

TODO: finish proof of completeness Add proof that all computable functions continuous and related notes Answer some questions (understand type signature of turing machines) Add context on understanding turing machines and oracles

To talk about computability in shift spaces, we need to first set up the space of all shift spaces and verify that it has some reasonable properties that will make our lives easier in the longrun. We want to show that the set of all shift spaces Σ_{inf} is a metric space that is complete and compact. Once we have established these properties we will introduce the notion of computability in generally, and then computability in metric spaces to show that there is a notion of Σ_{inf} being computable. Later we will be able to then discuss the computability of certain properties of shift spaces, like entropy. This question is deeply important to our use of computers in understanding processes in the physical world.

1. SHIFT SPACES AS A METRIC SPACE

Lemma 1.1. *Let $X, Y \in \Sigma_{inf}$ be shift spaces. The following is a metric,*

$$d_S(X, Y) = \begin{cases} 0 & X = Y \\ \frac{1}{2}^l & l = \inf\{w_l \in \mathcal{L}(X) \cup \mathcal{L}(Y) \setminus \mathcal{L}(X) \cap \mathcal{L}(Y)\} \end{cases}$$

That is, l is smallest word length that causes $\mathcal{L}_l(X) \neq \mathcal{L}_l(Y)$, and the smaller this number the further apart the spaces are.

Proof. To show something is a metric we must show for all X, Y, Z that

- (1) $d_S(X, X) = 0$,
- (2) $d_S(X, Y) = d_S(Y, X)$, and
- (3) $d_S(X, Y) \leq d_S(X, Z) + d_S(Z, Y)$, the triangle inequality.

The first two are required by the definition of d_S . The triangle inequality is shown by computation. Let $X, Y \in \Sigma_{inf}$. If $X = Y$ the proof is trivial, so suppose $X \neq Y$. Then we have $d_S(X, Y) = 2^{-l}$ for some l and minimal word w_l that the languages of X and Y disagree on. Suppose $w_l \in \mathcal{L}(X)$ and not in $\mathcal{L}(Y)$, the proof of the other direction is the same. Since l is minimal, $w_i \in \mathcal{L}(X) \cap \mathcal{L}(Y)$ for all $i < l$.

Now let Z be some other shift space. Either $d_S(X, Z) \geq d_S(X, Y)$, in which case the triangle inequality is proven, or $d_S(X, Z) < d_S(X, Y)$, in which case $d_S(X, Z) = 2^{-i} < 2^{-l}$ with $i > l$. So all smaller words w_j for $j < i$ are in $\mathcal{L}(X) \cap \mathcal{L}(Z)$. That means the original word the languages of X and Y disagreed on w_l is in $\mathcal{L}(Z)$. So $d_S(Y, Z)$ is at least 2^{-l} , which is good enough to show the triangle inequality. \square

Proposition 1.2. *The metric space (Σ_{inf}, d_S) is complete.*

Proof. We can show a metric space is complete by showing that all Cauchy sequences converge. Recall a sequence $(X_j) \in \Sigma_{inf}$ is Cauchy if given an $\epsilon > 0$ there exists an N such that for all $m, n \geq N$, $d_S(X_m, X_n) < \epsilon$. We would like to be able to think about this in terms of languages though, so we present an equivalent statement for a sequence of shift spaces to be Cauchy.

Let (X_i) be a Cauchy sequence in Σ_{inf} . Given any $\epsilon > 0$ we can find an l such that $2^{-l} \leq \epsilon$. If $d_S(X_n, X_m) \leq 2^{-l}$, that means the smallest word for which their languages disagree has length $k \geq l$. Now consider the index N required to exist by the Cauchy property, and note that this means $d_S(X_N, X_m) \leq 2^{-l}$ so $\mathcal{L}_l(X_N) = \mathcal{L}_l(X_m)$ for all $m \geq N$.

// TODO

For all $n \in \mathbb{N}$ there exists an index $J_n \in \mathbb{N}$ such that for all $j \geq J_n$ $\mathcal{L}_n(X_j) = \mathcal{L}_n(X_{J_n})$. That is, for every word length there is an index into the sequence after which the set of allowable words of that length stabilizes and remains the same for the rest of the sequence.

We will use these stabilized allowable word sets to build a language, and then show that the shift space generated by that language is the limit of the sequence. Let $L_n = \mathcal{L}_n(X^{J_n})$ and $L = \cup_n L_n$. To show L is a language we first need that any word w can be extended.

// TODO

Next we need to show that L is closed under subwords.

// TODO

Conclusion: since L is a language, X_L is a shift space. And $\lim_{n \rightarrow \infty} (X_n) = X_L$. And since we have shown that an arbitrary Cauchy sequence (X_n) converges, we have shown that (Σ_{inf}, d_S) is a complete metric space. \square

Proposition 1.3. *The metric space (Σ_{inf}, d_S) is compact.*

Proof. The following proof relies on the fact that in a metric space compactness is equivalent to sequential compactness. We will take an arbitrary sequence and produce a convergent subsequence, from which compactness follows.

Let (X_j) be a sequence in Σ_{inf} . We will construct a convergent subsequence by induction.

(n=1) Pick $j_1 \in \mathbb{N}$ such that $\mathcal{L}_1(X_{j_1}) = \mathcal{L}_1(X_j)$ in infinitely many $j \geq j_1$.

(n+1) Suppose we have $j_1 \dots j_n$ as above, and also with $\mathcal{L}_i(X_{j_i}) = \mathcal{L}_i(X_{j_k})$ for all $j_i < j_k \leq j_n$. (That is, after j_i the \mathcal{L}_i agree for all following indices up to j_n , and then there are infinitely many indices after that whose language at that length also agree.)

Then there exists some index $j_{n+1} > j_n$ such that $\mathcal{L}_1(X_{j_1}) = \mathcal{L}_1(X_{j_{n+1}}), \dots, \mathcal{L}_n(X_{j_n}) = \mathcal{L}_n(X_{j_{n+1}})$ (at the new index the language agrees with the language of words of length i at X_{j_i}) and $\mathcal{L}_{n+1}(X_{j_{n+1}}) = \mathcal{L}_{n+1}(X_j)$ for infinitely many $j \geq j_{n+1}$. QUESTION: how do we know we can find such an index?

By a similar argument as the last proof, this subsequence is Cauchy, and so it converges by the last theorem. \square

2. COMPUTABILITY

For now we will consider computability within \mathbb{R} with the usual metric $||$. In the next section we will generalize all of these definitions.

Definition 2.1. Let $x \in \mathbb{R}$. An *oracle* of x is a function $\varphi : \mathbb{N} \rightarrow \mathbb{Q}$ such that $|\varphi(n) - x| < 2^{-n}$.

The input in \mathbb{N} should be thought of as the desired degree of precision, and the fact that the output value is in \mathbb{Q} is significant because ... QUESTION: why?

Definition 2.2. We say $x \in \mathbb{R}$ is computable if there exists a turing machine χ which is an oracle of x .

Turing machines will not be formally defined here. The vibes definition is that a turing machine is a computer program. // QUESTION: more about how to think about turing machines and oracles and how they relate to these $< 2^{-n}$ bounds?

Definition 2.3. We say $x \in \mathbb{R}$ is *upper semi-computable* if there exists a turing machine $\chi : \mathbb{N} \rightarrow \mathbb{Q}$ such that $(\chi(n))$ is a nonincreasing sequence with $\lim_{n \rightarrow \infty} \chi(n) = x$. Similarly χ is *lower semi-computable* if $-x$ is upper semi-computable.

We state as a fact that a real number is computable if and only if it is upper and lower semi-computable. The proof idea is to let the upper and lower turing machines run until they are within 2^{-n} of each other.

Examples of familiar computable numbers are rational, algebraic and some transcendental numbers. Both π and e are computable. But most numbers are not computable! It is hard to produce an example of one because if we could write it down then we would have an algorithm for computing it. To produce an example, we make use of the halting problem, which states that there is no universal turing machine that decides if any given Turing machine halts. There are a countable number of turing machines, so let T_i be an ordering of all turing machines. Then define $t_n = \begin{cases} 0 & T_n \text{ halts} \\ 1 & T_n \text{ doesn't halt} \end{cases}$. Let $\alpha = 0.t_1t_2t_3\dots$. This number α must be uncomputable otherwise it would be a universal turing machine that solves the halting problem.

Now we consider the definition of a computable function.

Definition 2.4. Let $S \subseteq \mathbb{R}^n$. We say $g : S \rightarrow \mathbb{R}$ is *computable* if there exists a turing machine χ such that for all $x \in S$, oracles φ of x , and $n \in \mathbb{N}$ that $\chi(\varphi, n) \in \mathbb{Q}$ and $|\chi(\varphi, n) - g(x)| < 2^{-n}$.

We will now outline the reason why this definition means that all computable functions are continuous. // TODO

3. COMPUTABLE METRIC SPACES

QUESTION: what is the role of \mathbb{Q} ? We have given notions of oracles and computability in our favorite example $(\mathbb{R}, ||, \mathbb{Q})$. Here we generalize the notion of a computable metric space to allow us to use these definitions with other spaces. There is another generalization due to Bloom, Shub, and Smale that is more algebraic and not commonly used by computer scientists.

Definition 3.1. Let (X, d_X) be a complete, separable metric space with $S = \{s_i\}$ a dense, countable subset. We say (X, d_X, S) is a *computable metric space* if there exists a turing machine $\chi : \mathbb{N}^2 \times \mathbb{Z} \rightarrow \mathbb{Q}$ such that $|\chi(i, j, n) - d_X(s_i, s_j)| < 2^{-n}$. In other words, the function $(i, j) \rightarrow d_X(s_i, s_j)$ is computable.

Definition 3.2. Let (X, d_X, S) be a computable metric space, and let $x \in X$. We say $\varphi : \mathbb{N} \rightarrow \mathbb{N}$ is an *oracle of x* if $d_X(\varphi(s_n), x) < 2^{-n}$.

Definition 3.3. Let (X, d_X, S) be a computable metric space. We say $x \in X$ is *computable* if there exists a turing machine $\chi : \mathbb{N} \rightarrow \mathbb{N}$ such that χ is an oracle of x .

Definition 3.4. Let (X, d_X, S) be a computable metric space and $Y \subseteq X$. We say $g : Y \rightarrow \mathbb{R}$ is a *computable function* if there exists a turing machine χ such that for all $x \in Y$, all oracles φ of x that $\chi(\varphi, n) \in \mathbb{Q}$ and $|\chi(\varphi, n) - g(x)| < 2^{-n}$.

QUESTION: why are we using \mathbb{R} and \mathbb{Q} here? should they be X and S ?

For a long time there wasn't a general notion of computability at a point. Prof. Wolf introduced this notion in a paper recently, which recalls the definition of continuity at a point.

Definition 3.5. Let (X, d_X, S) be a computable metric space, $Y \subseteq X$ and $x_0 \in Y$. We say a function $g : Y \rightarrow \mathbb{R}$ is *computable at x_0* if there exists a turing machine χ such that for any oracle φ of x_0 in X , $\chi(\varphi, n) \in \mathbb{Q}$ and has the following property.

Let $l_{\varphi, n}$ be the maximal precision that χ queries in the calculation of $\chi(\varphi, n)$. Then for all $y \in S$ such that there exists an oracle φ' for y which coincides with φ up to the precision $l_{\varphi, n}$ we have that $\chi(\varphi, n) = \chi(\varphi', n)$ and $|\chi(\varphi, n) - g(y)| < 2^{-n}$.

// QUESTION: should this be d_S not $||$?