03/15/23 - Symbolic Dynamics - Prof. Wolf

Recall:
(X, d_X) a complete metric space with a countable dense set $S_X = \{s_i\} \subseteq X$ (makes X separable).
We say (X, d_X, S_X) is a computable metric space if $\exists$ turing machine $\mathcal{X}: \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathbb{Q}$ s.t. $|\mathcal{X}(i,j,n) - d_X(s_i, s_j)| < 2^{-n}$
(note: this is a uniform turing machine b/c not dependent on i, j)

Computability doesn't say anything about complexity!

fun e.g. fn that is 1 on $\mathbb{Q}$, 0 on $\mathbb{R} \setminus \mathbb{Q}$. Is not computable b/c it's not continuous and comp. fns are cont., but restricted to $\mathbb{Q}$ it is!

Let $B \subseteq X$, $g: B \to \mathbb{R}$ computable if $\exists$ turing machine $\mathcal{X}$ s.t. $\forall x \in B$ and $\forall$ oracles φ of x (recall, oracle outputs $s_i \in S_X$ w/in 2^{-n} of point): $|\mathcal{X}(\varphi, n) - g(x)| < 2^{-n}$

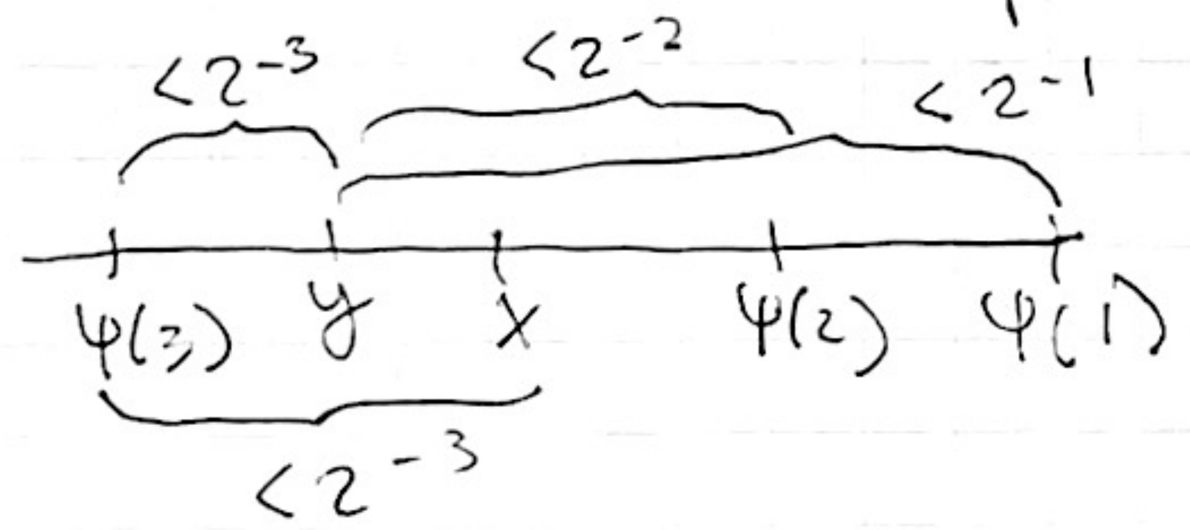
model this oracle as an input as a tape w/ increasing precision, $\mathcal{X}$ gets to choose how many times to query.

We say g is computable at x_0 in B if $\exists$ turing machine $\mathcal{X}$ which takes any oracle φ of x_0 and $n \in \mathbb{N}$ and outputs a rational number s.t.
let $l_{\varphi,n}$ be the precision that $\mathcal{X}$ queries the oracle φ to compute $\mathcal{X}(\varphi, n)$ then $\forall y \in B$ s.t. $\exists$ oracle φ' of y which coincides with φ up to $l_{\varphi,n}$ then:
- $\mathcal{X}(\varphi, n) = \mathcal{X}(\varphi', n)$
- $|\mathcal{X}(\varphi, n) - g(y)| < 2^{-n}$

Computer can compute $g(x_0)$ and the continuity estimate can also be computed

$\hookrightarrow$ computable at a point doesn't tell us about computability of the fn, b/c could have infinately many turing machines at diff. points.

(turing machine computes $g(x)$, but it doesn't know x only the oracle)



* this picture is important.

Computability of Subshifts

def: let $X \subseteq \Sigma_d^{\pm}$. We say φ is an oracle of X if $\varphi(n) = \mathcal{L}_{2n+1}(X)$ (symmetric cylinders)
(a sequence that gives us the language in increasing length)

def: If $x \in X$, then an oracle of x is a fn $\varphi(n) = x_{[-n,n]}$.

def: We say x (resp. X) is computable if $\exists$ turing machine $\mathcal{X}$ which is an oracle of x (resp. X).

fact: full shift $\Sigma_n^{\pm}$ is computable
intuition: given finite alphabet length n we can easily write a program that lists all words in increasing order.

def. We say X is upper semi-computable if $\exists$ Turing machine $\mathcal{X} = \mathcal{X}(n,k)$ s.t. $(\mathcal{X}(n,k))_k$ is a nonincreasing finite collection of words of length $2n+1$ s.t. $\mathcal{X}(n,k) = \mathcal{L}_{2n+1}(X)$ for k large enough.

(always an overset of $\mathcal{L}(X)$ that gets smaller)
(it's not clear how long you have to wait at a given (n,k).)

Shift space computable if upper and lower semi-computable.

We recall the metric $d = d_{\frac{1}{2}}$ on $\Sigma_d^{\pm}$ $\quad d(x,y) = \left(\frac{1}{2}\right)^{\inf\{|k| : x_k \neq y_k\}}$

idea: if I have oracles for x and y I can compute the distance between them to any precision.

Fix the total lexicographic order on $\Sigma_d^{\pm}$ by $x < y$ iff. $x_0 < y_0$ or $\exists\, k \in \mathbb{N}$ s.t. $x[-k,k] = y[-k,k]$ and either $x_{k+1} < y_{k+1}$ or $x_{k+1} = y_{k+1}$ and $x_{-k-1} < y_{-k-1}$

x

use this $[-k$ " " " " " " $k]$

y

if order, use this, if id.

idea: a computer can order them but not decide if they're equal.

Lemma: Let $X \in \Sigma_{\inf}$ a subshift. Assume X is given by an oracle φ. Then X is a computable metric space relative to φ.

proof: (idea: give S_X.) The distance between points x and y can be computed at any precision. For $\tau \in \varphi(n) = \mathcal{L}_{2n+1}(X)$ and $i \in \mathbb{Z}$ we define $S_{\tau,i}$, the smallest point in $[\tau]_i$ (according to lexicographic order) (cyl w/ τ centered at i.)

Then $\{S_{\tau,i}\}$ is a countable dense ~~set~~. b/c for each i there are finite τ, and each τ there are countable i. ∎

Recall: we know how to compute entropy of SFTs (b/c we can write as 1-step w/ trans. matrix) and sofic shifts (b/c factors of SFTs)
↳ natural idea is to approximate other entropies using these, first we want to know if that's even computable.

<u>Computability of SFTs & Sofic Shifts</u>

Lemma: Suppose E is a finite set of words. Then $\exists$ Turing machine $\mathcal{X}$ which takes E compact and produces a finite alphabet $\mathcal{A}'$ and a transition matrix A s.t. $X_{\mathcal{F}}$ is top. conj. to $X'_{A'}$.

let $P_{\mathcal{F}}(d)$ the set of all finite sets of words over alphabet $\mathcal{A}$.

cor: the map $P_F(d) \longrightarrow \mathbb{R}_0^+$ $\mathcal{F} \longmapsto h_{top}(X_{\mathcal{F}})$ is computable
(SFTs have computable entropy)

proof: conj. subshift which is a 1-step SFT, then compute spectral radius. And this is done with just one Turing machine for every SFT!
Same is true for Sofic shifts.

03/15/23 — Symbolic Dynamics — Prof. Wolf

& TODO: review

idea: compute extension of a sofic shift to an SFT, then use the same process b/c it has the same entropy.

prop: let $X \in \Sigma_{inf}$ subshift given by an oracle. Then $h_{top}(X)$ is upper semi-computable.
(i.e. $X \longmapsto h_{top}(X)$ computable)

proof: Given φ we can compute $\frac{1}{n}\log|\mathcal{L}_n(X)|$ at any precision.
Next we can select a rational number r_n s.t.
$\frac{1}{n}\log|\mathcal{L}_n(X)| < r_n \leq \frac{1}{n}\log|\mathcal{L}_n(X)| + \frac{1}{2^n}$

Define $q_n = \min\{r_1, \dots, r_n\}$. Then (q_n) is a nondecreasing computable sequence of rational numbers which converges to $h_{top}(X)$ (from above) (b/c $h_{top}(X) = \inf_n \frac{1}{n}\log|\mathcal{L}_n(X)|$)

Which objects are intrinsically accessible by a computer?

all sub shifts, distance between shifts

$(\Sigma_{inf}, d_{inf}, P_F(d))$, $\cong$ SFTs — makes space of all shift spaces a computable metric space
- complete
- separable b/c SFTs are dense, countable
- d_{inf} makes it a metric space

thm: the function $\Sigma_{inf} \ni X \to h_{top}(X)$ is computable at X iff. $h_{top}(X) = 0$.

proof: (with ~~some~~ technical details missing)
We will use the notion of computability at a point.

Case 1: $h_{top}(X) > 0$. Want to find for every n a shift space
fix $n \in \mathbb{N}$. $\exists$ SFT X_F' s.t. $d_{inf}(X, X_F') < 2^{-n}$ (i.e. $\mathcal{L}_n(X) = \mathcal{L}_n(X_F')$)

let $X_n^1 \dots X_n^\ell$ be the transitive components of X_F'
Since transitive SFTs satisfy that periodic points dense
$\exists$ periodic orbits $P_n^1, \dots, P_n^\ell$ s.t. $\mathcal{L}_{2n+1}(P_n^i) = \mathcal{L}_{2n+1}(X_n^i)$
(I can find a periodic orbit that has the same language as the approximating SFT up to a point.)

Well $Y = \bigcup_{i=1}^{\ell}\{P_n^i\}$ is an SFT $\Rightarrow h_{top}(Y) = 0$.

this is a point that breaks the def. of computability at a point.

issue is in one sided transitions between transient subshifts

Case 2: $h_{top}(X) = 0$
Define $\forall m \in \mathbb{N}$, $h(m) = \min_{\ell=1 \dots m} \frac{1}{\ell}\log|\mathcal{L}_\ell(X)|$
we know this converges from above to entropy, which is 0.
fix $n \in \mathbb{N}$. $\exists m \in \mathbb{N}$ s.t. $h(m) < \frac{1}{2^n}$, which can be computed.
Suppose Y is a subshift whose language up to len ~~n~~ m coincides w/ that of X.

Then $0 \le h_{top}(Y) \le h(m) < \frac{1}{2^n}$ $\boxed{}$

(this comes from the fact that we have this info def.)

This is computing a whole neighborhood, not just that 0 is computable.