Symbolic dynamics

Recall: For a finite alphabet A and monoid M, the **full shift** (A^M, σ) over M is defined by

$$\forall x \in A^M \ : \ \forall m \in M \ : \ \sigma^m(x)_k = x_{mk}.$$

In the cases of additive monoids $M = \mathbb{N}$ and $M = \mathbb{Z}$ we usually denote σ^1 by σ and call it the **left-shift**.

In symbolic dynamics we are interested in subsystems of (A^M, σ) (called **subshifts**) and their endomorphisms (called **cellular automata**).

Recall: A subshift is a **non-empty** topologically **closed** subset $S \subseteq A^M$ that is **invariant**: $\sigma^m(S) \subseteq S$ for all $m \in M$.

For a subshift S over a group we have that $\sigma^m(S) = S$ for all $m \in M$:

Subshifts

For $S \subseteq A^M$ define

$$L_S = \{w \in A^D \mid D \subseteq M \text{ finite}, [w] \cap S \neq \emptyset\},$$

and call L_S the **language** of S.

The language of a subshift thus consists of all finite patterns that appear in some element of S. It uniquely identifies the subshift:

Proposition. For subshifts: $S \neq T \Longrightarrow L_S \neq L_T$.

Proof.

We may describe subshifts using **forbidden patterns**.

Let P be a set of finite patterns. The set of configurations that do not contain any of patterns in P is

$$X_P = \{x \in A^M \mid \forall m \in M, \forall w \in P \ : \ \sigma^m(x) \notin [w] \ \}.$$

Proposition. Set $S \subseteq A^M$ is a subshift (or the empty set) if and only if $S = X_P$ for a set P of finite patterns.

Proof.

From the proof:
Corollary. $S = X_P$ where P is the complement of L_S.

Describing a subshift by its **language** vs. **forbidden patterns**:

- (a) Some sets of patterns are not a language of any subshift,
  (b) every set of patterns defines a subshift (or $\emptyset$) as forbidden patterns.

- (a) The language of a subshift bijectively identifies it,
  (b) different sets of forbidden patterns may define the same subshift.

We classify subshifts based on how simple sets of forbidden patterns can describe it.

Subshift of finite type (SFT)

Subshift S is called a **subshift of finite type (SFT)** if $S = X_P$ for a **finite set** P.

Note: If a subshift is defined by an infinite set of forbidden patterns, this does not imply that the subshift would not be of finite type. (There may be another finite set of forbidden patterns that defines the same subshift.)

Example (The Golden mean shift). Let $M = \mathbb{Z}$ and $A = \{0, 1\}$. The golden mean shift S_{gm} is defined by forbidding word 11. All 1's must be isolated in the configurations. This is an SFT.

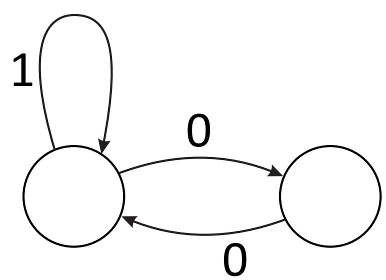
Example. Let $M = \mathbb{Z}$ and $A = \{0, 1\}$. Forbid all patterns that contain both symbols 0 and 1. This only allows two configurations ${}^{\infty}0^{\infty}$ and ${}^{\infty}1^{\infty}$. Although we forbid infinitely many patterns this subshift is of finite type: the same subshift is defined by forbidding two patterns 01 and 10.

Example (The even shift).

Let $M = \mathbb{Z}$ and $A = \{0, 1\}$. The even shift S_{even} is defined by forbidding words $10^{2n+1}1$ for all $n \in \mathbb{N}$. There must be an even number of 0's between consecutive 1's.

The even shift is not an SFT:

The even shift is an example of a **sofic shift**. The configurations are obtained by reading the labels along bi-infinite paths in the labeled directed graph:



In terms of formal languages, the set of subwords that appear in configurations of the even shift are a regular language (recognized by a finite automaton).

Homomorphisms

Let $S \subseteq A^M$ be a subshift and B an alphabet. Let

- $D \subseteq M$ be finite (a **neighborhood**),
- $\phi : (L_S \cap A^D) \longrightarrow B$ be a function (a **local rule**) that assigns symbols of B to D-patterns of S.

These determine a **block map** $f : S \longrightarrow B^M$ by

$$\forall x \in S \ : \ \forall m \in M \ : \qquad f(x)_m = \phi(\sigma^m(x)_{|D}).$$

(Function f applies ϕ on the D-pattern around each cell; similar to cellular automata on full shifts defined earlier.)

A block map f is a homomorphism $(S, \sigma) \longrightarrow (B^M, \sigma)$:

- **Continuous**:

- **Commutes with the shift**:

Also the converse holds: Block maps are precisely the homomorphisms between subshifts.

Curtis-Hedlund-Lyndon -theorem. Let $S \subseteq A^M$ be a subshift. A function $f : S \longrightarrow B^M$ is a homomorphism from (S, σ) to (B^M, σ) if and only if it is a block map.

Proof.

Corollary. The endomorphisms of a full shift are precisely its cellular automata, and its automorphisms are precisely the reversible cellular automata. This holds over any monoid M.

More generally, the endomorphisms of a subshift S are called **cellular automata on subshift S**. They are all block maps.

Example. Let S_{gm} be the golden mean subshift (word 11 forbidden). Define a block map $f : S_{gm} \longrightarrow \{0,1\}^{\mathbb{Z}}$ by

- neighborhood $D = \{0,1\} \subseteq \mathbb{Z}$,
- local rule $\phi : (L_{S_{gm}} \cap \{0,1\}^D) \longrightarrow \{0,1\}$ where

$$\begin{aligned} 00 &\mapsto 1, \\ 01 &\mapsto 0, \\ 10 &\mapsto 0. \end{aligned}$$

Then $f(S_{gm})$ **is the even shift** S_{even}:

A pre-image of $10^k 1$ must have form $00(10)^n 0$, and then $k = 2n$. So $f(S_{gm}) \subseteq S_{even}$.

Every $x \in S_{even}$ has a pre-image.

Thus f is a factor map from (S_{gm}, σ) to (S_{even}, σ).

Lemma.

(a) Let $f : A^M \longrightarrow A^M$ be an endomorphism of a full shift. Its fixed point set $\{x \in A^M \mid f(x) = x\}$ is an SFT (or empty).

(b) Let $f : A^M \longrightarrow B^M$ be a homomorphism between full shifts, and let $S \subseteq B^M$ be an SFT. Then $f^{-1}(S)$ is an SFT (or empty).

(c) A finite intersection of SFTs is an SFT (or empty).

Proof.

Proposition. Let $S \subseteq A^M$ and $T \subseteq B^M$ be conjugate subshifts. If S is of finite type so is T.

Proof.

Sofic subshifts

Factors of SFTs are called **sofic** subshifts. An SFT S_0 is called a **SFT cover** of the sofic subshift $S = h(S_0)$.

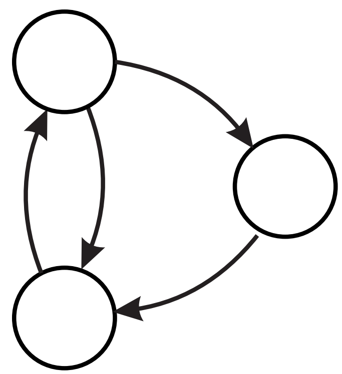
Example. The even shift is a factor of the golden mean shift, and the golden mean shift is an SFT. Thus the even shift is sofic.

Proposition. A factor of a sofic shift is sofic. In particular, a subshift that is conjugate to a sofic shift is sofic.

Proof.

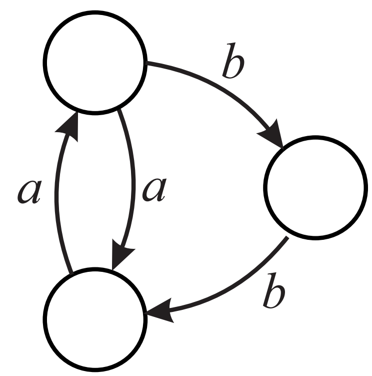
Edge shifts

Each (directed) graph G defines a subshift E_G, the **edge shift** of G, as follows: the set E_G consists of all two-way infinite paths of edges in G. The alphabet is thus the set E of the edges of the graph.



The edge shift is of **finite type**, and every subshift $S \subseteq A^{\mathbb{Z}}$ of finite type is conjugate to an edge shift of some graph (homework).

If we **label** the edges by letters of some alphabet A, we can also read the labels along all two-way infinite paths in the graph. The set of such sequences is a **sofic subshift**, because the labeling function gives a factor map from the edge shift E_G.



Moreover, every sofic subshift $S \subseteq A^{\mathbb{Z}}$ can be obtained in this manner.

Effective subshifts

Let $M = \langle G \rangle$ be **finitely generated** (G finite), with **decidable word problem** (there is an algorithm that, when given two products of generators, decides whether the products are the same element of M.)

• Elements of M are represented as products of generators, and there is thus an algorithm that tells if two given elements are the same.

• A finite pattern $p \in A^D$ is represented as a finite set

$$\{(m, a) \mid m \in D, a = p(m)\}$$

of pairs (m, a) where $m \in M$ (given as a product of generators) and $a \in A$. The pattern assigns symbol a in cell m.

Decidable word problem guarantees that:

- there is an algorithm to check that a given pattern is **consistent** (does not contain multiple pairs (m, a) and (m, b) with $a \neq b$).
- there is an algorithm to test if two given patterns are identical.

A set P of finite patterns is called

- **recursive** if there is an algorithm to decide if a given pattern w is in P,
- **recursively enumerable** if there is an enumeration algorithm that lists elements of P one-by-one.
  - Equivalently, $P = \{p_1, p_2, \dots\}$ in some order and there is an algorithm that produces for any given i the i'th pattern p_i.
  - Yet equivalently, there is a semi-algorithm to recognize elements of P.

Every recursive set is also recursively enumerable but the converse is not true.

However, subshifts X_P defined by forbidding recursive sets P turn out to be the same as subshifts defined by forbidding recursively enumerable sets P.

Proposition. Let M be a finitely generated monoid with decidable word problem, and let $S \subseteq A^M$ be a subshift. The following are equivalent:

(a) $S = X_P$ for a recursively enumerable set P of finite patterns.

(b) $S = X_P$ for a recursive set P of finite patterns.

(c) The complement of L_S is recursively enumerable.

A subshift satisfying any of the equivalent conditions (a)-(c) is called **effective**.

Examples. Let M be a finitely generated monoid with decidable word problem.

An **SFT** $S \subseteq A^M$ is trivially effective since a finite set P of patterns is recursive.

Let $S \subseteq B^M$ be a **factor of an effective subshift** $S_0 \subseteq A^M$, say $S = h(S_0)$ for a block map h with the neighborhood $N \subseteq M$ and local rule $\phi : A^N \longrightarrow B$.

Let us prove that S **is effective**. A pattern $p \in B^D$ is in the language L_S if and only if it has a "pre-image pattern" $p_0 \in A^{DN}$ that is in the language L_{S_0}. So p is in the complement of L_S if and only if all its pre-image patterns $p_0 \in A^{DN}$ are in the complement of L_{S_0}. We can recursively enumerate the complement of L_{S_0}, and if eventually all pre-images of p have been enumerated we can announce that p is in the complement of L_S.
$\Longrightarrow$ the complement of L_S is recursively enumerable.

In particular, all **sofic** subshifts $S \subseteq A^M$ are effective.