E_v^{m(v)}$ of the set of edges of v, for $m(v) \geq 1$. Let P be a partition of E. The *state split graph of G using P* has a vertex set comprised of $v^1, v^2, \ldots, v^{m(v)}$ for each vertex in V and an edge set with e^j for all edges in E, where $1 \leq j \leq m(t(e))$. For an edge $e \in E$ that goes from u to v, we have that $e \in E_u^i$, for some i. The initial state of e^j in the state split graph is defined by $i(e^j) = u^i$. Likewise, we have that the terminal state of e^j is $t(e^j) = v^j$. If a graph G is a *splitting* of a graph H, then H is an *amalgamation* of G. Depending on whether the outgoing or incoming edges are "split", we also have *out-splittings* and *in-splittings* of graphs and likewise *out-amalgamations* and *in-amalgamations*. State splittings preserve many basic characteristics of a graph. It is a fact that state splittings produce conjugate edge shifts. Furthermore, state splittings are notable for their use in transforming shifts, which we will see next.

Theorem 3.3 (Depomposition Theorem)**.** *Every conjugacy between edge shifts is a composition of state splitting codes and amalgamation codes.*

The full proof of the Decomposition Theorem is too involved for the purposes of this paper; however, the outline of the proof is as follows. Given a conjugacy $f\colon S_A \to S_B$, we can recode f to be a 1-block code from a higher block presentation of S_A to S_B. Here, we focus our attention to A and B being essential graphs, or graphs with no isolated points, as it is a fact that any graph contains a unique essential subgraph that produces an equal edge shift. Correspondingly, essential matrices are matrices without columns or rows of all zeroes. If f^{-1} is a 1-block code, we are done. However, if it is not, we can whittle down the memory and anticipation of f^{-1} using a helpful lemma. In a later section, we will use this theorem to prove an important result regarding topological conjugacies. In order to accomplish this, however, we must explore the matrix implications of this statement. Given adjacency matrices A, B, where the graph of B is a splitting of the graph represented by A, there exist matrices D, E with nonnegative integer entries such that $A = DE$ and $B = ED$.

4. Algebraic Invariants

A fundamental problem in the field of symbolic dynamics is the question of when two systems are topologically conjugate. The question has naturally been studied extensively in the world of shift spaces, though an effective classification of SFT's has not yet been achieved. It is useful to look into the various invariants of topological conjugacy, as it yields many interesting results. However, in this paper, we will narrow our focus to Bowen Franks Groups, as they appear in the discussion of flow equivalence.

As we've seen before, a shift of finite type can be represented by an edge shift S_A with an adjacency matrix A. The *Bowen-Franks group* of an $r \times r$ matrix A is defined as the following, where $\mathbb{Z}^r(I - A)$ is the image of $\mathbb{Z}^r$ under the matrix $I - A$ from the right.

$$BF(A) = \mathbb{Z}^r / \mathbb{Z}^r(I - A)$$

It is known that if two shifts are conjugate, they share the same Bowen-Franks group. As asserted earlier, Bowen-Franks groups are invariant under flow equivalence, which we will define next.

Definition 4.1. The shifts S_A and S_B are *flow equivalent* if and only if the following equations are satisfied for irreducible, non-permutation matrices A and B.

$$\det(I - A) = \det(I - B)$$

$$\operatorname{cok}(I - A) \cong \operatorname{cok}(I - B)$$

Theorem 4.2. *If two shift spaces are flow equivalent, they have the same Bowen-Franks Group.*

The proof of this statement is lengthy; however, we can easily show that Bowen Franks groups are invariant under shift equivalence. The difficulty lies in showing that shift equivalence implies flow equivalence. From these two statements, the desired result follows.

Though flow equivalence is a relatively weak relation, a general classification of SFT's up to flow equivalence has been achieved by the work of D. Huang in 1994. We will discuss stronger classifications of SFT's next.

5. Shift Equivalence and Strong Shift Equivalence

The notion of shift equivalence and strong shift equivalence was developed by R.F. Williams in his 1973 paper on the classification of shifts. Williams achieved many important results and laid the foundations for much of what is studied in the field today. The question, first proposed by Williams, of whether shift equivalence and strong shift equivalence were in fact the same remained an open problem for many years. However, in 1997, it was proven false by the joint work of Kim and Roush. In this section, we will discuss the work of Williams and his predecessors in hopes of illuminating the current state of affairs in the study of shift classification.

Definition 5.1. Matrices A and B are *elementary strong shift equivalent* if there exist matrices U, V such that $A = UV$ and $B = VU$. Furthermore, A and B are *strong shift equivalent* if there is a finite sequence of elementary shift equivalences going from one to the other. That is $A = A_0 A_1 \cdots A_n = B$, where A_i and A_{i+1} are elementary strong shift equivalent for $0 \leq i < n$.

Unlike elementary strong shift equivalence, strong shift equivalence is indeed an equivalence relation.

Proposition 5.2. *Strong shift equivalence is an equivalence relation on matrices with entries in $\mathbb{Z}_+$.*

Proof. We know that for any sequence of elementary strong shift equivalences from A to B, the reverse sequence can be used to go from B to A. So, we have that strong shift equivalence is symmetric. Furthermore, SSE is transitive since we can chain sequences together. Since we immediately have reflexivity, we can conclude that SSE is an equivalence relation. □

Theorem 5.3. *Edge shifts S_A and S_B are topologically conjugate if and only if A and B are strong shift equivalent.*

Proof. $\Longrightarrow$ By the Decomposition Theorem, we get that for essential matrices A and B, if S_A and S_B are topologically conjugate, then there is a sequence of matrix pairs (U_i, V_i) with nonnegative integer entries that has the following property.

$$\begin{aligned} A &= V_0U_0 & U_0V_0 &= A_1 \\ A_1 &= V_1U_1 & U_1V_1 &= A_2 \\ & \vdots & & \\ A_\ell &= V_\ell U_\ell & U_\ell V_\ell &= B \end{aligned}$$

That is, A and B are strong shift equivalent. It is easy to show that any inessential matrix is strong shift equivalent to an essential one. Thus, we have our desired result.

$\Longleftarrow$ Let G_A and G_B be directed graphs with adjacency matrices A and B, respectively. Let U be an adjacency matrix of edges with initial vertices in G_A and terminal vertices in G_B. Likewise, we define V as an adjacency matrix for edges from G_B to G_A. Let $U(i,k)$ denote the number of edges from a vertex i in G_A to a vertex k in G_B. Similarly, let $V(k,j)$ be the number of edges from k to a vertex j in G_A. We have the following equation for vertices i,j in G_A.

$$A(i,j) = (UV)(i,j) = \sum_k U(i,k)V(k,j)$$

We can write an analogous equation for vertices in G_B. Now, we choose bijections α, β between arcs and paths of length 2. For a point in S_A, we can apply α to each edge a_i, regroup, then apply β^{-1} to get a point in S_B.

$$\begin{aligned} \ldots a_0a_1a_2\cdots &\longleftrightarrow \ldots [u_0v_0]\,[u_1v_1]\,[u_2v_2]\ldots \\ &\longleftrightarrow \quad \ldots [v_0u_1]\,[v_1u_2]\,[v_2u_3]\cdots \longleftrightarrow \ldots b_0b_1b_2\ldots \end{aligned}$$

Since this bijection is given by a block code, we have an isomorphism. $\square$

Although strong shift equivalence can be used to classify SFT's up to topological conjugacy, it is difficult to understand and largely impractical. To this effect, Williams defined the notion of shift equivalence, a far more accessible relation on shifts.

Definition 5.4. Matrices A and B are *shift equivalent* if there exist U,V and $\ell \in \mathbb{Z}_+$ satisfying the following four equations. Note that we say that A and B are shift equivalent with *lag* ℓ.

$$\begin{aligned} A^\ell &= UV & UB &= AU \\ B^\ell &= VU & BV &= VA \end{aligned}$$

A and B are shift equivalent if and only if S_A and S_B are *eventually isomorphic*, or if $(S_A)^n$ and $(S_B)^n$ are isomorphic for all but finitely many n. We will show that as the names imply, strong shift equivalence implies shift equivalence. However, we must obtain a preliminary result.

Proposition 5.5. *If A and B are shift equivalent with a lag ℓ and B and C are shift equivalent with a lag ℓ', then A and C are shift equivalent with lag $\ell + \ell'$*

Proof. Let matrices U,V satisfy the shift equivalence equations for A and B. Similarly, let U',V' satisfy the equations for B and C. Then, we get that $A^\ell = UV$, $B^\ell = VU$ and $B^{\ell'} = U'V'$ and $C^{\ell'} = V'U'$. From here, we can easily check that the matrices UU' and $V'V$ satisfy the equations for A and C with a lag of $\ell + \ell'$. $\square$

Theorem 5.6. *If matrices A and B are strong shift equivalent, then they are shift equivalent.*

Proof. Suppose A and B are strong shift equivalent. Then, there exists a chain of elementary strong shift equivalences between them. Since elementary equivalences are shift equivalences with $\ell = 1$, we can apply Proposition 4.2 ℓ times to get that A and B are shift equivalent. □

The question of whether shift equivalence implies strong shift equivalence was unsolved for many years before proven false in 1997 by K.H. Kim and F. Roush. Kim and Roush were able to provide counterexamples for matrices with nonnegative integer entries; however, the problem remains open for other classes of matrices.

Acknowledgments

First and foremost, I would like to thank my mentor, Polina Baron, for guiding me through the daunting process of writing a mathematical paper. I also extend my gratitude to Peter May for organizing this program and for being a welcoming and supportive figure in the math department. Finally, I would like to thank Daniil Rudenko and my fellow REU participants for making this summer incredibly enriching and memorable.

References

[1] M Boyle. Algebraic Aspects of Symbolic Dynamics. In F. Blanchard, A. Maass, & A. Nogueira, eds. Topics in Symbolic Dynamics and Applications. Cambridge University Press. 2000. 57–88.
[2] K. H. Kim, F. W. Roush. The Williams Conjecture is false for irreducible subshifts. https://doi.org/10.2307/120975. 1997. 105-109.
[3] D. Lind, B. Marcus. An Introduction to Symbolic Dynamics and Coding 2nd ed. Cambridge University Press. 2021.
[4] S. G. Williams. Introduction to Symbolic Dynamics. Proceedings of Symposia in Applied Mathematics. https://www.southalabama.edu/mathstat/personal_pages/williams/wilshort.pdf
[5] R. F. Williams. Classification of subshifts of finite type. Annals of Math. 1973.