UNIVERSIDAD DE SANTIAGO DE CHILE
Facultad de Ciencia
Departamento de Matemática y Ciencia de la Computación

Computability Notions for Quasi-isometries

Paul Toussaint

Profesor Guía:
Sebastián Barbieri

Tesis para optar al grado de Magíster en Ciencia en la Especialidad de Matemática

Santiago - Chile
2025

Este trabajo fue parcialmente financiado por: ANID-Subdirección de Capital Humano/Magíster Nacional/2025-22251030, ANID FONDECYT Regular 1240085 y Programa de Cooperación Científica ECOS-ANID 230003.

Resumen

Definimos una noción bien portada de cuasi-isometría computable entre espacios métricos computables y damos tres resultados. Primero, mostramos la existencia de dos espacios métricos computables X e Y tales que hay una incrustación cuasi-isométrica de X en Y pero no hay ninguna incrustación cuasi-isométrica computable. Segundo, demostramos que los grupos Fuchsianos cocompactos son todos computablemente cuasi-isométricos. Tercero, como corolario, obtenemos que los subshifts de tipo finito sobre grupos Fuchsianos cocompactos comparten la misma clase de grados de Medvedev.

Palabras clave: Análisis computable, grupos Fuchsianos, dinámica simbólica.

Abstract

We define a well-behaved notion of computable quasi-isometry between computable metric spaces and then give three results. First, we show the existence of two computable metric spaces X and Y such that there is a quasi-isometric embedding of X into Y but there is no computable quasi-isometric embedding. Second, we prove that all cocompact Fuchsian groups are computably quasi-isometric. Third, we get as a corollary that the subshifts of finite type over cocompact Fuchsian groups share the same class of Medvedev degrees.

Keywords: Computable analysis, Fuchsian groups, symbolic dynamics.

Agradecimientos

En primer lugar agradezco a mis padres, Jérôme y Esther, por su trabajo y apoyo incondicional a mi decisión de estudiar matemáticas.

Agradezco a mi profesor guía, Sebastián Barbieri, por orientarme en el mundo de la investigación y transmitirme su entusiasmo cada semana. Agradezco a los profesores: Gonzalo Robledo por su apoyo y orientación en pregrado, Claudio Gutiérrez por introducirme a la lógica y Cristóbal Rivas por introducirme a la teoría geométrica de grupos. Cada uno ha sido un pilar fundamental en mi formación.

Agradezco a todos los amigos y colegas que me han acompañado en estos 6 años. A los distinguidos de Las Palmeras: Fernando, Hjalmar, Matías, Pedro, Bastián, Chabi, David, Pipe Rivera, Kenny, Cami y Franco. A la gente de la USACH por recibirme tan cálidamente: Javier, Esteban, Jack y Cris. Agradecer también a Nicanor Carrasco por su colaboración en este trabajo.

Por último pero no menos importante, agradezco especialmente a Emir Molina por ser tan buena persona y a Mariano Ortiz por criar a este cuervo. Tengo la certeza de que seremos colegas de por vida.

Contents

Introduction

The notion of an effective procedure has been intuitively understood since at least the time of the ancient Greeks. For instance, circa 300 BC Euclid described in the Elements his well-known algorithm to compute the greatest common divisor of two integers. However, this notion was formalized only recently in the 1930s with the work of Alonzo Church, Kurt Gödel, Stephen Kleene and Alan Turing. The famous paper that Turing published in 1936 [29] is considered to be the cornerstone of computability theory.

Algorithms are ubiquitous in mathematics, so it is no surprise that computability theory has found applications in many of its branches. One instance is computable structure theory (also known as computable algebra), which is the study of algebraic structures whose operations and relations are computable (see [9], [24] or [1] for an introduction). More precisely, it studies computable presentations of structures, rather than the abstract structures by themselves. An important fact is that a particular structure, say a group, may have multiple different computable presentations or even none at all. That is to say, one can have two presentations which are isomorphic but not computably isomorphic.

Another application is computable analysis, which examines the computability properties of objects from classical analysis, such as the real numbers, continuous functions, integrable functions, etc. (see [5] and [30]). The fundamental concept in this theory are again computable presentations, but of metric spaces. One can observe a clear analogy between computable structures and computable metric spaces. Melnikov in his article [23] systematically studied this analogy and gave various examples of computably categorical metric spaces, i.e., metric spaces which have a computably unique computable presentation. In addition, he showed that the space of continuous functions on the unit interval is not computably categorical.

Relevant to this thesis is geometric group theory. This is a vast subject of mathematics greatly influenced by Gromov [13], who studied the large-scale geometry of groups, that is, the geometry preserved under quasi-isometries. Intuitively, two metric spaces are quasi-isometric if they resemble each other when looking from far away. For example $\mathbb{Z}$ and $\mathbb{R}$ are quasi-isometric. It turns out that each finitely generated group has a unique well-defined large-scale geometry, so we may speak of quasi-isometric groups. From here, one analyzes how the properties of a group are related with its large-scale geometry.

This thesis aims at exploring the applications of computable analysis to geometric group theory and more generally to large-scale geometry. Not much work seems to have beeen done in this respect. Melnikov's article [23] mentions at the very last line the possibility of studying computable quasi-isometries between computable metric spaces. The work of Khoussainov and Takisaka [17] studies the large-scale geometry of infinite strings and in particular it analyzes the computability of quasi-isometries between computable infinite strings. They asked if there exists a pair of quasi-isometric computable strings with no computable quasi-isometry between them. This was later solved in the positive in [7]. Lastly, the main inspiration for this work is the preprint [3] by Barbieri and Carrasco-Vargas, where they introduce computable quasi-isometries between groups while studying the Medvedev degrees of symbolic dynamical systems over groups.

The main contribution of this thesis is setting the foundation for this study by defining a well-behaved notion of computable quasi-isometry between computable metric spaces, where well-behaved means that it induces an equivalence relation in the class of computable metric spaces. Once this definition is in place, we give three results.

- The first result is showing that this notion does not coincide with standard quasi-isometries. More precisely, we construct a computable metric space which admits a quasi-isometric embedding of the naturals $\mathbb{N}$ but does not admit a computable quasi-isometric embedding of $\mathbb{N}$. This is analogous to the result in [7] but in a slightly different context.

- The second result is giving a non-trivial instance of computably quasi-isometric groups. We prove that all cocompact Fuchsian groups are computably quasi-isometric to the hyperbolic plane. This solves a question posed in [3].

- The last result is an application of the previous one to symbolic dynamics. By using a result from [3] we show that subshifts of finite type over cocompact Fuchsian groups share the same class of Medvedev degrees.

This dissertation is divided into four chapters. In the first one we set out the necessary background from computability theory, geometric group theory and Fuchsian groups.

The second chapter is the main one and it is divided in three sections. In the first section we introduce computable metric spaces and some basic properties. Our first two results are contained in the second and third section respectively.

The third chapter describes the application to symbolic dynamics. We introduce some basic definitions from symbolic dynamics and see how computability is applied. We then describe Medvedev reducibility and prove our last result.

Lastly, in the fourth chapter we briefly discuss further possible work and some questions which were left open.

1 Preliminaries

We start by fixing some notation. We denote by $\mathbb{N} = \{0, 1, 2, \dots\}$ the set of natural numbers. We identify each $n \in \mathbb{N}$ with the set of its predecessors $\{0, 1, \dots, n-1\}$. Given sets A, B, we write B^A for the set of all functions $f \colon A \to B$. Thus, $2^{\mathbb{N}}$ is the set of binary sequences and $\mathbb{N}^n$ is the set of $n-$tuples of natural numbers. We implicitly identify subsets $A \subseteq \mathbb{N}$ with their characteristic functions $\chi_A \in 2^{\mathbb{N}}$. We write $f :\subseteq A \to B$ to denote a *partial* function from A to B, that is, a function whose domain is a subset of A.

The set of *strings* or *words* over a set A is the set $A^{<\mathbb{N}} = \bigcup_{n=0}^{\infty} A^n$. The *length* $l(\sigma)$ of a word $\sigma \in A^{<\mathbb{N}}$ is the number n for which $\sigma \in A^n$. Thus, a word is a partial function $\sigma :\subseteq \mathbb{N} \to A$ with $\mathrm{dom}(\sigma) = l(\sigma) = \{0, 1, \dots, l(\sigma)-1\}$. When $l(\sigma) = 0$ we have the *empty word* which we denote by ε. A partial ordering $\preceq$ on $A^{<\mathbb{N}}$ is defined by

$$\sigma \preceq \tau \iff l(\sigma) \leq l(\tau) \text{ and } \sigma(i) = \tau(i) \text{ for all } 0 \leq i < l(\sigma),$$

and we say that σ is a *prefix* of τ. Similarly, if $\sigma \in A^{<\mathbb{N}}$ and $x \in A^{\mathbb{N}}$, we define

$$\sigma \prec x \iff \sigma(i) = x(i) \text{ for all } 0 \leq i < l(\sigma).$$

The *Baire space* is the set $\mathbb{N}^{\mathbb{N}}$ equipped with the prodiscrete topology. A natural basis for this topology are the sets $U_\sigma = \{x \in \mathbb{N}^{\mathbb{N}} : \sigma \prec x\}$ where $\sigma \in \mathbb{N}^{<\mathbb{N}}$. The *Cantor space* is the subspace $2^{\mathbb{N}}$ of $\mathbb{N}^{\mathbb{N}}$.

1.1 Computability theory

In this section we present the fundamental concepts of computability theory that we will use throughout this thesis. Computability theory is the study of functions of the natural numbers that can be computed through a mechanical procedure. In modern terms, this means one can write (at least theoretically) a computer program that computes the function. There are many equivalent formalizations of this notion, from which the simplest and most intuitive one is probably the one given by Alan Turing in 1936 [29], where he defines what are now called Turing machines. Once this formalization is done, a rich and highly versatile theory develops. We will describe the basic definitions and how they can be translated to sets other than the natural numbers. Then, we show some fundamental properties of the computably enumerable sets. Finally, we define oracle Turing machines, a concept which formalizes the notion of relative computability. The interested reader may consult [28] for more information about this topic.

1.1.1 Turing machines and codings

Informally, a Turing machine consists of: (1) a two-way infinite tape divided into cells, (2) a head which can read and write symbols one cell of the tape at a time, and (3) a finite set of internal states $Q = \{q_0, q_1, \dots, q_n\}$. Each cell of the tape may be blank (B) or have the symbol 1 on it. The machine starts on the state q_1 reading some cell. Based on the current state and the scanned symbol, the machine does three things in a single step: change from one state to another, change the scanned symbol to another one from $S = \{1, B\}$, and move the head one cell to the right (R) or left (L). The behavior of the machine is determined by a partial function $\delta :\subseteq Q \times S \to Q \times S \times \{R, L\}$ called its program. We interpret this function as follows: if $\delta(q_i, s) = (q_j, s', X)$ then, when the machine is in state q_i reading the symbol s, in the next step it will change to the state q_j, replace the symbol s by s' and lastly move one cell to the right if $X = R$ (left if $X = L$). The machine keeps going in this way until it reaches the state q_0 called the halting state.

As we can see, a Turing machine is completely determined by its program, so formally we define a Turing machine as just a program.

Definition 1.1.1. A Turing machine (or Turing program) is a partial function $\delta :\subseteq Q \times S \to Q \times S \times \{R, L\}$ where $Q = \{q_0, q_1, \dots, q_n\}$ is a finite set called the states of the machine and $S = \{1, B\}$ is called the alphabet.

In order to compute with a Turing machine we need to set up some input and output conventions.

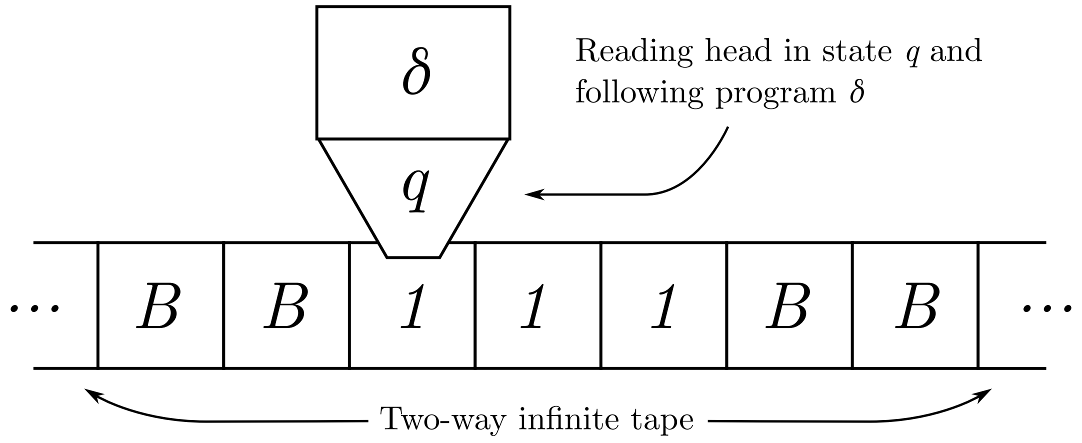

Figure 1: Turing machine

Input convention: To input $n \in \mathbb{N}$, place $n+1$ consecutive 1's on the tape with the rest of the cells blank. Then set the head in the starting state q_1 reading the leftmost 1.
Output convention: If the machine ever reaches the halting state q_0, we say that the machine halts and the output is the total number of 1's left on the tape.

If the machine reaches a position (q_i, s) which is not in the domain of δ, or it never reaches the halting state, we say that the machine does not halt and we have no output.

Observe that a machine may halt in some inputs and not halt in others. So, Turing machines compute *partial* functions of the natural numbers.

Definition 1.1.2. We say that a Turing machine *computes* the partial function $\psi :\subseteq \mathbb{N} \to \mathbb{N}$ when $\psi(x) = y$ if and only if the machine with input x eventually halts and yields output y. In that case, ψ is said to be a partial computable (p.c.) function.

For example, the following machine computes the function $f(x) = 0$ for all $x \in \mathbb{N}$.

$$\begin{aligned}\delta(q_1, 1) &= (q_1, B, R)\\ \delta(q_1, B) &= (q_0, B, R).\end{aligned}$$

This next one computes the function $f(x) = x + 3$.

$$\begin{aligned}\delta(q_1, 1) &= (q_1, 1, R)\\ \delta(q_1, B) &= (q_2, 1, R)\\ \delta(q_2, B) &= (q_0, 1, R).\end{aligned}$$

A Turing machine may also compute functions of k variables with the folowing input convention: The vector $(n_1, \ldots, n_k)$ gets coded on the tape as

$$\underbrace{1\ldots1}_{n_1+1 \text{ times}} \; B \; \underbrace{1\ldots1}_{n_2+1 \text{ times}} \; B \ldots B \; \underbrace{1\ldots1}_{n_k+1 \text{ times}}$$

For example, the following machine computes the function $f(x, y) = x + y$.

$$\begin{aligned}\delta(q_1, 1) &= (q_2, B, R)\\ \delta(q_2, 1) &= (q_2, 1, R)\\ \delta(q_2, B) &= (q_3, B, R)\\ \delta(q_3, 1) &= (q_0, B, R).\end{aligned}$$

Turing machines try to capture the notion of an "effective process", or what we call now an algorithm. One may ask if Turing machines really achieve this, maybe there is a function which we consider intuitively computable but cannot be computed by a Turing machine. Over time, many people have come up with different attempts to formalize the notion of algorithm but they all turned to be equivalent to Turing's definition. Moreover, Turing in his original paper gives a philosophical argument for why his machines capture all mechanical processes. Simultaneously, his advisor Alonzo Church made a similar argument for his own formalization, the lambda calculus. This way, they formulated their famous thesis.

Church-Turing's thesis. Every "intuitively" computable function is computable by a Turing machine.

Of course, this claim cannot be formally proven but it is widely accepted as true. We will implicitly use Church-Turing's thesis to prove that a certain function is computable by giving an informal algorithm that computes it instead of the full Turing program.

Now that we have formalized computability on the natural numbers, we can extend this formalization to other countable sets by means of *codings*. This is the fundamental technique that Gödel used in the proofs of his famous incompleteness theorems, where he assigned a so called Gödel number to each logical formula.

Definition 1.1.3. A coding for a set A is a surjection $\nu\colon \mathbb{N} \twoheadrightarrow A$. If $a \in A$ and $n \in \mathbb{N}$ are such that $\nu(n) = a$, we say that n is a *ν-code* of a.

Let ν and μ be codings for the sets A and B respectively. A function $f\colon A \to B$ is said to be *(ν, μ)-computable* (or simply *computable*) if there exists a computable function $F\colon \mathbb{N} \to \mathbb{N}$ such that for all $n \in \mathbb{N}$, $f \circ \nu(n) = \mu \circ F(n)$. That is, the following diagram commutes:

$$\begin{tikzcd}
A \arrow[r, "f"] & B \\
\mathbb{N} \arrow[u, "\nu"] \arrow[r, "F"] & \mathbb{N} \arrow[u, "\mu"]
\end{tikzcd}$$

Example 1.1.4 (Basic codings)**.** The following codings and notations will be used extensively throughout this thesis.

- A coding for $\mathbb{Z}$ is given by its standard bijection with $\mathbb{N}$,

$$\nu(n) = \begin{cases} n/2 & \text{if } n \text{ is even,} \\ -(n+1)/2 & \text{if } n \text{ is odd.} \end{cases}$$

- Given $n, m \in \mathbb{N}$, we denote

$$\langle n, m \rangle = \frac{1}{2}(n+m)(n+m+1) + m.$$

  This defines a bijection from $\mathbb{N}^2$ to $\mathbb{N}$ called the *standard pairing* function. Its inverse is a bijective coding for $\mathbb{N}^2$. Moreover, it is computable, so $\mathbb{N}$ and $\mathbb{N}^2$ are essentially the same from the point of view of computability.

- Continuing inductively, we have computable bijections from $\mathbb{N}^k$ to $\mathbb{N}$ for any $k \geq 1$,

$$\langle n_0, \dots, n_{k-1} \rangle = \langle \langle n_0, \dots, n_{k-2} \rangle, n_{k-1} \rangle.$$

- Let ν and μ be codings for the sets A and B respectively, we define a coding $\nu \times \mu$ for the set $A \times B$ in the following way. For each $u \in \mathbb{N}$, let $\langle (u)_0, (u)_1 \rangle = u$, then define

$$(\nu \times \mu)(u) = (\nu((u)_0), \mu((u)_1)).$$

- The rational numbers $\mathbb{Q}$ may be coded as

$$\nu(n) = (-1)^{(n)_0} \frac{(n)_1}{(n)_2 + 1}, \text{ where } \langle (n)_0, (n)_1, (n)_2 \rangle = n.$$

  It is not hard to see that addition $(q, r) \mapsto q + r$ and multiplication $(q, r) \mapsto q \cdot r$ are $(\nu \times \nu, \nu)$-computable functions. Note that this coding is not bijective, but we can easily make it so and preserve the computability of the operations. (See example 2.1.2)

- A coding for the set $\mathbb{N}^{<\mathbb{N}}$ of finite strings of natural numbers will be very useful to have. Given $\sigma \in \mathbb{N}^{<\mathbb{N}}$, we denote

$$\langle \sigma \rangle = \begin{cases} 0 & \text{if } \sigma \text{ is the empty word } \varepsilon, \\ \langle l(\sigma)-1, \langle \sigma(0), \sigma(1), \dots, \sigma(l(\sigma)-1)\rangle\rangle + 1 & \text{otherwise.} \end{cases}$$

  where $l(\sigma)$ is the length of σ. This is a bijection between $\mathbb{N}^{<\mathbb{N}}$ and $\mathbb{N}$, so its inverse is a bijective coding called the *standard coding* for $\mathbb{N}^{<\mathbb{N}}$. One has to be careful since $\langle . \rangle$ denotes a different function depending if the argument is a tuple or a string, so $\langle a, b, c\rangle \neq \langle abc \rangle$ for example.

  One easily checks the following functions and relations are computable under the standard coding:

  - The length function $\sigma \mapsto l(\sigma)$.
  - The evaluation function $(\sigma, i) \mapsto \sigma(i), i < l(\sigma)$.
  - Concatenation $(\sigma, \tau) \mapsto \sigma\tau$.
  - The prefix relation $\preceq$.

A very important coding is one for the set of partial computable functions. Recall that a Turing program is a partial function $\delta :\subseteq Q \times \{B, 1\} \to Q \times \{B, 1\} \times \{R, L\}$ where Q is a finite set. In other words, δ is a finite set of quintuples of the form (q_i, s, q_j, s', r), where, $i, j \in \mathbb{N}$, $s, s' \in \{B, 1\}$ and $r \in \{R, L\}$. Moreover, this set is consistent in the sense that if $(q_i, s, q_j, t, r) \in \delta$ and $(q_i, s, q_k, t', r') \in \delta$ then $q_j = q_k, t = t'$ and $r = r'$.

We define a coding for the set of all quintuples in the following way. Denote $s_0 = B, s_1 = 1, r_0 = R$ and $r_1 = L$, then a quintuple of numbers $\langle i, k, j, l, m\rangle$ corresponds to the quintuple $(q_i, s_k, q_j, s_l, r_m)$. More precisely, for each $t \in \mathbb{N}$, let $\langle i, k, j, l, m\rangle = t$ and define

$$\nu(t) = \begin{cases} (q_i, s_k, q_j, s_l, r_m) & \text{if } 0 \leq k, l, m \leq 1, \\ (q_0, s_0, q_0, s_0, r_0) & \text{otherwise.} \end{cases}$$

Now we code all programs. For each $e \in \mathbb{N}$, let $\sigma \in \mathbb{N}^{<\mathbb{N}}$ be the string coded by e and write

$$P_e = \{\nu(\sigma(0)), \nu(\sigma(1)), \dots, \nu(\sigma(l(\sigma)-1))\},$$

only if this is a consistent set of quintuples, otherwise we set $P_e = \emptyset$ the empty program.

We say that P_e is the e-th Turing program. We denote by $\varphi_e^{(n)}$ the partial function of n variables computed by P_e. We abbreviate $\varphi_e^{(1)}$ as φ_e.

Now that we have coded the partial computable functions, consider the function $\Psi :\subseteq \mathbb{N}^2 \to \mathbb{N}$, defined by $\Psi(e, x) = \varphi_e(x)$. One can see that, in the spirit of Church-Turing's thesis, Ψ is computable: Take e, find the string σ that it encodes, decode each component of σ into a quintuple and thus recover the list of quintuples. Then, follow the instructions from this list with x as input until it halts, in that case, output the result. This is the fundamental result of computability.

Theorem 1.1.5 (Enumeration Theorem)**.** *The partial function $\Psi :\subseteq \mathbb{N}^2 \to \mathbb{N}$ defined by $\Psi(e, x) = \varphi_e(x)$ is computable. Thus, there exists z such that $\varphi_z^{(2)}(e, x) = \varphi_e(x)$.*

Formal proofs of the enumeration theorem may be found in elementary computability theory textbooks such as [28] and [19].

1.1.2 Computably enumerable sets

As usual in computability theory, we now forget about Turing machines and start to build the theory upon the enumeration theorem and Church-Turing's thesis.

Definition 1.1.6. Let $A \subseteq \mathbb{N}^n$ be a n-ary relation of natural numbers. Notice that, when $n = 1$, A is just a set.

- A is said to be computable (or decidable) if its characteristic function $\chi_A \colon \mathbb{N}^n \to \{0, 1\}$ is computable.

- A is said to be computably enumerable (c.e.) if it is the domain of some partial computable function $\varphi : A \subseteq \mathbb{N}^n \to \mathbb{N}$.
- A is said to be co-computably enumerable (co-c.e.) if its complement $A^c = \{x \in \mathbb{N}^n : x \notin A\}$ is c.e.

Proposition 1.1.7. *If A is a computable set, then it is c.e.*

Proof. Let ψ be the restriction of χ_A to A, that is,

$$\psi(x) = \begin{cases} 1 & \text{if } x \in A, \\ \emptyset & \text{if } x \notin A, \end{cases}$$

where $\emptyset$ means the function is undefined. Since χ_A is computable, ψ is computed by the following algorithm: Given x, evaluate $\chi_A(x)$. If $\chi_A(x) = 1$ output 1. If $\chi_A(x) = 0$ enter in an infinite loop so that we never halt. Therefore ψ is a p.c. function with domain A. □

Intuitively speaking, a set A is computable if we have an effective procedure that always halts on every input n and answers whether $n \in A$ or not. A set A is computably enumerable if we have an effective procedure that halts and gives an answer when $n \in A$, but goes on forever when $n \notin A$. Because of this, computably enumerable sets are also known as *semidecidable* sets.

The natural question arises: Are there non-computable sets? The answer is yes. The proof of this is a classical diagonalization argument.

Definition 1.1.8. Given a partial function ψ, if $x \in \operatorname{dom} \psi$ we say that $\psi(x)$ *converges*, which we write as $\psi(x)\downarrow$, otherwise, we say that $\psi(x)$ *diverges* ($\psi(x)\uparrow$).

Theorem 1.1.9. *The set $K = \{x : \varphi_x(x)\downarrow\}$ is c.e. but not computable.*

Proof. K is the domain of the partial computable function $\psi(x) = \Psi(x, x) = \varphi_x(x)$ from the enumeration theorem, so it is c.e. (This can also be seen by the Church-Turing's thesis, since we have a procedure to semidecide K). Now suppose that K had a computable characteristic function. Then the following function would be computable

$$f(x) = \begin{cases} \varphi_x(x) + 1 & \text{if } x \in K, \\ 0 & \text{if } x \notin K. \end{cases}$$

However, $f \neq \varphi_e$ for any e. Indeed, if $e \in K$, then $f(e) = \varphi_e(e) + 1 \neq \varphi_e(e)$. If $e \notin K$, then $f(e) = 0$ and $\varphi_e(e)\uparrow$. □

Thus, there is no algorithm that decides for any x whether $x \in K$ or not. This is what is known as an unsolvable problem. The *halting problem* is to decide for any x and y whether $\varphi_x(y)$ converges, that is, whether program P_x with input y halts. The unsolvability of K lets us establish the unsolvability of the halting problem.

Corollary 1.1.10. *The relation $K_0 = \{(x, y) : \varphi_x(y)\downarrow\}$ is not computable.*

Proof. Observe that $x \in K$ iff $(x, x) \in K_0$. So, if K_0 had a computable characteristic function, so would K, contrary to theorem 1.1.9. □

We now turn to some additional properties of c.e. sets and more generally c.e. relations. We have seen that they are not necessarily computable, but we will see that they are approximated by computable relations in some sense.

Definition 1.1.11. Let $n \geq 1$. We write $\varphi^{(n)}_{e,s}(x_1, \ldots, x_n) = y$ if $\max(x_1, \ldots, x_n, y, e) < s$ and y is the output of $\varphi^{(n)}_e(x_1, \ldots, x_n)$ in less than s steps of the Turing program P_e. If such a y exists we say that $\varphi^{(n)}_{e,s}(x_1, \ldots, x_n)$ *converges*, which we denote by $\varphi^{(n)}_{e,s}(x_1, \ldots, x_n)\downarrow$, otherwise, $\varphi^{(n)}_{e,s}(x_1, \ldots, x_n)$ *diverges* ($\varphi^{(n)}_{e,s}(x_1, \ldots, x_n)\uparrow$). Again, when $n = 1$ we abbreviate $\varphi^{(1)}_{e,s}(x)$ to $\varphi_{e,s}(x)$.

To reduce notation we abbreviate $\vec{x} = (x_1, \ldots, x_n) \in \mathbb{N}^n$.

Proposition 1.1.12. *Let $n \geq 1$. The following relations are computable:*

- $\{(e, \vec{x}, s) : \varphi_{e,s}^{(n)}(\vec{x}) \downarrow\}$,
- $\{(e, \vec{x}, y, s) : \varphi_{e,s}^{(n)}(\vec{x}) = y\}$.

Proof. We give an informal proof with Church-Turing's thesis. Given $e, \vec{x}, s$, we can decide $\varphi_{e,s}^{(n)}(\vec{x}) \downarrow$ by retrieving the program P_e and following s instructions with $\vec{x}$ as input. This procedure always terminates because we have a bound on the number of steps. The second relation is similar. □

Notice that since every c.e. relation is the domain of some partial computable function, we have a coding of the c.e. relations for each $n \geq 1$.

Definition 1.1.13. We denote the e-th c.e. n-ary relation by

$$W_e^{(n)} = \text{dom } \varphi_e^{(n)} = \{\vec{x} : \varphi_e^{(n)}(\vec{x}) \downarrow\},$$

and

$$W_{e,s}^{(n)} = \text{dom } \varphi_{e,s}^{(n)} = \{\vec{x} : \varphi_{e,s}^{(n)}(\vec{x}) \downarrow\}.$$

We abbreviate $W_e^{(1)} = W_e$ and $W_{e,s}^{(1)} = W_{e,s}$.

Note that $\varphi_e(x) = y$ iff $\varphi_{e,s}(x) = y$ for some s and $x \in W_e$ iff $x \in W_{e,s}$ for some s. Thus, $W_e = \bigcup_{s \geq 0} W_{e,s}$.

Definition 1.1.14. Let $A \subseteq \mathbb{N}^n$ be a n-ary relation.

- A is said to be a Σ_1 relation if for all $\vec{x} \in \mathbb{N}^n$ we have $\vec{x} \in A \iff (\exists y)R(\vec{x}, y)$, for some computable relation $R \subseteq \mathbb{N}^{n+1}$.
- A is said to be a Π_1 relation if for all $\vec{x} \in \mathbb{N}^n$ we have $\vec{x} \in A \iff (\forall y)R(\vec{x}, y)$, for some computable relation $R \subseteq \mathbb{N}^{n+1}$.
- A is said to be a Δ_1 relation if it is both Σ_1 and Π_1.

This definition is part of a more general concept called the arithmetical hierarchy which we will not use so we omit its description, see [28, Chapter IV].

Theorem 1.1.15 (Normal Form Theorem)**.** *A relation A is c.e. iff A is Σ_1.*

Proof. We will assume that A is a 1-ary relation, the general case follows the same proof. Let A be c.e., then, $A = W_e$ for some e. Hence,

$$x \in W_e \iff (\exists s)[x \in W_{e,s}] \iff (\exists s)[\varphi_{e,s}(x) \downarrow].$$

By proposition 1.1.12 the relation $\{(e, x, s) : \varphi_{e,s} \downarrow\}$ is computable, therefore, A is Σ_1.

Let $A = \{x : (\exists y)\, R(x, y)\}$, with R computable. We define ψ the following partial function,

$$\psi(x) = \begin{cases} \min\{y : R(x, y)\} & \text{if } (\exists y)R(x, y), \\ \emptyset & \text{otherwise.} \end{cases}$$

Clearly, dom $\psi = A$. Now, ψ is computed by the following algorithm: Given x, check $R(x, 0)$, $R(x, 1)$, $R(x, 2), \ldots$ until we find the first y such that $R(x, y) = 1$ and set it as output. Note that this always terminates for x in the domain since R is computable. Therefore, A is c.e. □

Corollary 1.1.16. *A relation A is co-c.e. iff A is Π_1.*

Proof. Observe that A is Π_1 iff its complement A^c is Σ_1 since $[\vec{x} \in A \iff (\forall y)\, R(\vec{x}, y)]$ is equivalent to $[\vec{x} \notin A \iff (\exists y)\, \neg R(\vec{x}, y)]$ and $\neg R$ is still a computable relation. □

Corollary 1.1.17. *Let $A \subseteq \mathbb{N}^{n+1}$ be a c.e. relation, then the relation $B \subseteq \mathbb{N}^n$ defined by*

$$\vec{x} \in B \iff (\exists y)A(\vec{x}, y),$$

is c.e. We say that c.e. relations are closed by existential quantification.

Proof. By the normal form theorem, there is a computable relation $R \subseteq \mathbb{N}^{n+2}$ such that $A(\vec{x}, y) \iff (\exists z)R(\vec{x}, y, z)$. So,

$$B(\vec{x}) \iff (\exists y)(\exists z)R(\vec{x}, y, z).$$

Define $S \subseteq \mathbb{N}^{n+1}$ by

$$S(\vec{x}, y) \iff (y = \langle (y)_0, (y)_1 \rangle) \text{ and } R(\vec{x}, (y)_0, (y)_1),$$

where $\langle . \rangle$ denotes the standard pairing function from example 1.1.4. Since this function and R are computable, it is easy to see that S is computable, moreover, $B(\vec{x}) \iff (\exists y)S(\vec{x}, y)$. Therefore, B is Σ_1 and thus, c.e. $\square$

Now we can prove a more intuitive characterization of the computably enumerable sets which justifies their name.

Theorem 1.1.18 (Listing Theorem). *A set A is c.e. iff $A = \emptyset$ or A is the range of a total computable function f.*

Proof. The case $A = \emptyset$ is trivial. Suppose that A is the range of a total computable function f, then,

$$x \in A \iff (\exists s)[f(s) = x].$$

Since f is total computable, the relation $\{(s, x) : f(s) = x\}$ is clearly computable. Therefore A is Σ_1 and thus c.e. by the normal form theorem.

Now, let A be non-empty c.e., that is, $A = W_e \neq \emptyset$ for some e. Let $a \in A$, and define the total function f,

$$f(\langle x, s \rangle) = \begin{cases} x & \text{if } \varphi_{e,s}(x)\downarrow, \\ a & \text{otherwise.} \end{cases}$$

Here, we are using the standard pairing function $\langle ., . \rangle : \mathbb{N}^2 \to \mathbb{N}$ which is bijective and computable, so f is well defined and it is clearly computable: Given u, decompose it into a pair $\langle x, s \rangle$ with the inverse of the pairing function and then check $\varphi_{e,s}(x)\downarrow$ which is a computable relation. Furthermore,

$$x \in A \iff \varphi_e(x)\downarrow \iff (\exists s)[\varphi_{e,s}(x)\downarrow] \iff (\exists s)[f(\langle x, s \rangle) = x],$$

so that $A = \text{range}(f)$. $\square$

Corollary 1.1.19. *An infinite c.e. set A is the range of an injective total computable function f.*

Proof. Let g be a total computable function with $\text{range}(g) = A$. Define f recursively in the following way:

$$f(0) = g(0),$$
$$f(n+1) = g(\min\{y : g(y) \neq f(i)\, \forall i \leq n\}).$$

Since A is infinite, f is total and clearly computable. By construction, if $i < j$, then $f(i) \neq f(j)$, so f is injective. $\square$

This last theorem relates c.e. sets to computable sets.

Theorem 1.1.20 (Complementation Theorem). *Let $A \subseteq \mathbb{N}$ be a set. A is computable iff both A and its complement A^c are c.e.*

Proof. If A is computable then A^c is also computable so both A and A^c are computable.

Let A and A^c be c.e. By the listing theorem, there are computable functions f, g such that $A = \text{range}(f)$ and $A^c = \text{range}(g)$. We now describe an algorithm for deciding $x \in A$ or not: Compute $f(0), g(0), f(1), g(1), \dots,$ until we find a s such that $f(s) = x$ or $g(s) = x$, then, $x \in A$ if $f(s) = x$ and $x \notin A$ if $g(s) = x$. Notice that s always exists because $A \cup A^c = \mathbb{N}$ so this procedure always halts. $\square$

By the normal form theorem, this means that the computable sets are exactly the Δ_1 sets.

1.1.3 Oracle Turing machines

From theorem 1.1.9 we have an example of a non-computable set. We may think of this set as containing information not accesible to a Turing machine. An oracle Turing machine is a more powerful machine which can access this information via an "oracle". It is very similar to a standard Turing machine, but it has an extra read only tape, called the *oracle tape*, over which the characteristic function of some set A (called the *oracle*) is encoded. The old tape is called the *work tape* and operates just as before. The reading head moves along both tapes simultaneously.

Definition 1.1.21. An oracle Turing machine (or oracle Turing program) is a partial function $\delta :\subseteq Q \times S_1 \times S_2 \to Q \times S_2 \times \{R, L\}$, where $Q = \{q_0, q_1, \ldots, q_n\}$ is a finite set of states, $S_1 = \{B, 0, 1\}$ is the oracle tape alphabet, $S_2 = \{B, 1\}$ is the work tape alphabet and $\{R, L\}$ are the head moving operations.

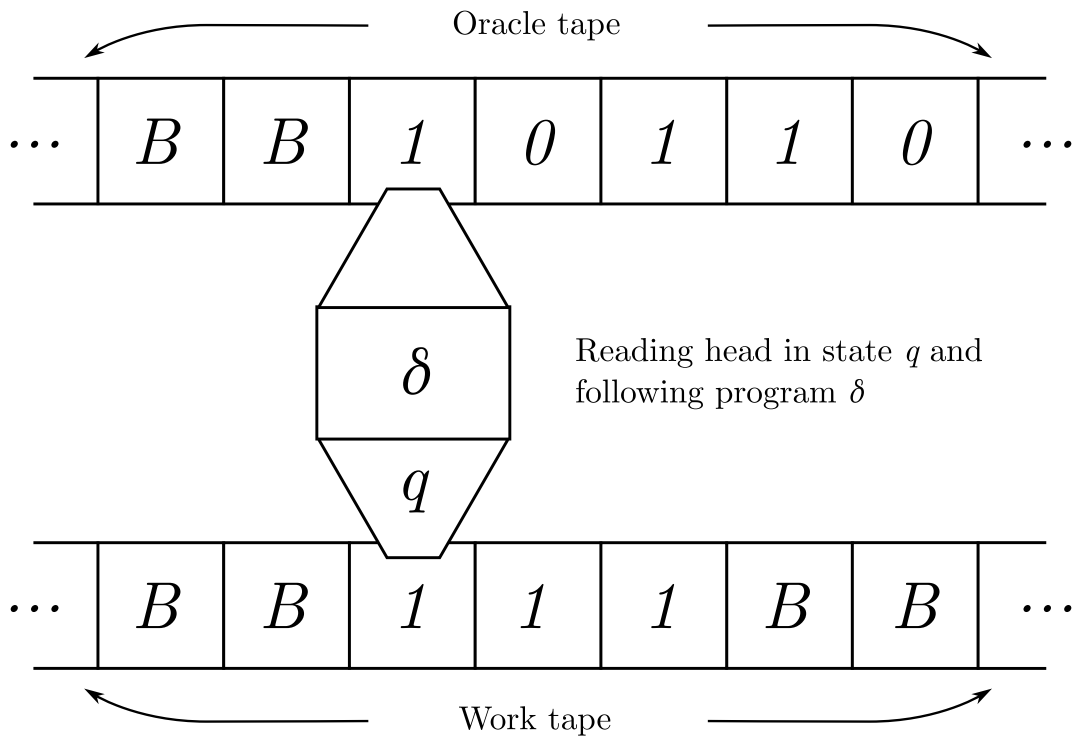

Figure 2: Oracle Turing machine

We interpret $\delta(q_i, a, b) = (q_j, c, X)$ as follows: when the machine is in state q_i reading the symbol a on the oracle tape and symbol b on the working tape, then in the next step, it will pass to the state q_j, replace the symbol b by c on the work tape and move one cell to the right on both tapes if $X = R$ (left if $X = L$).

Notice that now we are working with two inputs: a set $A \subseteq \mathbb{N}$ and a natural number n. The input and output conventions are the same as for standard Turing machines, except that the reading head also starts on the cell of the oracle tape which codes $\chi_A(0)$.

Of course, these new oracle Turing programs are just finite sets of sixtuples, so we can code them as we did with normal Turing programs. From now on, we fix some coding of all oracle Turing programs and we denote $\hat{P}_e$ the e-th such program.

Definition 1.1.22. A partial function ψ is *computable relative to* A, if there is a program $\hat{P}_e$ such that if the machine has χ_A written on the oracle tape, then for all x and y, $\psi(x) = y$ iff $\hat{P}_e$ on input x halts and yields output y. In this case we write $\psi = \Phi_e^A$.

A program $\hat{P}_e$ is independent of the oracle A. In effect, a program computes a *functional* which takes as input a set A and outputs a partial function ψ. We denote Φ_e the functional computed by $\hat{P}_e$.

The intuitive notion behind oracle Turing machines is the one of relative computability: Let $A \subseteq \mathbb{N}$ and assume we have a magic 8-ball which instantly answers any question of the form "is $n \in A$?", then, one asks what else can one compute with this additional ability? By analogy, most results from standard computability *relativize*, for example the Church-Turing's thesis.

Relativized Church-Turing's thesis. Every "intuitively" computable function relative to A is computable by a Turing machine with an oracle to A.

In this next definition, binary strings $\sigma \in 2^{<\mathbb{N}}$ are to be viewed as finite initial segments of characteristic functions. We identify sets with their characteristic functions, so, $\sigma \prec A$ means that σ is a prefix of χ_A. Also, recall from example 1.1.4 the bijective coding for strings $\mathbb{N}^{<\mathbb{N}}$, so we may speak for example of computable sets of strings by just identifying them with their respective code numbers.

Definition 1.1.23. We write $\Phi^A_{e,s}(x) = y$ if $\max(x, y, e) < s$, $s > 0$, $\Phi^A_e(x) = y$ in less than s steps of the program $\hat{P}_e$, and only the first s cells from the oracle tape are used in the computation.

Let $\sigma \in 2^{<\mathbb{N}}$. We write $\Phi^\sigma_{e,s}(x) = y$, if $\Phi^A_{e,s}(x) = y$ for some $A \succ \sigma$, and only elements $z < l(\sigma)$ from the oracle tape are used in the computation.

We write $\Phi^\sigma_e(x) = y$ if $(\exists s)[\Phi^\sigma_{e,s}(x) = y]$.

Theorem 1.1.24. *The relation* $\{(e, \sigma, x, y, s) : \Phi^\sigma_{e,s}(x) = y\}$ *is computable.*

Proof. As in the enumeration theorem, we give an informal proof with the Church-Turing's thesis: Retrieve the program $\hat{P}_e$, write σ on the oracle tape and x on the work tape. Do the computation until an output occurs or the first s steps have been completed or the head steps outside of σ. If an output has occured, check if it is equal to y. This procedure always terminates because there is a bound on the numbers of steps. □

Theorem 1.1.25 (Use Principle)**.**

1. $\Phi^A_e(x) = y \implies (\exists s)(\exists \sigma \prec A)[\Phi^\sigma_{e,s}(x) = y]$.
2. $\Phi^\sigma_{e,s}(x) = y \implies (\forall t \geq s)(\forall \tau \succeq \sigma)[\Phi^\tau_{e,t}(x) = y]$.
3. $\Phi^\sigma_e(x) = y \implies (\forall A \succ \sigma)[\Phi^A_e(x) = y]$.

Proof. The proof follows directly from the definition of oracle computation. For (1) any computation that halts does so after finitely many steps and having used only finitely many elements from the oracle tape. (2) and (3) follow at once. □

What makes relative computability useful is that it allows us to compare the complexity of sets of numbers.

Definition 1.1.26. Let $A, B \in 2^{\mathbb{N}}$. We say that A is *Turing reducible* to B (written $A \leq_T B$) if χ_A is computable relative to B.

We may think of the set B containing more information, or being more complex, than A. Clearly, the relation $\leq_T$ on $2^{\mathbb{N}}$ is reflexive and transitive, so it induces a equivalence relation $\equiv_T$ defined by $A \equiv_T B$ iff $A \leq_T B$ and $B \leq_T A$. The equivalence class of A under $\equiv_T$ is called the *Turing degree* of A.

For example, from the definitions it is direct that $A \leq_T \emptyset$ if and only if A is computable. Moreover, $\emptyset$ is reducible to any set. This means that the computable sets form a single Turing degree which is the minimum under $\leq_T$. We interpret this as computable sets containing no information. On the other hand, the set K from theorem 1.1.9 is not reducible to $\emptyset$ so it contains strictly more information than the computable sets. In an intuitive sense, K can look ahead into infinite time and predict when a computation will halt, whereas computable sets cannot.

Importantly for this work, oracle Turing machines also allow us to extend notions of computability to other sets which are not necessarily countable. The starting point is computability on the Cantor and Baire spaces, afterwards in chapter 2, we will speak of computability on metric spaces.

Definition 1.1.27. Let $F :\subseteq 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ be a partial functional. We say that F is computable if there exists a Turing functional Φ_e such that $\Phi^A_e = F(A)$ for every $A \in \mathrm{dom}(F)$.

In this way, definition 1.1.26 may be rephrased as $A \leq_T B$ iff there exists a computable $F :\subseteq 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ such that $B \in \operatorname{dom}(F)$ and $A = F(B)$.

To define computability on the Baire space $\mathbb{N}^{\mathbb{N}}$ we need a way to code its elements into the oracle tape. This is straightforward: Given $f \in \mathbb{N}^{\mathbb{N}}$, let

$$\hat{f} = 1^{f(0)}01^{f(1)}01^{f(2)} \cdots \in 2^{\mathbb{N}},$$

where 1^n is the string of n consecutive 1's. It is easy to see that the map $f \mapsto \hat{f}$ is injective and even a topological embedding of $\mathbb{N}^{\mathbb{N}}$ into $2^{\mathbb{N}}$. We write similarly for strings: Given $\sigma \in \mathbb{N}^{<\mathbb{N}}$, let

$$\hat{\sigma} = 1^{\sigma(0)}01^{\sigma(1)}0 \ldots 01^{\sigma(l(\sigma)-1)}0 \in 2^{<\mathbb{N}}.$$

The map $\sigma \mapsto \hat{\sigma}$ is a computable injection from $\mathbb{N}^{<\mathbb{N}}$ to $2^{<\mathbb{N}}$. Finally, notice that if $\sigma \prec f$ then $\hat{\sigma} \prec \hat{f}$.

Definition 1.1.28. Let $f \in \mathbb{N}^{\mathbb{N}}$. A partial function ψ is *computable relative to f* if there is a program $\hat{P}_e$ such that if the machine has $\hat{f}$ written on the oracle tape, then for all x and y, $\psi(x) = y$ iff $\hat{P}_e$ on input x halts and yields output y. In this case we write $\psi = \Phi_e^f$.

As in the Cantor space, we say that $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ is computable if there exists Φ_e such that $\Phi_e^f = F(f)$ for all $f \in \operatorname{dom}(F)$.

When doing computability on $\mathbb{N}^{\mathbb{N}}$, topological considerations become very important. Recall that the Baire space comes with a topology generated by open sets of the form $U_\sigma = \{x \in \mathbb{N}^{\mathbb{N}} : \sigma \prec x\}$, where $\sigma \in \mathbb{N}^{<\mathbb{N}}$.

Definition 1.1.29. A set $V \subseteq \mathbb{N}^{\mathbb{N}}$ is called *effectively open* or Σ_1^0 if there is a c.e. set $I \subseteq \mathbb{N}^{<\mathbb{N}}$ such that

$$V = \bigcup_{\sigma \in I} U_\sigma.$$

A set $C \subseteq \mathbb{N}^{\mathbb{N}}$ is called *effectively closed* or Π_1^0 if its complement $\mathbb{N}^{\mathbb{N}} \setminus C$ is effectively open, i.e, there exists a c.e. set $I \subseteq \mathbb{N}^{<\mathbb{N}}$ such that

$$C = \mathbb{N}^{\mathbb{N}} \setminus \left(\bigcup_{\sigma \in I} U_\sigma \right) = \bigcap_{\sigma \in I} (\mathbb{N}^{\mathbb{N}} \setminus U_\sigma).$$

By the listing theorem 1.1.18, a non-empty set U is effectively open if and only if there is a computable function $f \colon \mathbb{N} \to \mathbb{N}^{<\mathbb{N}}$ such that $U = \bigcup_{n \in \mathbb{N}} U_{f(n)}$. We can think of this as being able to effectively approximate U from the inside. Similarly, effectively closed sets are closed sets that we can approximate from the outside with an algorithm.

In contrast to regular open sets, effectively open sets are not closed under arbitrary unions. We need stronger computability hypotheses in order to have a closure property.

Definition 1.1.30. A sequence of sets $\{V_n\}_{n \in \mathbb{N}}$ of $\mathbb{N}^{\mathbb{N}}$ are said to be *uniformly open* or *uniformly* Σ_1^0 if there is a c.e. relation $I \subseteq \mathbb{N} \times \mathbb{N}^{<\mathbb{N}}$ such that

$$V_n = \bigcup \{U_\sigma : (n, \sigma) \in I\}, \text{ for all } n \in \mathbb{N}.$$

A sequence of sets $\{C_n\}_{n \in \mathbb{N}}$ of $\mathbb{N}^{\mathbb{N}}$ are said to be *uniformly closed* or *uniformly* Π_1^0 if there is a c.e. relation $I \subseteq \mathbb{N} \times \mathbb{N}^{<\mathbb{N}}$ such that

$$C_n = \bigcap \{(\mathbb{N}^{\mathbb{N}} \setminus U_\sigma) : (n, \sigma) \in I\}, \text{ for all } n \in \mathbb{N}.$$

Proposition 1.1.31. *If $\{V_n\}_{n \in \mathbb{N}}$ are uniformly open, then $\bigcup_{n \in \mathbb{N}} V_n$ is effectively open. Similarly, if $\{C_n\}_{n \in \mathbb{N}}$ are uniformly closed, then $\bigcap_{n \in \mathbb{N}} C_n$ is effectively closed.*

Proof. Let $I \subseteq \mathbb{N} \times \mathbb{N}^{<\mathbb{N}}$ be c.e. such that $V_n = \bigcup \{U_\sigma : (n, \sigma) \in I\}$ for all $n \in \mathbb{N}$. Let $J \subseteq \mathbb{N}^{<\mathbb{N}}$ be the set defined by $\sigma \in J \iff (\exists n) I(n, \sigma)$, then by corollary 1.1.17, J is c.e. Moreover, $\bigcup_{n \in \mathbb{N}} V_n = \bigcup_{\sigma \in J} U_\sigma$, so it is effectively open. The second part is analogous. □

We say that *effective unions* of effectively open sets are effectively open and *effective intersections* of effectively closed sets are effectively closed.

Definition 1.1.32. A partial function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ is said to be *effectively continuous* if there is a sequence $\{V_\sigma\}_{\sigma\in\mathbb{N}^{<\mathbb{N}}}$ of uniformly open sets such that $F^{-1}(U_\sigma) = \mathrm{dom}(F) \cap V_\sigma$ for all $\sigma \in \mathbb{N}^{<\mathbb{N}}$.

Thus, an effectively continuous function not only pullbacks open sets to open sets, we also have an effective procedure to uniformly approximate them. The following theorem shows the close relationship between computability and topology. One may argue that computability is a refinement of topology.

Theorem 1.1.33. *A partial functional $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ is computable if and only if it is effectively continuous.*

Proof. Suppose F is computable. Let Φ_e be such that $\Phi_e^x = F(x)$ for all $x \in \mathrm{dom}(F)$. Let $I \subseteq \mathbb{N}^{<\mathbb{N}} \times \mathbb{N}^{<\mathbb{N}}$ be the relation defined by

$$I(\sigma,\tau) \iff (\exists s \in \mathbb{N})(\forall m < l(\sigma))[\Phi_e^{\hat{\tau}}(m) = \sigma(m)],$$

where $\hat{\tau} \in 2^{<\mathbb{N}}$ denotes the computable coding described just before definition 1.1.28. One can see that theorem 1.1.24 implies that I is c.e. Let $V_\sigma = \bigcup\{U_\tau : (\sigma,\tau) \in I\}$ for each $\sigma \in \mathbb{N}^{<\mathbb{N}}$, then by definition $\{V_\sigma\}_{\sigma\in\mathbb{N}^{<\mathbb{N}}}$ are uniformly open.

Let $\sigma \in \mathbb{N}^{<\mathbb{N}}$, we check that $F^{-1}(U_\sigma) = \mathrm{dom}(F) \cap V_\sigma$. If $x \in \mathrm{dom}(F)$ is such that $F(x) \succ \sigma$, then by the use principle (theorem 1.1.25), there exists s and $\tau \prec x$ such that $\Phi_{e,s}^{\hat{\tau}}(m) = \sigma(m)$ for all $m < l(\sigma)$, therefore, $I(\sigma,\tau)$ is satisfied and $x \in U_\tau$. Conversely, if $x \in \mathrm{dom}(F) \cap V_\sigma$, then $x \in U_\tau$ for some $\tau \in \mathbb{N}^{<\mathbb{N}}$ which satisfies $I(\sigma,\tau)$. This means that there is a s such that $\Phi_{e,s}^{\hat{\tau}}(m) = \sigma(m)$ for all $m < l(\sigma)$. By the use principle, this implies that $\Phi_e^x(m) = \sigma(m)$ for all $m < l(\sigma)$, that is, $F(x) \succ \sigma$.

Conversely, suppose F is effectively continuous. Let $I \subseteq \mathbb{N}^{<\mathbb{N}} \times \mathbb{N}^{<\mathbb{N}}$ be the c.e. relation such that for all $\sigma \in \mathbb{N}^{<\mathbb{N}}$, $V_\sigma = \bigcup\{U_\tau : (\sigma,\tau) \in I\}$ and $F^{-1}(U_\sigma) = \mathrm{dom}(F) \cap V_\sigma$. By the listing theorem 1.1.18, there is a computable $f\colon \mathbb{N} \to \mathbb{N}^{<\mathbb{N}} \times \mathbb{N}^{<\mathbb{N}}$ with $\mathrm{range}(f) = I$. Given $x \in \mathbb{N}^{\mathbb{N}}$ and $m \in \mathbb{N}$, we compute $F(x)(m)$ in the following way: Keep computing $f(s)$ for each $s = 0, 1, 2, \ldots$ until we find a pair (σ,τ) such that σ is a string of length at least $m+1$ and $\tau \prec x$. This means that $\sigma \prec F(x)$, so output $\sigma(m)$. This procedure always terminates when $x \in \mathrm{dom}(F)$. $\square$

Definition 1.1.34. Let $A, B \subseteq \mathbb{N}^{\mathbb{N}}$. A functional $F\colon A \to B$ is called a *computable homeomorphism* if it is computable and has a computable inverse $F^{-1}\colon B \to A$.

Example 1.1.35. Let $\varphi\colon \mathbb{N} \to \mathbb{N}$ be a computable bijection, its inverse is easily seen to be computable. Define $F\colon \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ where $F(x)(n) = x(\varphi(n))$ for all $n \in \mathbb{N}$. This is clearly computable and has the computable inverse given by $F^{-1}(x)(n) = x(\varphi^{-1}(n))$. Therefore, F is a computable homeomorphism.

Example 1.1.36. For another example, let $A \subset \mathbb{N}$ be a finite set of naturals, then $A^{\mathbb{N}}$ is computably homeomorphic to $2^{\mathbb{N}}$. Indeed, we may assume that $A = \{0, 1, \ldots, n-1\}$. Define $f\colon A \to 2^{<\mathbb{N}}$ the following substitution

$$f(m) = \begin{cases} 0^m 1 & \text{if } m < n-1, \\ 0^m & \text{if } m = n-1, \end{cases}$$

where 0^m means the string of m consecutive 0's. Then, define $F : A^{\mathbb{N}} \to 2^{\mathbb{N}}$ by

$$F(x) = f(x(0))f(x(1))f(x(2))\ldots$$

for all $x \in A^{\mathbb{N}}$. We are just replacing each component $x(i)$ of x by the corresponding word according to f. Since f is a finite function, it is easy to see that F is computable. Its inverse is also computable: Given $x \in 2^{\mathbb{N}}$, $F^{-1}(x)(0)$ is the number of zeros before finding a 1, maximum $n-1$, continue in this way for $F^{-1}(x)(1)$ and so on.

1.2 Geometric group theory

Geometric group theory studies groups under the lenses of geometry. It does so by associating geometric objects to groups and deducing algebraic data from them. The fundamental example of this is the Cayley graph of a finitely generated group, on which we can use the theory of metric spaces. However, we will quickly see that the exact geometry of the graph is not that useful and it is best to study the large scale geometry of it, i.e., the geometry preserved by quasi-isometries. After introducing quasi-isometries we will present the fundamental Schwarz-Milnor lemma. One may consult [21] for more information.

1.2.1 Group presentations and the word problem

We start with some basic definitions from combinatorial group theory, a reference for this topic is [22]. Afterwards we relate this concepts to computability and formalize the word problem.

Let S be a set and let $S^{-1} = \{s^{-1} : s \in S\}$ be a disjoint copy of it. Take the set of words over elements of $S \cup S^{-1}$

$$W(S) = (S \cup S^{-1})^{<\mathbb{N}} = \{s_1^{\varepsilon_1} s_2^{\varepsilon_2} \cdots s_n^{\varepsilon_n} : s_i \in S,\, \varepsilon_i \in \{+1, -1\}, \text{for each } 1 \leq i \leq n\},$$

and equip it with the operation of concatenation. This operation is associative and the empty word ε is the identity element. Define an equivalence relation $\sim$ over $W(S)$ as follows: $w \sim v$ iff we can reach v from w by adding or removing pairs of symbols of the form ss^{-1} or $s^{-1}s$ with $s \in S$. For example, if $S = \{a, b\}$ then $abb \sim abaa^{-1}bb^{-1}b$. We denote by $[w]$ the equivalence class of w under $\sim$.

Definition 1.2.1. Let S be a set. The *free group $F(S)$ with basis S* is the group $W(S)/\sim$ equipped with the operation $[w] \cdot [v] = [wv]$ for each $w, v \in W(S)$.

A free group is determined up to isomorphism by the cardinality of its basis, so we may speak of F_n the free group with n generators.

Definition 1.2.2. Let G be a group, let S be a set and let $R \subset F(S)$. We say that the pair $(S \mid R)$ is a *presentation* for G if $G \cong F(S)/N$ where N is the normal closure of R in $F(S)$, i.e., the smallest normal subgroup of $F(S)$ containing R.

We denote $G \cong \langle S \mid R \rangle$. The image of S in G is a generating set, so we call the elements of S the *generators* of G. We will most of the time regard S as a subset of G itself. The elements of R end up at the identity of G, so they define *relations* in G.

Example 1.2.3.

- The cyclic group of order n has the presentation $\mathbb{Z}/n\mathbb{Z} \cong \langle a \mid a^n \rangle$.
- The free group of n generators has the presentation $F_n \cong \langle a_0, a_1, \ldots, a_{n-1} \mid \rangle$.
- We have $\mathbb{Z} \times \mathbb{Z} \cong \langle x, y \mid xyx^{-1}y^{-1} \rangle$.
- The dihedral group of order $2n$ is the group $D_n = \langle r, s \mid r^n, s^2, srs^{-1} = r^{-1} \rangle$.

A presentation $\langle S \mid R \rangle$ is *finitely generated* if the set S is finite. If additionally R is finite, we say that the presentation is *finite*. A group is *finitely generated* if it has a finitely generated presentation, and is *finitely presented* if it has a finite presentation.

Group presentations are a very compact way of defining groups, however it is not simple to extract information from them. An apparently simple question is the *word problem*, which asks for an algorithm for deciding when two words $w, v \in F(S)$ are equal in G, or equivalently, and algorithm for deciding when a word $w \in F(S)$ is equal to the identity in G. We now formalize the word problem for the case of finitely generated groups.

Let G be a group generated by the finite set $S = \{s_0, \ldots, s_{n-1}\}$. We will construct a coding (as in definition 1.1.3) for G. First, we assign a number to each s_i and s_i^{-1}: $0 \mapsto s_0$, $1 \mapsto s_0^{-1}$, $2 \mapsto s_1$, etc. More precisely, let $\mu_S \colon \mathbb{N} \to G$ be such that

$$\mu_S(i) = \begin{cases} s_{i/2} & \text{if } i \text{ is even and } < 2n, \\ s_{(i-1)/2}^{-1} & \text{if } i \text{ is odd and } < 2n, \\ e & \text{otherwise.} \end{cases}$$

Then, for each $u \in \mathbb{N}$, let $\sigma \in \mathbb{N}^{<\mathbb{N}}$ be such that $\langle \sigma \rangle = u$ and define the coding $\nu_S : \mathbb{N} \to G$ as

$$\nu_S(u) = \mu_S(\sigma(0)) \cdot \mu_S(\sigma(1)) \cdots \mu_S(\sigma(l(\sigma) - 1)).$$

Essentially, we are just representing each element of G as a word over S. When we do computations we are just manipulating words over S without really knowing the elements they represent.

Proposition 1.2.4. *The group operation $(g,h) \mapsto gh$ is a $(\nu_S \times \nu_S, \nu_S)$-computable function. Taking inverses $g \mapsto g^{-1}$ is a (ν_S, ν_S)-computable function.*

Proof. Let $g, h \in G$ and $n, m \in \mathbb{N}$ be such that $\nu_S(n) = g$ and $\nu_S(m) = h$. We have to compute u such that $\nu_S(u) = gh$, this is straightforward from the construction of ν_S: Retrieve the strings $\sigma, \tau \in \mathbb{N}^{<\mathbb{N}}$ that are coded by n and m respectively and compute their concatenation $\sigma\tau$, output $u = \langle \sigma\tau \rangle$.

For the inverse, let n and g be such that $\nu_S(n) = g$. Then $g^{-1} = \mu_S(\sigma(l(\sigma)-1))^{-1} \cdots \mu_S(\sigma(0))^{-1}$, where σ is the string in $\mathbb{N}^{<\mathbb{N}}$ coded by n. So, to compute a code for g^{-1}, invert σ and change each component so that they code their respective inverses according to the definition of μ_S, more concretely let τ be the string of length $l(\sigma)$ defined by:

$$\tau(i) = \begin{cases} \sigma(l(\sigma)-1-i)+1 & \text{if } \sigma(l(\sigma)-1-i) \text{ is even,} \\ \sigma(l(\sigma)-1-i)-1 & \text{if } \sigma(l(\sigma)-1-i) \text{ is odd,} \end{cases}$$

for each $i < l(\sigma)$. Output $u = \langle \tau \rangle$. □

We have defined ν_S by fixing a finite generating set S. The following proposition shows that this choice is not important.

Proposition 1.2.5. *Let G be a finitely generated group and let S and T be two finite generating sets. The identity* $\mathrm{id}\colon G \to G$ *is (ν_S, ν_T)-computable.*

Proof. The idea is that since S is finite, we may computably rewrite any word over S as a word over T. More formally, let $S = \{s_0, \ldots, s_{n-1}\}$, then for each $i < 2n$ there is a string $\tau_i \in \mathbb{N}^{<\mathbb{N}}$ such that $\mu_S(i) = \nu_T(\langle \tau_i \rangle)$, the following function $f\colon \mathbb{N} \to \mathbb{N}^{<\mathbb{N}}$ is trivially computable,

$$f(i) = \begin{cases} \tau_i & \text{if } i < 2n, \\ \varepsilon & \text{otherwise.} \end{cases}$$

Let $g \in G$ and n be a ν_S-code for g, we compute a ν_T-code from n in the following way: Take σ the string coded by n and replace each $\sigma(i)$ by the string $f(\sigma(i))$. We end up with the concatenation

$$\tau = f(\sigma(0)) f(\sigma(1)) \cdots f(\sigma(l(\sigma)-1))$$

so output $\langle \tau \rangle$. By the construction of ν_T and f, we have

$$\nu_T(\langle \tau \rangle) = \nu_T(\langle f(\sigma(0)) \rangle) \cdots \nu_T(\langle f(\sigma(l(\sigma)-1)) \rangle) = \mu_S(\sigma(0)) \cdots \mu_S(\sigma(l(\sigma-1))) = \nu_S(\langle \sigma \rangle) = g,$$

thus $\langle \tau \rangle$ is indeed a ν_T-code for g. □

By switching S and T in the previous proposition, we have that the identity is (ν_T, ν_S)-computable. Intuitively this means we may computably translate a code under S into a code under T and viceversa. We say that ν_S and ν_T are *equivalent codings.* From this we get the following corollary.

Corollary 1.2.6. *Let G be a finitely generated group and let S and T be two finite generating sets. Then, the set $\{n \in \mathbb{N} : \nu_S(n) = e\}$ is computable if and only if $\{n \in \mathbb{N} : \nu_T(n) = e\}$ is computable.*

Proof. Suppose $\{n \in \mathbb{N} : \nu_T(n) = e\}$ is computable. By the previous proposition, there exists a computable $f : \mathbb{N} \to \mathbb{N}$ such that $\nu_T(f(n)) = \nu_S(n)$ for all n. Then, we can check $\nu_S(n) = e$ by computing $f(n)$ and checking if $\nu_T(f(n)) = e$. The reciprocal is exactly the same. □

Definition 1.2.7. A finitely generated group G has *solvable word problem* if for some (and hence every) finite generating set S of G, the set $\{n \in \mathbb{N} : \nu_S(n) = e\}$ is computable.

Theorem 1.2.8. *Let G be an infinite finitely generated group with solvable word problem. There exists a bijective coding $\nu\colon \mathbb{N} \to G$ that makes the group operation computable. Moreover this coding is unique in the following sense: if $\mu\colon \mathbb{N} \to G$ is another bijective coding that makes the group operation computable, then μ is equivalent to ν, i.e., $\nu^{-1} \circ \mu$ and $\mu^{-1} \circ \nu$ are computable.*

Proof. Let S be a finite generating set of G and let ν_S be the associated coding. Define an equivalence relation on $\mathbb{N}$ by $n \sim m$ iff $\nu_S(n) = \nu_S(m)$. Since the word problem is solvable and the group operations are computable, we may computably check if $n \sim m$ in the following way: Compute u such that $\nu_S(u) = \nu_S(n) \cdot \nu_S(m)^{-1}$ and check if $\nu_S(u) = e$.

Now, choose an element from each class by taking the minimum of each, that is, let $D = \{n \in \mathbb{N} : n = \min[n]\}$, where $[n]$ denotes the equivalence class of n. Then, D is computable: Given n, check if $n \sim m$ for each $m < n$. Moreover, ν_S restricted to D is a bijection with G which is infinite by hypothesis. Thus, D is an infinite c.e. set, so by corollary 1.1.19 there exists a computable bijection $f : \mathbb{N} \to D$. Define the bijective coding $\nu = \nu_S \circ f$, then the group operation is computable: Given n and m, we can compute u such that $\nu_S(u) = \nu_S(f(n)) \cdot \nu_S(f(m))$, then, find s with $f(s) \sim u$ and output it. We have $\nu(s) = \nu(n) \cdot \nu(m)$.

For the uniqueness, let $\mu \colon \mathbb{N} \to G$ be another bijective coding that makes the group operation computable. Then, there is a computable function $F : \mathbb{N}^2 \to \mathbb{N}$ such that $\mu(F(n, m)) = \mu(n) \cdot \mu(m)$ for all n, m. In this way, the set $\mathbb{N}$ with the operation F becomes a group isomorphic to G. Let $H : \mathbb{N}^2 \to \mathbb{N}$ be the computable function that $\nu(H(n, m)) = \nu(n) \cdot \nu(m)$ for all n, m, similarly, $\mathbb{N}$ with H become a group isomorphic to G. Let $K = \mu^{-1}(S \cup S^{-1})$ be the μ-codes of the finite set $(S \cup S^{-1})$. By construction, $\nu^{-1} \circ \mu$ is an isomorphism from $(\mathbb{N}, F)$ to $(\mathbb{N}, H)$ and so it is determined by its values on the finite generating set K, let $f : K \to \mathbb{N}$ be the finite function defined by $f(k) = (\nu^{-1} \circ \mu)(k)$. We compute $\nu^{-1} \circ \mu$ in the following way: Given $m \in \mathbb{N}$, computably enumerate all the strings $\sigma \in K^{<\mathbb{N}}$ and find one that $F(\sigma(0), \sigma(1), \ldots, \sigma(l(\sigma) - 1)) = m$, this must happen since S generates G. Then compute and output $H(f(\sigma(0)), \ldots, f(\sigma(l(\sigma) - 1)))$. The computability of $\mu^{-1} \circ \nu$ is analogous. □

Corollary 1.2.9. *Let G and H be infinite finitely generated groups with solvable word problem and let $\varphi \colon G \to H$ be a group isomorphism. Then φ is computable (with respect to any bijective codings that make the operations computable).*

Proof. Let $\nu \colon \mathbb{N} \to G$ and $\mu \colon \mathbb{N} \to H$ be bijective codings that make the respective operations computable. Then, $\varphi \circ \nu$ is a bijective coding for H that makes its operation computable, so by the previous theorem $\mu^{-1} \circ \varphi \circ \nu$ is computable. That is to say, φ is computable. □

The previous results show that finitely generated groups with solvable word problem have a unique computable structure compatible with their algebraic structure. This property is known as *computable categoricity*. In the non-finitely generated context it turns out that there are groups with multiple computability structures. This kind of properties are studied by the much more general theory of computable structures, a very interesting subject that is beyond the scope of this work. A nice introduction is the forthcoming [9], other references are [24] and [1].

It turns out that the word problem, though simple in appearance, is unsolvable even in the finitely presented case. This means that there are finitely presented groups that do not have a compatible computable structure.

Theorem 1.2.10 (Boone-Novikov)**.** *There exists a finitely presented group with unsolvable word problem.*

The proof is a classic reduction from the halting problem, however the details are very complicated. One may get this result as a simple corollary from the following (also difficult to prove) theorem.

A group G is *recursively presented* if it has a presentation $G \cong \langle S \mid R\rangle$, where S is finite and R is a computably enumerable subset of $F(S)$ (under some effective coding of $F(S)$).

Theorem 1.2.11 (Higman's embedding theorem)**.** *Any recursively presented group G can be embedded in some finitely presented group.*

Proof of Theorem 1.2.10. Let K be a computably enumerable set which is not computable, as in theorem 1.1.9. Let

$$G = \langle a, b, c, d \mid a^{-k}ba^k = c^{-k}dc^k, k \in K\rangle,$$

then clearly G is recursively presented. Moreover, it is not hard to check that

$$a^{-n}ba^n = c^{-n}dc^n \iff n \in K,$$

so one could compute K if the word problem were to be solvable. Thus, G has unsolvable word problem. By Higman's theorem, G embeds in a finitely presented group H. Since G is finitely generated with unsolvable word problem, H must have unsolvable word problem. □

A proof of Higman's theorem may be found in [22, Chapter IV]. A direct proof of the Boone-Novikov theorem may be found in [25].

1.2.2 Graphs and quasi-isometries

We turn to the geometry of groups. First, some definitions from graph theory.

Definition 1.2.12. A (simple, undirected) *graph* is a pair $X = (V, E)$ where E is a relation on V such that

1. E is irreflexive: $(v, v) \notin E \quad \forall v \in V$,
2. E is symmetric: $(v, v') \in E \implies (v', v) \in E \quad \forall v, v' \in V$.

The elements of V are the *vertices* and the elements of E are the *edges*. We say that two vertices v, v' are adjacent when $(v, v') \in E$.

Definition 1.2.13. Let $X = (V, E)$ be a graph.

1. Let $n \in \mathbb{N} \cup \{\infty\}$. A *path* in X of length n is a sequence $v_0, \dots, v_n \in V$ of different vertices such that $(v_j, v_{j+1}) \in E$ for all $j \in \{0, \dots, n-1\}$; if $n < \infty$ then we say that this path connects the vertices v_0 and v_n.
2. The graph X is called *connected* if every pair of its vertices can be connected by a path.
3. Let $n \in \mathbb{N}$ with $n > 2$. A *cycle* in X of length n is a path $v_0, \dots, v_{n-1}$ in X with $(v_{n-1}, v_0) \in E$.
4. A *tree* is a connected graph with no cycles.

Example 1.2.14 (Binary tree)**.** Let $V = 2^{<\mathbb{N}}$ and put an edge between two words $\sigma, \tau \in V$ if and only if $\tau = \sigma a$ for some $a \in 0, 1$. The resulting graph is depicted in figure 3. It is easy to see that this graph is a tree. Moreover, if $\sigma_0, \sigma_1, \dots \in V$ is an infinite path, then there exists $k \in \mathbb{N}$ such that, $\sigma_i \preceq \sigma_j$ for all $k \leq i \leq j$, and $l(\sigma_n) \to \infty$ as n goes to infinity. Thus, there exists $x \in 2^{\mathbb{N}}$ with $\sigma_i \prec x$ for all $i \geq k$. Therefore, every infinite path in $2^{<\mathbb{N}}$ corresponds to an element of the Cantor space $2^{\mathbb{N}}$.

Example 1.2.15 (Infinite branching tree)**.** Similarly, let $V = \mathbb{N}^{<\mathbb{N}}$ and put an edge between two words $\sigma, \tau \in V$ if and only if $\tau = \sigma a$ for some $a \in \mathbb{N}$. Every infinite path in $\mathbb{N}^{<\mathbb{N}}$ corresponds to an element of the Baire space $\mathbb{N}^{\mathbb{N}}$.

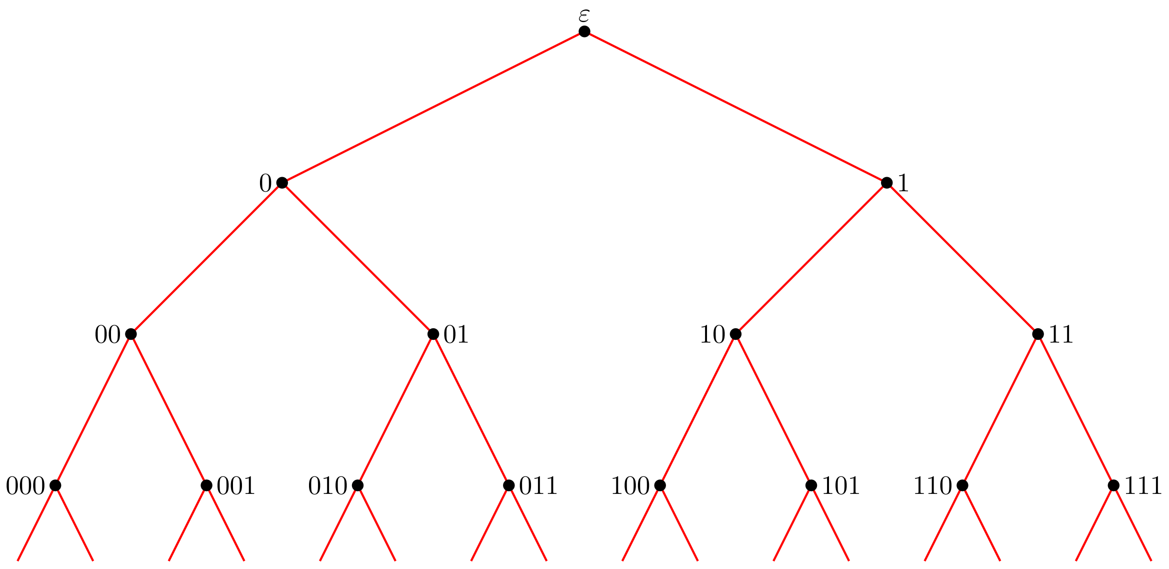

Figure 3: The binary tree $2^{<\mathbb{N}}$

Connected graphs have a natural metric space structure.

Definition 1.2.16. Let $X = (V, E)$ be a connected graph. We define the *metric on V associated with the graph X* as the map $d\colon V \times V \to [0, \infty)$ with

$$d(v, w) = \min\{n \in \mathbb{N} : \text{there is a path of length } n \text{ connecting } v \text{ and } w\}.$$

We now describe the fundamental construction in geometric group theory.

Definition 1.2.17. Let G be a group and let $S \subseteq G$ be a subset of G. The *Cayley graph* of G with respect to S is the graph $\mathrm{Cay}(G, S)$ whose

- set of vertices is G, and whose
- set of edges is $\{(g, g \cdot s) : g \in G, s \in (S \cup S^{-1}) \setminus \{e\}\}$.

So, two elements of G are adjacent when you can reach one from the other by a single multiplication of (the inverse of) an element of the set S. If we furthermore assume that S is a generating set of G, it easy to see that the resulting graph is connected. As an example, figure 4 depicts the Cayley graphs of the groups $\mathbb{Z}/5\mathbb{Z}$ and D_3 described in example 1.2.3 with respect to the generating sets $\{[1]\}$ and $\{r, s\}$ respectively. Figure 5 shows the Cayley graph of F_2 with respect to a free generating set $\{a, b\}$.

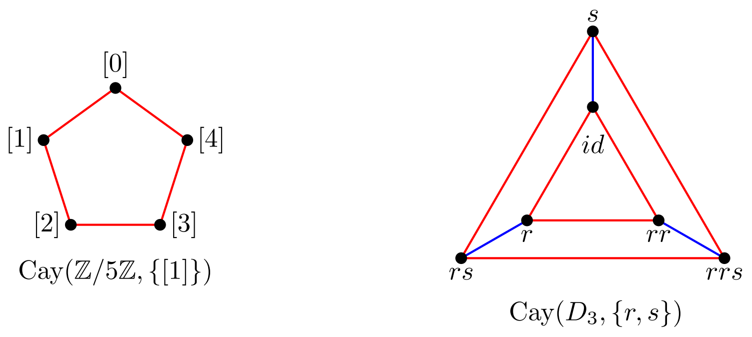

Figure 4: Examples of Cayley graphs

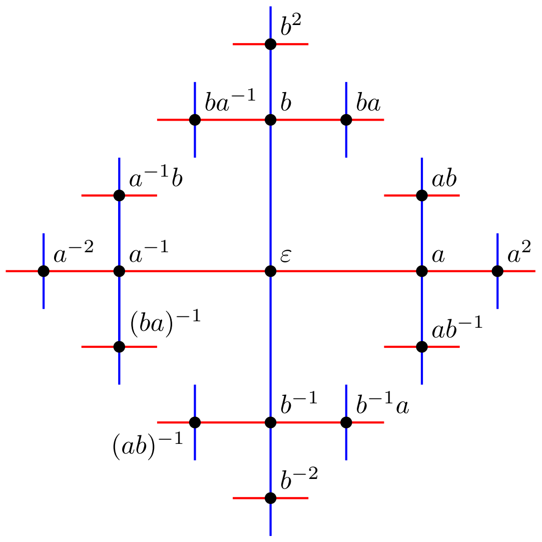

Figure 5: Cayley graph of the free group F_2 generated by $\{a, b\}$

Definition 1.2.18. Let G be a group and $S \subseteq G$ be a generating set. The *word metric d_S* on G with respect to S is the metric on G associated with the Cayley graph $\mathrm{Cay}(G, S)$. In other words,

$$d_S(g, h) = \min\{n \in \mathbb{N} : (\exists s_1, \dots, s_n \in S \cup S^{-1})\,[h = g \cdot s_1 \cdots s_n]\}$$

for all $g, h \in G$.

As we have seen with the examples above, this metric depends on the generating set, that is, different generating sets yield non isometric metric spaces. We would like to study the geometry of the group itself independent from a particular generating set, so we turn to quasi-isometries.

Definition 1.2.19. Let $f\colon X \to Y$ be a function between metric spaces (X, d_X) and (Y, d_Y).

- The function f is a *quasi-isometric embedding* if there are constants $a, b \in (0, \infty)$ such that

$$\forall x, x' \in X \quad \frac{1}{a} \cdot d_X(x, x') - b \leq d_Y(f(x), f(x')) \leq a \cdot d_X(x, x') + b.$$

  We may also say that f is a (a, b)-*quasi-isometric embedding.*

- The function f has *quasi-dense image* if there is a constant $c \in [0, \infty)$ such that

$$\forall y \in Y \exists x \in X \quad d_Y(f(x), y) \leq c.$$

- The function f is a *quasi-isometry* if it is a quasi-isometric embedding with quasi-dense image.

Intuitively, a quasi-isometry forgets about the local information but preserves the geometry seen from far away. The canonical example is the inclusion $\iota\colon \mathbb{Z} \to \mathbb{R}$ where $\mathbb{Z}$ and $\mathbb{R}$ have their standard euclidean metrics. ι is a $(1, 0)$-quasi-isometric embedding, also known as an isometric embedding, and has a quasi-dense image since every real number is at most $1/2$ from an integer. Let $h\colon \mathbb{R} \to \mathbb{Z}$ be the floor function, i.e.,

$$h(x) = \max\{k \in \mathbb{Z} : k \leq x\} \quad \forall x \in \mathbb{R}.$$

It is not hard to see that h is a $(0, 1)$-quasi-isometric embedding and surjective. Roughly speaking, we are "compressing" the bounded sets $[k, k+1)$, $k \in \mathbb{Z}$ into points, which is the main idea of coarse geometry, in some sense we identify things by "boundedness". Another example of this is the following definition.

Definition 1.2.20. Two functions $f, f' : X \to Y$ between metric spaces (X, d_X) and (Y, d_Y) are said to be at *finite distance* from each other if

$$\sup_{x \in X} d_Y(f(x), f'(x)) < \infty.$$

A function $g : Y \to X$ is a *quasi-inverse* of f is $g \circ f$ is at finite distance from the identity in X and $f \circ g$ is at finite distance from the identity in Y.

Coming back to our example above, $h \circ \iota$ is the identity in $\mathbb{Z}$ and $(\iota \circ h)(x)$ is at most 1 from x for every $x \in \mathbb{R}$, so h is a quasi-inverse of ι. More generally, every quasi-isometry has a quasi-inverse.

Proposition 1.2.21. *A function $f\colon X \to Y$ between metric spaces (X, d_X) and (Y, d_Y) is a quasi-isometry if and only if it is a quasi-isometric embedding and there exists a quasi-isometric embedding $g\colon Y \to X$ which is a quasi-inverse of f.*

Proof. Let $f\colon X \to Y$ be a quasi-isometry, then there exists a constant $c > 0$ such that

$$(\forall x, x' \in X) \quad \frac{1}{c} d_X(x, x') - c \leq d_Y(f(x), f(x')) \leq c d_X(x, x') + c,$$

$$(\forall y \in Y)(\exists x \in X)\, d_Y(f(x), y) \leq c.$$

For each $y \in Y$, choose $g(y) \in X$ such that $d_Y(f(g(y)), y) \leq c$. This defines a function $g\colon Y \to X$. We have for all $y \in Y$,

$$d_Y(f \circ g(y), y) \leq c,$$

and for all $x \in X$,

$$d_X(g(f(x)), x) \leq c d_Y(f(g(f(x))), f(x)) + c^2 \leq c^2 + c^2 = 2c^2.$$

So $f \circ g$ and $g \circ f$ have finite distance from the respective identity functions, therefore g is a quasi-inverse of f.

Moreover, g is a quasi-isometric embedding: Let $y, y' \in Y$, then

$$\begin{aligned} d_X(g(y), g(y')) &\leq c d_Y(f(g(y)), f(g(y'))) + c^2 \\ &\leq c\,(d_Y(f(g(y)), y) + d_Y(y, y') + d_Y(f(g(y')), y')) + c^2 \\ &\leq c\,(c + d_Y(y, y') + c) + c^2 \\ &= c d_Y(y, y') + 3c^3, \end{aligned}$$

and

$$\begin{aligned} d_X(g(y), g(y')) &\geq \frac{1}{c} d_Y(f(g(y)), f(g(y'))) - 1 \\ &\geq \frac{1}{c}\,(d_Y(y, y') - d_Y(f(g(y)), y) - d_Y(f(g(y')), y')) - 1 \\ &\geq \frac{1}{c} d_Y(y, y') - \frac{2c}{c} - 1. \end{aligned}$$

Conversely, let $f\colon X \to Y$ be a quasi-isometric embedding and $g\colon X \to Y$ be a quasi-inverse quasi-isometric embedding. Then, there exists a constant $c > 0$ such that

$$d_Y(f(g(y)), y) \leq c \qquad \text{for all } y \in Y,$$

so f has quasi-dense image. Therefore f is a quasi-isometry. □

Definition 1.2.22. Let (X, d_X) and (Y, d_Y) be metric spaces. We say that X is *quasi-isometric* to Y if there is a quasi-isometry from X to Y. We write $X \sim_{QI} Y$.

The previous proposition shows that $\sim_{QI}$ is a symmetric relation. A simple calculation shows that the composition of two quasi-isometries is itself a quasi-isometry, so $\sim_{QI}$ is transitive. Finally, the identity is a quasi-isometry. Therefore, $\sim_{QI}$ is an equivalence relation in the class of metric spaces. We now come back to groups.

Proposition 1.2.23. *Let G be a finitely generated group, and let S and S' be finite generating sets of G. Then the identity map id_G is a quasi-isometry between (G, d_S) and $(G, d_{S'})$.*

Proof. Let $t = \max\{d_{S'}(e, s) : s \in S\}$ and $g, h \in G$ with $d_S(g, h) = n$, so that $h = g \cdot s_1 \cdots s_n$ for some $s_1, \ldots, s_n \in S$. We have by the triangle inequality and the G-invariance of $d_{S'}$

$$\begin{aligned} d_{S'}(g, h) &\leq d_{S'}(g, g \cdot s_1) + d_{S'}(g \cdot s_1, g \cdot s_1 \cdot s_2) + \cdots + d_{S'}(g \cdot s_1 \cdots s_{n-1}, g \cdot s_1 \cdots s_n) \\ &= d_{S'}(e, s_1) + d_{S'}(e, s_2) + \cdots + d_{S'}(e, s_n) \\ &\leq t \cdot n = t \cdot d_S(g, h). \end{aligned}$$

Let $t' = \max\{d_S(e, s') : s' \in S'\}$, by the same calculation we have

$$d_S(g, h) \leq t' \cdot d_{S'}(g, h) \quad \text{for all } g, h \in G.$$

Therefore, the identity is a $(c, 0)$-quasi-isometry, where $c = \max\{t, t'\}$. □

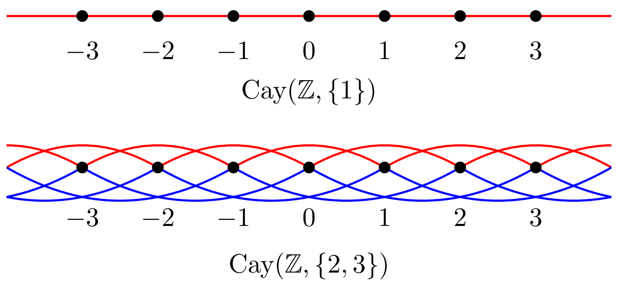

Figure 6: Quasi-isometric Cayley graphs of ℤ

In fact, from the proof it can be seen that id_G is a *Bilipschitz equivalence*, that is to say, a bijective $(c, 0)$-quasi-isometry for some $c \geq 1$, which is strictly stronger than mere quasi-isometry, see [10]. So, changing the generating set does not alter the large scale geometry of the group, see figure 6 for an example. This means we may speak of the quasi-isometry type of a finitely generated group.

Definition 1.2.24. Let G be a finitely generated group. The group G is *quasi-isometric* to a metric space X if for some (and hence every) finite generating set S of G the metric spaces (G, d_S) and X are quasi-isometric. We write $G \sim_{QI} X$.

We have seen that $(\mathbb{Z}, d_S)$ with $S = \{1\}$ is quasi-isometric to $\mathbb{R}$ with the euclidean metric by the inclusion, so we now say that $\mathbb{Z}$ *as a group* is quasi-isometric to $\mathbb{R}$. More generally, $\mathbb{Z}^n \sim_{QI} \mathbb{R}^n$ for all $n \in \mathbb{N}$. Another example of quasi-isometric groups are all finite groups. Since they have finite diameter they are all quasi-isometric to a point. Conversely, infinite groups have infinite diameter with respect to any finite generating set, so they cannot be quasi-isometric to a point. We say that being finite is a geometrical property or a quasi-isometry invariant of groups.

We end this section by stating the Schwarz-Milnor lemma, which is often called the fundamental lemma of geometric group theory since it links the geometry of groups to the geometry of natural spaces that arise in geometry and topology. It is a big source of examples of groups quasi-isometric to nice metric spaces. First, some preliminary definitions.

Definition 1.2.25. Let (X, d) be a metric space and G a group.

- Let $L \in [0, \infty)$. A *geodesic of length L* in X is an isometric embedding $\gamma\colon [0, L] \to X$, where the interval $[0, L]$ carries the standard metric from $\mathbb{R}$. We say that γ connects the points $\gamma(0)$ and $\gamma(L)$.
- The metric space X is called *geodesic* if for all $x, x' \in X$ there exists a geodesic in X that connects them.
- X is *proper* if for all $x \in X$ and all $r > 0$ the closed ball $\{y \in X : d(x, y) \leq r\}$ is compact.
- An action $G \times X \to X$ is *proper* if for all compact sets $B \subset X$ the set $\{g \in G : g \cdot B \cap B \neq \emptyset\}$ is finite.
- An action $G \times X \to X$ is *cocompact* if the quotient space of orbits $G \backslash X$ is compact.

Theorem 1.2.26 (Schwarz-Milnor lemma)**.** *Let G be a group acting by isometries on a proper and geodesic metric space (X, d). Furthermore, suppose that this action is proper and cocompact. Then G is finitely generated, and for all $x \in X$ the map*

$$\begin{aligned} G &\longrightarrow X \\ g &\longmapsto g \cdot x \end{aligned}$$

is a quasi-isometry.

Proof. A proof may be found in [21, Corollary 5.4.2]. □

1.3 Fuchsian groups

In this section we briefly introduce Fuchsian groups and give an application of the Schwarz-Milnor lemma that we will effectivize in section 2.3. We omit most proofs. The reader may consult [16] and chapter 9 of [11] for more details.

Definition 1.3.1. The *hyperbolic plane* $\mathbb{H}$ is the open upper half-plane

$$\mathbb{H} = \{z \in \mathbb{C} : \mathrm{Im}(z) > 0\}$$

endowed with the smooth structure inherited from $\mathbb{C}$ (as a real manifold) and the Riemannian metric

$$ds = \frac{\sqrt{dx^2 + dy^2}}{y}.$$

A Riemannian metric is essentially just a way to define the length of curves. Let $\gamma\colon [a, b] \to \mathbb{H}$ be a smooth curve and denote $\gamma(t) = x(t) + iy(t)$ for all $t \in [a, b]$. The *hyperbolic length* of γ is defined by

$$L(\gamma) = \int_\gamma ds = \int_a^b \frac{\sqrt{x'(t)^2 + y'(t)^2}}{y(t)}\, dt.$$

We define $d\colon \mathbb{H} \times \mathbb{H} \to [0, \infty)$ by

$$d(z, z') = \inf\{L(\gamma) : \gamma \text{ is a smooth curve from } z \text{ to } z'\},$$

for all $z, z' \in \mathbb{H}$.

Proposition 1.3.2. *The function d is a metric and it induces the subspace topology of $\mathbb{H} \subset \mathbb{C}$. In consequence, $(\mathbb{H}, d)$ is a proper metric space.*

Theorem 1.3.3. *Let $z, z' \in \mathbb{H}$ with $z \neq z'$. There exists precisely one geodesic in $(\mathbb{H}, d)$ from z to z'. In particular, the metric space $(\mathbb{H}, d)$ is geodesic.*

Thus, $(\mathbb{H}, d)$ satisfies the hypothesis of the Schwarz-Milnor lemma. This next theorem shows an explicit formula for computing d, so intuitively speaking, d is a computable function.

Theorem 1.3.4. *For all $z, z' \in \mathbb{H}$ we have*

$$d(z, z') = \operatorname{arccosh}\left(1 + \frac{|z - z'|^2}{2 \cdot \operatorname{Im} z \cdot \operatorname{Im} z'}\right),$$

where $\operatorname{arccosh}(x) = \ln(x + \sqrt{x^2 - 1})$ *for all $x \in \mathbb{R}$ with $x \geq 1$.*

We now describe the isometries of $\mathbb{H}$. Denote by $\mathrm{SL}_2(\mathbb{R})$ the special linear group of 2×2 matrices with determinant 1. Let $\mathrm{PSL}_2(\mathbb{R})$ be the quotient group $\mathrm{SL}_2(\mathbb{R})/\{\pm I_2\}$, where I_2 is the 2×2 identity matrix.

Definition 1.3.5. For

$$A = \pm \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{PSL}_2(\mathbb{R})$$

the associated Möbius transformation is the function

$$\begin{aligned} g_A \colon \mathbb{H} &\longrightarrow \mathbb{H} \\ z &\longmapsto \frac{az + b}{cz + d}. \end{aligned}$$

Theorem 1.3.6. *Every Möbius transformation is an orientation-preserving isometry. Moreover, the map*

$$\begin{aligned} \mathrm{PSL}_2(\mathbb{R}) &\longrightarrow \mathrm{Isom}^+(H, d) \\ A &\longmapsto g_A \end{aligned}$$

is a group isomorphism, where $\mathrm{Isom}^+(\mathbb{H}, d)$ *denotes the group of all orientation preserving isometries of* $(\mathbb{H}, d)$.

In particular, $\mathrm{PSL}_2(\mathbb{R})$ acts by isometries on $\mathbb{H}$. A subgroup $\Gamma \leq \mathrm{PSL}_2(\mathbb{R})$ is said to be cocompact if its action on $\mathbb{H}$ is cocompact, i.e., the quotient space $\Gamma\backslash\mathbb{H}$ is compact.

The group $\mathrm{PSL}_2(\mathbb{R})$ has a natural smooth structure coming from $\mathbb{R}$ that makes it a connected locally compact Lie group (see [18, Chapter 7 and 21] for an introduction to Lie groups). We will use the following compatible norm on $\mathrm{PSL}_2(\mathbb{R})$: For

$$A = \pm \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \in \mathrm{PSL}_2(\mathbb{R})$$

define the norm of A as

$$\|A\| = \max_{1 \leq i,j \leq 2} |a_{ij}|.$$

Definition 1.3.7. A *Fuchsian group* is a subgroup $\Gamma \leq \mathrm{PSL}_2(\mathbb{R})$ such that the induced topology on Γ is the discrete topology.

Proposition 1.3.8. *Let $\Gamma \leq \mathrm{PSL}_2(\mathbb{R})$ be a Fuchsian group, then its action on $\mathbb{H}$ is proper.*

We have the following direct consequence of the Schwarz-Milnor lemma.

Corollary 1.3.9. *Every cocompact Fuchsian group is quasi-isometric to* $\mathbb{H}$*.*

Each subgroup $\Gamma \leqslant \mathrm{PSL}_2(\mathbb{R})$ also has a natural left action on $\mathrm{PSL}_2(\mathbb{R})$ by left multiplication whose orbit space we denote $\Gamma\backslash \mathrm{PSL}_2(\mathbb{R})$. The following well-known proposition will be useful later on.

Proposition 1.3.10. *A Fuchsian group* Γ *is cocompact if and only if* $\Gamma\backslash \mathrm{PSL}_2(\mathbb{R})$ *is compact.*

Proof. Let $K = \{g \in \mathrm{PSL}_2(\mathbb{R}) : g{\cdot}i = i\}$ be the stabilizer subgroup of i. A straightforward computation shows that K is the projective special orthogonal group, that is, $K = \mathrm{PSO}_2(\mathbb{R}) = \mathrm{SO}_2(\mathbb{R})/\{\pm I_2\}$, where

$$\mathrm{SO}_2(\mathbb{R}) = \left\{ \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} : \theta \in \mathbb{R} \right\}.$$

It follows that we have a natural bijection $\varphi\colon \mathrm{PSL}_2(\mathbb{R})/\mathrm{PSO}_2(\mathbb{R}) \to \mathbb{H}$ given by $\varphi(g\,\mathrm{PSO}_2(\mathbb{R})) = g \cdot i$. Moreover, this bijection is a homeomorphism (see [26, Theorem 1.1]). Put $S = \mathrm{PSL}_2(\mathbb{R})/\mathrm{PSO}_2(\mathbb{R})$, again, Γ acts on S by left multiplication and we can form the double quotient

$$\Gamma\backslash S = \Gamma\backslash\left(\mathrm{PSL}_2(\mathbb{R})/\mathrm{PSO}_2(\mathbb{R})\right).$$

The homeomorphism φ preserves the respective actions of Γ, so $\Gamma\backslash S$ is homeomorphic to $\Gamma\backslash\mathbb{H}$. Since $\mathrm{PSO}_2(\mathbb{R})$ is compact, it follows that $\Gamma\backslash \mathrm{PSL}_2(\mathbb{R})$ is compact if and only if $\Gamma\backslash S$ is compact (see [26, Proposition 1.9]) and thus the result follows. □

Finally, we state the following consequence from the general theory of hyperbolic groups.

Theorem 1.3.11. *Every cocompact Fuchsian group is finitely presented and has solvable word problem.*

Proof. A proof may be found in [21, Section 7.4]. □

2 Computable quasi-isometries

This is the main chapter of the thesis. We start by introducing computable metric spaces and some basic definitions such as computable functions between them.

In section 2.2 we define computable quasi-isometries. We will see that the obvious way to define them is not ideal since it does not induce an equivalence relation in the class of computable metric spaces, so we propose another definition with does have this property and also generalizes well with the standard notions of computablity on discrete sets. Then, we show that this definition is not trivial by constructing a computable metric space which admits a quasi-isometric embedding of $\mathbb{N}$ but does not admit a computable one.

In section 2.3 we apply this notion to finitely generated groups with solvable word problem. In particular, we show that all cocompact Fuchsian groups are computably quasi-isometric.

2.1 Computable metric spaces

Computable metric spaces are a concept from computable analysis, an area which extends the ideas from computability theory to the realm of analysis. They let us do computability on spaces which are not necessarily countable, for example the real numbers. The main theme in computable analysis is regarding computability as a refinement of continuity, so topology plays a big role. A major difference with classical computability is that there are several non-equivalent definitions of computable function. We choose the Type-2 Theory of Effectivity laid out in [30]. Another good reference for computable analysis is [5]. Lastly, [4] shows how this theory can be applied to dynamical systems.

Definition 2.1.1. A *computable metric space* is a triple $(X, d, (s_i)_{i\in\mathbb{N}})$ where (X, d) is a metric space and $(s_i)_{i\in\mathbb{N}}$ is a dense sequence with no repetitions in X such that there is a computable function $f\colon \mathbb{N}^3 \to \mathbb{Q}$ with

$$|d(s_i, s_j) - f(i, j, n)| \leq 2^{-n}, \text{ and}$$

We say that $(s_i)_{i\in\mathbb{N}}$ is a *computable structure* on X and refer to its elements as the *special points*.

In other words, we have a bijective coding for a countable dense subset of X with which we can effectively approximate the distance between its elements.

Example 2.1.2.

- Take the real numbers $\mathbb{R}$ with the euclidean metric $d(x, y) = |x-y|$ and the coding of the rational numbers $(q_n)_{n\in\mathbb{N}}$ given in example 1.1.4. Since this coding makes addition and multiplication of rationals computable, it is easy to see that the function $(i, j) \mapsto |s_i - s_j|$ is computable, therefore, $(q_n)_{n\in\mathbb{N}}$ would be a computable structure on $\mathbb{R}$, except it has repetitions. We eliminate them as follows. (Compare with theorem 1.2.8) Let

  $$D = \{n \in \mathbb{N} : n = \langle (n)_0, (n)_1, (n)_2\rangle \text{ where } (n)_0 \in \{0, 1\} \text{ and } (n)_1, (n)_2 + 1 \text{ are coprime}\},$$

  then D is a computable subset of $\mathbb{N}$ in bijection with $\mathbb{Q}$. By corollary 1.1.19 there exists a computable bijection $g\colon \mathbb{N} \to D$. Thus, $(q_{g(n)})_{n\in\mathbb{N}}$ is an enumeration of the rationals without repetitions. Since g is computable, addition and multiplication are still computable, so this sequence is a computable structure on $\mathbb{R}$.

- We can similarly construct a bijective coding for the set $\{p + iq : p, q \in \mathbb{Q}, q > 0\} \subset \mathbb{H}$, the rational points of the hyperbolic plane. It is easy to intuitively see from its explicit formula (theorem 1.3.4) that the hyperbolic distance is computable. Formally proving this requires more work which we will omit.

- The Baire space $\mathbb{N}^{\mathbb{N}}$ has the compatible metric $d(x, y) = 2^{-\min\{n\in\mathbb{N}: x_n \neq y_n\}}$ and the countable dense set of eventually 0 sequences $S = \{x \in \mathbb{N}^{\mathbb{N}} : (\exists n \in \mathbb{N})\,(\forall k \geq n)(x_k = 0)\}$. We construct a coding for S. Let

  $$D = \{\sigma \in \mathbb{N}^{<\mathbb{N}} : \sigma(l(\sigma) - 1) \neq 0\},$$

  then, as before, D is an infinite computable subset of $\mathbb{N}^{<\mathbb{N}}$ in bijection with S, so there exists a computable bijection $g\colon \mathbb{N} \to D$. Define for each $n \in \mathbb{N}$, $s_n = g(n)0^\infty \in S$. The sequence $(s_n)_{n\in\mathbb{N}}$ is a bijective coding for S and it is not hard to see that it makes the distance computable, therefore it is a computable structure on $\mathbb{N}^{\mathbb{N}}$.

- The Cantor space $2^{\mathbb{N}}$ has a computable structure analogous to the Baire space.
- Let G be a finitely generated group with solvable word problem, and let S be a finite generating set of G. The metric space (G, d_S) with the word metric is discrete and countable, so a computable structure on it is simply a coding for G that makes d_S computable. Let $\nu\colon \mathbb{N} \to G$ be the bijective coding given by theorem 1.2.8. We compute d_S in the following way: Since ν is bijective we identify each element of G by its ν-code. Given $g, h \in G$, computably enumerate all words $w \in (S \cup S^{-1})^{<\mathbb{N}}$ ordered by length. Since the group operations are computable, we may check one by one if $g^{-1}hw = e$. Output $l(w)$ for the first w that satisfies $g^{-1}hw = e$.
- Define a metric d on $\mathbb{N}^{<\mathbb{N}}$ in the following way:

$$d(\sigma,\tau) = \begin{cases} l(\tau) - l(\sigma) & \text{if } \sigma \preceq \tau, \\ l(\sigma) - l(\tau) & \text{if } \tau \preceq \sigma, \\ l(\sigma) + l(\tau) - 2\min\{i : \sigma(i) \neq \tau(i)\} & \text{otherwise.} \end{cases}$$

  This is just the graph metric when regarding $\mathbb{N}^{<\mathbb{N}}$ as a graph, as in example 1.2.15. Again, a computable structure is just a coding of $\mathbb{N}^{<\mathbb{N}}$ that makes d computable. The standard coding for $\mathbb{N}^{<\mathbb{N}}$ defined in example 1.1.4 makes the prefix relation, the length and the evaluation functions computable, so d is clearly computable.

The requirement in definition 2.1.1 for the computable structure to have no repetitions is a technical detail which we use in propositions 2.2.4 and 2.2.5. In fact, this requirement is superfluous since one can computably eliminate the repetitions of a computable structure, see [12].

The following technical lemma will be used later in this section.

Lemma 2.1.3. *Let $(X, d, (s_i))$ be a computable metric space, then the relation*

$$\{(i,j,q) \in \mathbb{N}^2 \times \mathbb{Q} : d(s_i, s_j) < q\}$$

is computably enumerable.

Proof. Let $f\colon \mathbb{N}^3 \to \mathbb{Q}$ be the computable function from definition 2.1.1. We have for all $i, j \in \mathbb{N}^2$ and all $q \in \mathbb{Q}$,

$$d(s_i, s_j) < q \iff (\exists n \in \mathbb{N})\left(f(i,j,n) < q - 2^{-n}\right)$$

Indeed, if $d(s_i,s_j) < q$, take $n \in \mathbb{N}$ such that $2^{-n+1} < q - d(s_i,s_j)$, then $|d(s_i,s_j) - f(i,j,n)| \leq 2^{-n}$. We have two cases: If $f(i,j,n) \leq d(s_i,s_j)$ then obviously $f(i,j,n) < q - 2^{-n}$, if $f(i,j,n) > d(s_i,s_j)$ then

$$f(i,j,n) = f(i,j,n) - d(s_i,s_j) + d(s_i,s_j) < 2^{-n} + q - 2^{-n+1} = q - 2^{-n}.$$

Conversely, if there exists $n \in \mathbb{N}$ with $f(i,j,n) < q - 2^{-n}$ then

$$d(s_i,s_j) = d(s_i,s_j) - f(i,j,n) + f(i,j,n) < 2^{-n} + q - 2^{-n} = q.$$

Now, the relation $\{(n,i,j,q) \in \mathbb{N}^3 \times \mathbb{Q} : f(i,j,n) < q - 2^{-n}\}$ is clearly computable, so by the normal form theorem 1.1.15 we are done. □

Let $(X, d, (s_i))$ be a computable metric space. For $q \in \mathbb{Q}$ and $x \in X$, we denote by $B(x,q) = \{y \in X : d(x,y) < q\}$ the open ball with center x and radius q. A *basic open ball* is an open ball with rational radius centered on a special point. The basic open balls form a countable base for the topology of X of which we will fix the following enumeration

$$B_n = B(s_{(n)_0}, q_{(n)_1}),$$

where $\langle (n)_0, (n)_1 \rangle = n$ and $(q_i)_{i\in\mathbb{N}}$ is the standard enumeration of the rationals given in example 1.1.4.

Of course, every open set is a countable union of basic open balls. Analogous to definition 1.1.29, a set $U \subseteq$ is called an *effectively open set* or a Σ^0_1 set if it is an effective countable union of basic open balls, that is, there exists a c.e. set $I \subseteq \mathbb{N}$ such that

$$U = \bigcup_{n\in I} B_n.$$

A set F is called an *effectively closed set* or a Π^0_1 set if its complement is an effectively open set, that is, there exists a c.e. set $I \subseteq \mathbb{N}$ such that

$$F = \bigcap_{n\in I} (X \setminus B_n).$$

A sequence of effectively open sets $\{U_n\}_{n\in\mathbb{N}}$ are *uniformly open* if there is a c.e. relation $I \subseteq \mathbb{N}^2$ such that

$$U_n = \bigcup \{B_j : (n,j) \in I\}, \text{ for all } n \in \mathbb{N}.$$

It is not hard to see that when $X = \mathbb{N}^{\mathbb{N}}$ (with the computable structure described in example 2.1.2) these definitions coincide with definition 1.1.29, so it is indeed a generalization.

Recall from definition 1.1.3 that in order to do computability in arbitrary countable sets we use codings, which are just surjections of the natural numbers onto the set in question. In the context of computable metric spaces, the analogous tools are surjections of the Baire space, which are called *representations*.

Definition 2.1.4. Let $(X, d, (s_i))$ be a computable metric space. A *Cauchy name* for $x \in X$ is an infinite sequence $f \in \mathbb{N}^{\mathbb{N}}$ such that $(s_{f(n)})_{n\in\mathbb{N}}$ converges rapidly to x, that is,

$$d(s_{f(n)}, x) < 2^{-n} \quad \forall n \in \mathbb{N}.$$

The *Cauchy representation* of X is the partial function $\delta_X :\subseteq \mathbb{N}^{\mathbb{N}} \to X$ defined by

$$\delta_X(f) = x \iff f \text{ is a Cauchy name for } x.$$

We say that x is a *computable point* of X if it has a computable Cauchy name.

As implied by the name, the Cauchy representation is not the only possible representation for a computable metric space, but we will not dive further into this theory of represented spaces which is much more general, for example one may speak of computable topological spaces by defining suitable representations. The interested reader may consult [30].

Example 2.1.5.

- A Cauchy name for a special point $s_i \in X$ is the constant sequence $(i\ i\ i \dots)$ which is clearly computable. Thus, every special point is a computable point.
- The number e is given by the series $\sum_{k=0}^{\infty} \frac{1}{k!}$ which by elementary analysis converges rapidly, therefore the sequence $(f(n))_{n\in\mathbb{N}}$ with $r_{f(n)} = \sum_{k=0}^{n} \frac{1}{k!}$ is a Cauchy name for e. It is easy to see that f is computable, and thus e is a computable point of $\mathbb{R}$ with the standard computable structure given before.
- A non computable example is the real number $h = \sum_{k\in K} 2^{-k}$ where K is the non computable set from theorem 1.1.9. h can be seen as the real whose nth digit in its binary expansion is a 1 if $n \in K$ and a 0 otherwise. Suppose that h has a computable Cauchy name f, then we can decide $n \in K$ in the following way: Compute $f(n+1)$, we have $|s_{f(n+1)} - h| < 2^{-n-1}$, which means that the binary expansions of $s_{f(n+1)}$ and h agree up to the nth digit, so, check the nth digit of the rational $s_{f(n+1)}$, if it is 1, then $n \in K$, otherwise $n \notin K$.

Definition 2.1.6. Let X and Y be computable metric spaces. A function $F : X \to Y$ is *computable (in the metric sense)* if there exists a computable functional $G :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ such that for each point $x \in X$ and for every Cauchy name χ for x, $G(\chi)$ is a Cauchy name for $F(x)$, i.e., the following diagram commutes:

$$\begin{tikzcd} X \arrow[r, "F"] & Y \\ \mathbb{N}^{\mathbb{N}} \arrow[u, "\delta_X"] \arrow[r, "G"] & \mathbb{N}^{\mathbb{N}} \arrow[u, "\delta_Y"'] \end{tikzcd}$$

The next proposition lets us reduce to the classical computability notions when we are dealing with a special kind of metric space. A metric space (X, d) is *uniformly discrete* if there exists $a \in \mathbb{R}$ such that $d(x, x') > a$ for all $x, x' \in X$ with $x \neq x'$. For example, any connected graph is uniformly discrete, in particular the Cayley graph of a finitely generated group.

Proposition 2.1.7. *Let $(X, d_X, (s_i))$ and $(Y, d_Y, (r_i))$ be uniformly discrete computable metric spaces. Then the maps $\nu\colon n \mapsto s_n$ and $\mu\colon n \mapsto r_n$ are codings for X and Y respectively. Furthermore, a function $F\colon X \to Y$ is computable in the metric sense if and only if it is (ν, μ)-computable.*

Proof. The only dense set in a discrete space is the space itself, so ν and μ are surjective.

Let $F : X \to Y$ be computable in the metric sense, then there is a computable $G :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ such that for each $x \in X$ and each Cauchy name χ for x, $G(\chi)$ is a Cauchy name for $F(x)$. We must find a computable $f : \mathbb{N} \to \mathbb{N}$ such that $F(s_n) = r_{f(n)}$ for all n. Let $k \in \mathbb{N}$ be such that $d_Y(y, y') > 2^{-k}$ for all $y, y' \in Y$ with $y \neq y'$. Given $n \in \mathbb{N}$, we compute $f(n)$ in the following way: The constant sequence $(n, n, \dots)$ is a computable Cauchy name for s_n and so $G((n, n, \dots))$ is a computable Cauchy name for $F(s_n)$. Output $f(n) = G((n, n, \dots))(k)$. We have that $d_Y(r_{f(n)}, F(s_n)) < 2^{-k}$ so by the choice of k, $r_{f(n)} = F(s_n)$. Therefore, F is (ν, μ)-computable.

Conversely, let $F : X \to Y$ be (ν, μ)-computable, then there is a computable $f : \mathbb{N} \to \mathbb{N}$ such that $F(s_n) = r_{f(n)}$ for all n. We must find a computable $G :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ that takes any Cauchy name for any $x \in X$ and outputs a Cauchy name for $F(x)$. Again, let $l \in \mathbb{N}$ be such that $d_X(x, x') > 2^{-l}$ for all $x, x' \in X$ with $x \neq x'$. Given χ a Cauchy name for x, let $G(\chi)$ be the constant sequence $(f(\chi(l)), f(\chi(l)), \dots)$ which is a Cauchy name for $r_{f(\chi(l))}$. G is clearly computable and we see that by the choice of l, $s_{\chi(l)} = x$ and so $r_{f(\chi(l))} = F(s_{\chi(l)}) = F(x)$. □

The main theorem of computable analysis is the fact that, analogous to theorem 1.1.33, computable functions are continuous. In order to prove this, we first prove some basic properties of the Cauchy representation. As in theorem 1.1.33, denote $U_\sigma = \{x \in \mathbb{N}^{\mathbb{N}} : \sigma \prec x\}$ for $\sigma \in \mathbb{N}^{<\mathbb{N}}$.

Lemma 2.1.8. *Let $(X, d, (s_i))$ be a computable metric space, let $(B_n)_{n\in\mathbb{N}}$ be the standard enumeration of its basic open balls and let $\delta :\subseteq \mathbb{N}^{\mathbb{N}} \to X$ be its Cauchy representation.*

1. *δ is effectively continuous, i.e., there is a c.e. relation $I \subseteq \mathbb{N} \times \mathbb{N}^{<\mathbb{N}}$ such that*

$$\delta^{-1}(B_n) = \mathrm{dom}(\delta) \cap \bigcup\{U_\sigma : (n, \sigma) \in I\} \text{ for all } n \in \mathbb{N}.$$

2. *δ is effectively open, i.e., there is a c.e. relation $J \subseteq \mathbb{N}^{<\mathbb{N}} \times \mathbb{N}$ such that*

$$\delta(U_\sigma) = \bigcup\{B_n : (\sigma, n) \in J\} \text{ for all } \sigma \in \mathbb{N}^{<\mathbb{N}}.$$

Proof. We prove (1) first. Let $I \subseteq \mathbb{N} \times \mathbb{N}^{<\mathbb{N}}$ be the relation defined by

$$I(n, \sigma) \iff n = \langle (n)_0, (n)_1 \rangle \text{ and } d(s_{(n)_0}, s_{\sigma(l(\sigma)-1)}) < q_{(n)_1} - 2^{-l(\sigma)+1},$$

where $(q_j)_{j\in\mathbb{N}}$ is the standard enumeration of the rationals. Then, it is easy to see that lemma 2.1.3 implies that R is computably enumerable. Let $n \in \mathbb{N}$, we check

$$\delta^{-1}(B_n) = \mathrm{dom}(\delta) \cap \bigcup\{U_\sigma : (n, \sigma) \in I\}.$$

If $f \in \delta^{-1}(B_n)$, then $(s_{f(i)})_{i\in\mathbb{N}}$ converges rapidly to $\delta(f)$ and $d(s_{(n)_0}, \delta(f)) < r_{(n)_1}$. By an elementary metric argument, this implies that there exists $m \in \mathbb{N}$ such that

$$d(s_{(n)_0}, s_{f(m)}) < r_{(n)_1} - 2^{-m},$$

so the string $\sigma = f|(m+1) = f(0)f(1)\cdots f(m)$ satisfies $R(n, \sigma)$ and $\sigma \prec f$.

Conversely, if $f \in \mathrm{dom}(\delta) \cap U_\sigma$ for some σ that satisfies $I(n, \sigma)$, then

$$d(s_{\sigma(l(\sigma)-1)}, \delta(f)) < 2^{-l(\sigma)+1} \text{ and } d(s_{(n)_0}, s_{\sigma(l(\sigma)-1)}) < q_{(n)_1} - 2^{-l(\sigma)+1},$$

which both imply that $\delta(f) \in B_n$.

We now prove (2). Let $J \subseteq \mathbb{N}^{<\mathbb{N}} \times \mathbb{N}$ be the relation defined by

$$J(\sigma, n) \iff n = \langle (n)_0, (n)_1 \rangle \text{ and } (\forall i < l(\sigma))\, [d(s_{\sigma(i)}, s_{(n)_0}) < 2^{-i} - q_{(n)_1}],$$

then again by lemma 2.1.3, J is c.e. Let $\sigma \in \mathbb{N}^{<\mathbb{N}}$, we check

$$\delta(U_\sigma) = \bigcup\{B_n : (\sigma, n) \in J\} \text{ for all } \sigma \in \mathbb{N}^{<\mathbb{N}}.$$

If $x \in \delta(U_\sigma)$, then $x = \delta(f)$ for some $f \succ \sigma$, which means

$$(\forall i < l(\sigma))\,[d(s_{\sigma(i)}, x) < 2^{-i}],$$

that is to say,

$$x \in B = B(s_{\sigma(0)}, 2^0) \cap B(s_{\sigma(1)}, 2^{-1}) \cap \cdots \cap B(s_{\sigma(l(\sigma)-1)}, 2^{-l(\sigma)+1}),$$

so B is a nonempty open set. By density, there exists $s_j \in B$ and so there is some $q_k \in \mathbb{Q}$ such that

$$(\forall i < l(\sigma))\,[d(s_{\sigma(i)}, s_j) < 2^{-i} - q_k] \text{ and } x \in B(s_j, q_k)$$

which is just $R(\sigma, \langle j,k\rangle)$ and $x \in B_{\langle j,k\rangle}$.

Conversely, let $x \in B_n$ such that $J(\sigma, n)$ is satisfied, this means

$$d(x, s_{(n)_0}) < q_{(n)_1} \text{ and } (\forall i < l(\sigma))\,[d(s_{\sigma(i)}, s_{(n)_0}) < 2^{-i} - q_{(n)_1}],$$

which by metric arguments implies

$$(\forall i < l(\sigma))\,[d(s_{\sigma(i)}, x) < 2^{-i}].$$

From this we observe that σ is a valid prefix for a Cauchy name for x, so we can find a Cauchy name f for x with $\sigma \prec f$. We conclude $x \in \delta(U_\sigma)$. □

Analogous to theorem 1.1.33 we have the following.

Theorem 2.1.9. *Let $(X, d_X, (s_i))$ and $(Y, d_Y, (r_i))$ be computable metric spaces with basic open balls $\{B_n\}_{n\in\mathbb{N}}$ and $\{V_n\}_{n\in\mathbb{N}}$ respectively. A function $F\colon X \to Y$ is computable if and only if it is effectively continuous, i.e., the sequence of sets $\{F^{-1}(V_n)\}_{n\in\mathbb{N}}$ are uniformly open.*

Proof. Let $F\colon X \to Y$ be computable, then there is a computable $G :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ such that $(F \circ \delta_X)(\chi) = (\delta_Y \circ G)(\chi)$ for all $\chi \in \mathrm{dom}(\delta_X)$, where δ_X and δ_Y are the respective Cauchy representations. From this we have

$$F^{-1}(V_n) = \delta_X(G^{-1}(\delta_Y^{-1}(V_n))) \text{ for all } n \in \mathbb{N}.$$

By lemma 2.1.8 and theorem 1.1.33, δ_X is effectively open and δ_Y and G are effectively continuous, so it is not hard to see that we can uniformly enumerate the basic open balls that make up $\delta_X(G^{-1}(\delta_Y^{-1}(V_n)))$. We omit the details.

Conversely, assume that F is effectively continuous. By lemma 2.1.8, δ_X is effectively continuous and it is not hard to see that the composition of two effectively continuous functions is effectively continuous, so $F \circ \delta_X$ is effectively continuous. This means there is a c.e. relation $I \subseteq \mathbb{N} \times \mathbb{N}^{<\mathbb{N}}$ such that $\delta_X^{-1}(F^{-1}(V_n)) = \bigcup\{U_\sigma : (n, \sigma) \in I\}$. By the listing theorem 1.1.18, there is a computable function $f\colon \mathbb{N} \to \mathbb{N} \times \mathbb{N}^{<\mathbb{N}}$ with $\mathrm{range}(f) = I$.

Let $G\colon \mathrm{dom}(\delta_X) \to \mathbb{N}^\mathbb{N}$ be the functional computed in the following way: Let $\chi \in \mathbb{N}^\mathbb{N}$ be a Cauchy name for some $x \in X$. To compute $G(\chi)(m)$, keep computing $f(s)$ for each $s = 0, 1, 2, \ldots$ until we find a pair (n, σ) such that V_n is a ball of radius at most 2^{-m} and $\sigma \prec \chi$. This means that $F(x) \in B(r_{(n)_0}, 2^{-m})$, where $n = \langle (n)_0, (n)_1\rangle$, so output $G(\chi)(m) = (n)_0$. It is easy to see that $G(\chi)$ is a Cauchy name for $F(x)$. □

2.2 Computable quasi-isometries

Recall that our goal is to come up with an adequate definition of computable quasi-isometry. One way to do this is, as it is done in [3], simply defining them as quasi-isometries which are additionally computable in the sense of definition 2.1.6. However, this is too strong, as theorem 2.1.9 implies that all computable functions are continuous, and quasi-isometries are rarely continuous. For example, recall from the discussion following definition 1.2.19 that the inclusion $\iota\colon \mathbb{Z} \to \mathbb{R}$ is a quasi-isometry and it is clearly computable in the metric sense. However, no quasi-inverse $h\colon \mathbb{R} \to \mathbb{Z}$ can be continuous, let alone computable, since $\mathbb{Z}$ is discrete and $\mathbb{R}$ is connected. Therefore, there would be computable quasi-isometries that have no computable quasi-inverse.

We will instead use a weaker notion of computability inspired from [23].

Definition 2.2.1. Let X and Y be computable metric spaces and $(s_i)_{i\in\mathbb{N}}$ the computable structure of X. A function $f\colon X\to Y$ is *computable on the special points* if there exists a computable function $F\colon \mathbb{N}^2\to\mathbb{N}$ such that, for every $i\in\mathbb{N}$, $(F(i,n))_{n\in\mathbb{N}}$ is a Cauchy name for $f(s_i)$.

In other words, we have a procedure that enumerates a Cauchy name for $f(s_i)$ for each i uniformly.

Proposition 2.2.2. *Let $(X,d_X,(s_i))$ and $(Y,d_Y,(r_i))$ be uniformly discrete computable metric spaces and let $\nu\colon n\mapsto s_n$ and $\mu\colon n\mapsto r_n$ be the canonical codings for X and Y respectively. A function $F\colon X\to Y$ is computable on the special points if and only if it is (ν,μ)-computable.*

Proof. This is analogous to the proof of proposition 2.1.7. □

We are now ready to state the main definition of this thesis.

Definition 2.2.3. A quasi-isometry $f\colon X\to Y$ between computable metric spaces X and Y is called a *computable quasi-isometry* if it is computable on the special points. In this case we say that X and Y are *computably quasi-isometric* and write $X\sim_{CQI} Y$.

Coming back to the example of $\mathbb{Z}$ and $\mathbb{R}$, it is clear that the inclusion $\iota\colon\mathbb{Z}\to\mathbb{R}$ is computable on the special points, so it is a computable quasi-isometry in this sense. This time, we do have a computable quasi-inverse, the floor function $h\colon\mathbb{R}\to\mathbb{Z}$, given by $h(x)=\max\{k\in\mathbb{Z} : k\le x\}$ for all $x\in\mathbb{R}$, is computable in the special points. Indeed, given a rational number q, we can computably find the integer k such that $h(q)=k$.

In fact, we always have a computable quasi-inverse, as the following effective version of proposition 1.2.21 shows. Notice that the proof of proposition 1.2.21 is completely non-constructive since it uses the axiom of choice, so we have to be more precise and give an actual construction of the quasi-inverse, at least in the special points. This is a recurring theme when adapting proofs to the computable world.

Proposition 2.2.4. *Let $(X,d_X,(s_i))$ and $(Y,d_Y,(r_i))$ be computable metric spaces. Every computable quasi-isometry $f\colon X\to Y$ has a computable quasi-inverse quasi-isometry $g\colon Y\to X$.*

Proof. Let $c>0$ be a constant rational number such that

$$(\forall x,x'\in X)\quad \frac{1}{c}d_X(x,x')-c\le d_Y(f(x),f(x'))\le c\,d_X(x,x')+c,$$

$$(\forall y\in Y)(\exists x\in X)d_Y(f(x),y)<c.$$

Since $(s_i)_{i\in\mathbb{N}}$ is dense in X we may assume that c satisfies

$$(\forall y\in Y)(\exists i\in\mathbb{N})d_Y(f(s_i),y)<c. \tag{1}$$

Let $F\colon\mathbb{N}^2\to\mathbb{N}$ be the computable function from definition 2.2.1 that computes f. For each $i,n\in\mathbb{N}$, denote $y_i^n=r_{F(i,n)}\in Y$, so $d_Y(f(s_i),y_i^n)<2^{-n}$. By the same metric arguments of lemma 2.1.3 we have for every $i,j\in\mathbb{N}$

$$d_Y(r_j,f(s_i))<c \iff (\exists n\in\mathbb{N})(d_Y(r_j,y_i^n)<c-2^{-n}).$$

By lemma 2.1.3 the relation $\{(i,j,n)\in\mathbb{N}^3 : d_Y(r_j,y_i^n)<c-2^{-n}\}$ is computably enumerable, so by the normal form theorem 1.1.15 there exists a computable relation $R\subseteq\mathbb{N}^4$ such that

$$d_Y(r_j,f(s_i))<c \iff (\exists n\in\mathbb{N})(\exists m\in\mathbb{N})R(i,j,n,m).$$

We first define g on the special points of Y. For each $j\in\mathbb{N}$, let m be the least natural number such that $m=\langle (m)_0,(m)_1,(m)_2\rangle$ and $R((m)_0,j,(m)_1,(m)_2)$ (here $\langle .\rangle$ denotes the standard pairing from example 1.1.4). We define $g(r_j)=s_{(m)_0}$. By (1) and the fact that the sequence $(r_i)_{i\in\mathbb{N}}$ has no repetitions, $g(r_j)$ is well-defined for all j. For the non-special points Y we just use the axiom of choice as in proposition 1.2.21. By the same calculation, g is a quasi-inverse quasi-isometry of f.

It is easy to see that g is computable on the special points: For each $j\in\mathbb{N}$ we can effectively compute the least m with $R((m)_0,j,(m)_1,(m)_2)$ and then output the constant Cauchy name $((m)_0,(m)_0,\dots)$. □

With this we have symmetry of $\sim_{CQI}$. For transitivity we again run into problems because the composition of two computable on the special points functions is not necessarily computable on the special points. For example, take the functions $f, g\colon \mathbb{R} \to \mathbb{R}$ defined by

$$f(x) = x + \sqrt{2} \quad \text{and} \quad g(x) = \begin{cases} 0 & \text{if } x \in \mathbb{Q}, \\ h & \text{if } x \notin \mathbb{Q}, \end{cases}$$

where h is a non-computable real number (such as the one given in example 2.1.5). Then, f and g are computable on the special points, but the composition $g \circ f$ is constant equal to h on $\mathbb{Q}$, so it cannot be computable on the special points.

Fortunately, the notion of quasi-isometry is quite flexible and gives us a way around this issue.

Proposition 2.2.5. *Let $(X, d_X, (s_i))$, $(Y, d_Y, (r_i))$ and $(Z, d_Z, (t_i))$ be computable metric spaces. If $f\colon X \to Y$ and $g\colon Y \to Z$ are computable quasi-isometries then there exists a computable quasi-isometry $h\colon X \to Z$ at finite distance from $g \circ f$.*

Proof. For each $i \in \mathbb{N}$, we compute $h(s_i)$ in the following way: Compute $y_i^0 \in Y$ the first term of the Cauchy name for $f(s_i)$ given by the algorithm of f, then $d_Y(y_i^0, f(s_i)) < 1$. Lastly, use the algorithm of g for outputting a Cauchy name for $g(y_i^0)$.

This procedure well-defines $h(s_i) = g(y_i^0)$. For the non-special points $x \in X$ we just define $h(x) = g(f(x))$. Let $c > 0$ be a quasi-isometry constant for g. We have for every $i \in \mathbb{N}$

$$d_Z(h(s_i), g(f(s_i))) = d_Z(g(y_i^0), g(f(s_i))) \leq c \cdot d_Y(y_i^0, f(s_i)) + c < 2c,$$

and for $x \in X$ non-special $d_Z(h(x), g(f(x))) = 0$. Therefore, h is at finite distance from the quasi-isometry $g \circ f$. It is easy to see that this implies that h is also a quasi-isometry. □

Corollary 2.2.6. *The relation $\sim_{CQI}$ is an equivalence relation in the class of computable metric spaces.*

Now that we have established the fundamental properties of computable quasi-isometries, the natural question arises: Do they coincide with standard quasi-isometries? More specifically, is there a pair of computable metric spaces which are quasi-isometric but not computably quasi-isometric? We partially answer this question by constructing a computable tree which admits quasi-isometric embeddings of $\mathbb{N}$ such that none of them can be computable. This is a well-known construction in computability theory called the Kleene tree.

Definition 2.2.7. A *subtree* $T \subseteq \mathbb{N}^{<\mathbb{N}}$ is a subset of strings closed under prefixes, i.e., if $\sigma \preceq \tau$ and $\tau \in T$, then $\sigma \in T$.

A subtree $T \subseteq \mathbb{N}^{<\mathbb{N}}$ is said to be *computable* if it is a computable subset of $\mathbb{N}^{<\mathbb{N}}$ (under the coding given in example 1.1.4).

As we have seen in example 2.1.2, the full tree $\mathbb{N}^{<\mathbb{N}}$ is a computable metric space under the standard graph metric. The same is true for computable subtrees.

Proposition 2.2.8. *Let $T \subseteq \mathbb{N}^{<\mathbb{N}}$ be a computable subtree and d the induced graph metric from $\mathbb{N}^{<\mathbb{N}}$, then (T, d) has a natural computable structure that makes it a computable metric space.*

Proof. If T is finite the result is obvious. Assume T is infinite. Let $\nu\colon \mathbb{N} \to \mathbb{N}^{<\mathbb{N}}$ be the standard coding of $\mathbb{N}^{<\mathbb{N}}$ and $A = \nu^{-1}(T)$, then A is a computable subset of $\mathbb{N}$. By corollary 1.1.19 there exists a computable bijection $f\colon \mathbb{N} \to A$. Let $\mu = \nu \circ f$, then μ is a bijective coding from $\mathbb{N}$ to T that makes d computable. □

Definition 2.2.9. Let $x \in \mathbb{N}^{\mathbb{N}}$ and $n \in \mathbb{N}$, then $x|n$ denotes the restriction of x to the first n coordinates. Given a subtree $T \subseteq \mathbb{N}^{<\mathbb{N}}$, an *infinite path* in T is an element $x \in \mathbb{N}^{\mathbb{N}}$ such that $x|n \in T$ for all $n \in \mathbb{N}$. Moreover, if x is computable (as a function from $\mathbb{N}$ to $\mathbb{N}$), we say it is a *computable path.* We denote by $[T]$ the set of all infinite paths of T.

Notice that if x is an infinite path in T, then the function $f\colon \mathbb{N} \to T$ defined by $f(n) = x|n$ is an isometric embedding, in particular a quasi-isometric embedding. The converse is also true even in the computable setting.

Proposition 2.2.10. *Let $T \subseteq \mathbb{N}^{<\mathbb{N}}$ be a computable subtree and $f\colon \mathbb{N} \to T$ a computable quasi-isometric embedding, then T has a computable infinite path.*

Proof. We will show that $f(n)$ converges to an element of $[T]$ when n goes to infinity. Let $a, b \in \mathbb{N}$ be constants such that

$$\frac{1}{a}|i-j| - b \leq d(f(i), f(j)) \leq a|i-j| + b.$$

Given $\sigma, \tau \in T$, denote

$$\rho(\sigma,\tau) = \begin{cases} l(\sigma) & \text{if } \sigma \preceq \tau, \\ l(\tau) & \text{if } \tau \preceq \sigma, \\ \min\{i : \sigma(i) \neq \tau(i)\} & \text{otherwise.} \end{cases}$$

Then, $d(\sigma,\tau) = l(\sigma) + l(\tau) - 2\rho(\sigma,\tau)$. We have,

$$\begin{aligned} \frac{n}{a} - b &\leq d(f(0), f(n)) \\ &= l(f(n)) + l(f(0)) - 2\rho(f(0), f(n)) \\ &\leq l(f(n)) + l(f(0)), \end{aligned}$$

so,

$$l(f(n)) \geq \frac{n}{a} - b - l(f(0)), \text{ for all } n \in \mathbb{N}$$

Thus, $l(f(n)) \to \infty$ when $n \to \infty$. Moreover,

$$l(f(n+1)) + l(f(n)) - 2\rho(f(n+1), f(n)) = d(f(n+1), f(n)) \leq a + b,$$

so that,

$$2\rho(f(n+1), f(n)) \geq \frac{n+1}{a} + \frac{n}{a} - 2l(f(0)) - a - 3b.$$

Therefore, for each $k \in \mathbb{N}$, we may computably find $N(k) \in \mathbb{N}$ such that $f(n)_k$, the kth coordinate of $f(n)$, exists and is constant for every $n \geq N(k)$. Define $x\colon \mathbb{N} \to \mathbb{N}$ as $x(k) = f(N(k))_k$. Then $x \in [T]$ and it is computable. $\square$

Lemma 2.2.11 (König's lemma)**.** *Let $T \subseteq 2^{<\mathbb{N}}$ be an infinite subtree of the binary tree, then T has an infinite path.*

Proof. Given $\sigma \in T$, denote by $T_\sigma = \{\tau \in T : \sigma \preceq \tau\}$. Since $T = T_0 \cup T_1$ and T is infinite, T_i is infinite for some $i \in \{0,1\}$, define $x(0) = i$. Continue recursively in the following way

$$x(n+1) = \begin{cases} 0 & \text{if } T_{x(0)\dots x(n)0} \text{ is infinite,} \\ 1 & \text{otherwise.} \end{cases}$$

Then, by induction $T_{x|n}$ is infinite for all $n \in \mathbb{N}$, in particular $x|n \in T$, thus $x \in [T]$. $\square$

Lemma 2.2.12 (Kleene tree)**.** *There exists a computable infinite subtree $T \subseteq 2^{<\mathbb{N}}$ with no computable paths.*

Proof. First we prove that there exists a disjoint pair of c.e. sets $A, B \subseteq \mathbb{N}$ such that there is no computable set C with $A \subseteq C$ and $C \cap B = \emptyset$. Such a pair are said to be *computably inseparable.* Let

$$\begin{aligned} A &= \{n \in \mathbb{N} : \varphi_n(n) = 0\}, \\ B &= \{n \in \mathbb{N} : \varphi_n(n) = 1\}, \end{aligned}$$

where $\{\varphi_e\}_{e \in \mathbb{N}}$ is a computable enumeration of the p.c. functions, as in theorem 1.1.5. A, B are clearly c.e. Suppose that C is a computable set with $A \subseteq C$ and $C \cap B = \emptyset$. Then there exists an index $e \in \mathbb{N}$ such that φ_e is the characteristic function of C. Now, if $e \in C$, then $\varphi_e(e) = 1$, but this means $e \in B$ which cannot occur since $C \cap B = \emptyset$. On the other hand, if $e \notin C$, then $\varphi_e(e) = 0$, which means $e \in A \subseteq C$. Thus, $e \in C$ iff $e \notin C$, a contradiction.

Let $e, i \in \mathbb{N}$ be indices such that $W_e = A$ and $W_i = B$ (recall definition 1.1.13). Define a computable subtree $T \subseteq 2^{<\mathbb{N}}$ in the following way: Given $\sigma \in 2^{<\mathbb{N}}$, let $s = l(\sigma)$ and put $\sigma \in T$ if and only if

$$\forall n < s \quad (n \in W_{e,s} \implies \sigma(n) = 1) \text{ and } (n \in W_{i,s} \implies \sigma(n) = 0).$$

In this way, $x \in [T]$ if and only if

$$\forall n \in \mathbb{N} \quad (n \in W_e \implies \sigma(n) = 1) \text{ and } (n \in W_i \implies \sigma(n) = 0),$$

that is, x separates W_e and W_i. Therefore, the infinite paths of T are exactly the separating sets of A and B, of which none are computable. □

Corollary 2.2.13. *There exists a computable metric space T that admits a quasi-isometric embedding of $\mathbb{N}$ but no computable quasi-isometric embedding of $\mathbb{N}$.*

Proof. Let T be the tree from the previous lemma. By König's lemma there exists $x \in [T]$. The map $n \mapsto x|n$ is an isometric embedding of $\mathbb{N}$ into T. Since T has no computable paths, proposition 2.2.10 implies there is no computable quasi-isometric embedding of $\mathbb{N}$. □

2.3 Computably quasi-isometric groups

Let G be a finitely generated group with decidable word problem. By theorem 1.2.8, G has a unique computable structure which naturally transfers to the metric space (G, d_S) for any finite generating set S, as we have seen in example 2.1.2. If S' is another finite generating set, then we know by proposition 1.2.23 that the identity map $id\colon (G, d_S) \to (G, d_{S'})$ is a quasi-isometry. By proposition 1.2.5, the identity is also a computable function so we conclude that G has a well defined computable quasi-isometry type.

Definition 2.3.1. Let G be a finitely generated group with decidable word problem. The group G is *computably quasi-isometric* to a computable metric space X if for some (and hence every) finite generating set S of G the computable metric spaces (G, d_S) and X are computably quasi-isometric. We write $G \sim_{CQI} X$.

Theorem 2.3.2 (Effective Schwarz-Milnor lemma)**.** *Let G be a finitely generated group with decidable word problem acting by computable isometries on a proper, geodesic and computable metric space $(X, d, (s_i))$. Furthermore, suppose that this action is proper and cocompact. Then for all computable $x \in X$ the map*

$$\begin{aligned} G &\longrightarrow X \\ g &\longmapsto g \cdot x \end{aligned}$$

is a computable quasi-isometry.

Proof. By theorem 1.2.26 we just have to show that the map $f\colon G \to X$ with $f(g) = g \cdot x$ is computable on the special points. Let S be a finite generating set of G and for each $s \in S$, let $f_s\colon X \to X, x \mapsto s \cdot x$ be the computable isometry associated to s. Given $g \in G$, find $s_0, \ldots, s_{n-1} \in S$ with $g = s_0 \cdots s_{n-1}$, then $f(g) = (f_{s_0} \circ \cdots \circ f_{s_{n-1}})(x)$, so just enumerate a Cauchy name for x and use the algorithms of each f_{s_i} to get a Cauchy name for $f(g)$. □

Recall from theorem 1.3.11 that cocompact Fuchsian groups have solvable word problem and from example 2.1.2 that $\mathbb{H}$ is a computable metric space. We will use the effective Schwarz-Milnor lemma to show that cocompact Fuchsian groups are computably quasi-isometric to $\mathbb{H}$. We do this by replacing the coefficients of the elements of the group by computable real numbers. We shall need the following result of A. Weil [31].

Theorem 2.3.3 (Weil [31])**.** *Let G be a connected Lie group and Γ a discrete group. Denote*

$$\mathcal{R} = \{\varphi\colon \Gamma \to G \mid \varphi \text{ is a group homomorphism}\}$$

and give it the induced topology from the product G^Γ. Let $\mathcal{R}_0$ be the subset of all $\varphi\colon \Gamma \to G$ injective group homomorphisms such that $\varphi(\Gamma)$ is discrete in G with compact quotient space $G/\varphi(\Gamma)$. Then $\mathcal{R}_0$ is an open subset of $\mathcal{R}$.

Applying this to our context we have the following corollary.

Corollary 2.3.4. *Let Γ be a cocompact Fuchsian group generated by $\gamma_1, \dots, \gamma_n$. Then, there exists $\delta > 0$ such that, if $\varphi\colon \Gamma \to \mathrm{PSL}_2(\mathbb{R})$ is a group homomorphism with $\|\gamma_i - \varphi(\gamma_i)\| < \delta$ for every $i = 1, \dots, n$, then $\varphi(\Gamma)$ is a cocompact Fuchsian group isomorphic to Γ.*

Proof. Let $G = \mathrm{PSL}_2(\mathbb{R})$ in the previous theorem. Since Γ is generated by $\gamma_1, \dots, \gamma_n$, $\mathcal{R}$ is homeomorphic to $(\mathrm{PSL}_2(\mathbb{R}))^n$. By proposition 1.3.10, the set $\mathcal{R}_0$ coincides with the subset of all $\varphi\colon \Gamma \to \mathrm{PSL}_2(\mathbb{R})$ such that φ is an injective group homomorphism and $\varphi(\Gamma)$ is a cocompact Fuchsian group. Therefore, our δ comes from a neighborhood of the inclusion map contained in the open set $\mathcal{R}_0$. ☐

Proposition 2.3.5. *Let Γ be a cocompact Fuchsian group. Then there exists a cocompact Fuchsian group Γ' isomorphic to Γ such that the coefficients of the elements of Γ' are all computable real numbers.*

Proof. Let Γ be a cocompact Fuchsian group generated by $\gamma_1, \dots, \gamma_n \in \mathrm{PSL}_2(\mathbb{R})$. By adjoining the finite number of coefficients of each γ_i to $\mathbb{Q}$, we may regard Γ as a subgroup of $\mathrm{PSL}_2(K)$, where $K = \mathbb{Q}(\alpha_1, \dots, \alpha_k, a_1, \dots, a_l)$, $\alpha_1, \dots, \alpha_k$ are algebraically independent over $\mathbb{Q}$ and $a_1, \dots, a_l$ are algebraic over $L = \mathbb{Q}(\alpha_1, \dots, \alpha_k)$. For each $i = 1, \dots, l$, let $p_i \in L[x]$ be the minimal polynomial of a_i over L. L is isomorphic to $\mathbb{Q}(x_1, \dots, x_l)$ the field of rational functions of l variables, so we may regard p_i as a rational function $p_i(x_1, \dots, x_l, x)$ of $l+1$ variables with rational coefficients. Since L is of characteristic 0 and p_i is irreducible over $L[x]$, $\frac{\partial p_i}{\partial x}(\alpha_1, \dots, \alpha_l, a_i) \neq 0$, therefore, by the implicit function theorem, we have an open set $U_i \subseteq \mathbb{R}^l$ containing the point $(\alpha_1, \dots, \alpha_l)$ and a continuous function $f_i\colon U_i \to \mathbb{R}$ such that $f_i(\alpha_1, \dots, \alpha_l) = a_i$ and $p_i(x_1, \dots, x_l, f_i(x_1, \dots, x_l)) = 0$ for every $(x_1, \dots, x_l) \in U_i$. In this way, for every algebraically independent k-tuple $(\beta_1, \dots, \beta_k)$ sufficiently close to $(\alpha_1, \dots, \alpha_k)$, we have by elementary field theory an isomorphism $\varphi\colon K \to K' = \mathbb{Q}(\beta_1, \dots, \beta_k, f_1(\vec{\beta}), \dots, f_l(\vec{\beta}))$ which naturally induces a isomorphism of groups $\psi\colon \Gamma \to \Gamma' \subset \mathrm{PSL}_2(K')$.

In order to assure that Γ' is a cocompact Fuchsian group we use corollary 2.3.4. Let $a \in K$, by field theory, we know that $\varphi(a)$ is a linear combination over $\mathbb{Q}(\beta_1, \dots, \beta_k)$ of products and powers of $f_1(\vec{\beta}), \dots, f_l(\vec{\beta})$, therefore $\varphi(a)$ is continuous relative to $(\beta_1, \dots, \beta_k)$, which means that by taking $(\beta_1, \dots, \beta_k)$ sufficiently close to $(\alpha_1, \dots, \alpha_k)$ we can make sure that a and $\varphi(a)$ are arbitrarily close. Now, by doing this with every coefficient of each γ_i, we can make γ_i and $\psi(\gamma_i)$ arbitrarily close for all $i = 1, \dots, n$. Therefore, by corollary 2.3.4, Γ' is a cocompact Fuchsian group for $(\beta_1, \dots, \beta_k)$ sufficiently close to $(\alpha_1, \dots, \alpha_n)$.

Finally, we just need to make $\beta_1, \dots, \beta_k$ computable numbers. Let p_i be the i-th prime number. By the Lindemann-Weierstrass theorem (see [2, Theorem 1.4]), $(e^{\sqrt{p_1}}, \dots, e^{\sqrt{p_k}})$ are algebraically independent and clearly computable. Then, for each $i = 1, \dots, n$, take $\beta_i = q_i e^{\sqrt{p_i}}$ with q_i a rational number such that $q_i e^{\sqrt{p_i}}$ is sufficiently close to α_i. This makes K' a subfield of the field of computable real numbers, therefore, the coefficients of each element of Γ' are computable numbers. ☐

Corollary 2.3.6. *Every cocompact Fuchsian group is computably quasi-isometric to $\mathbb{H}$.*

Proof. Let Γ and Γ' be as in proposition 2.3.5, and $\varphi\colon \Gamma \to \Gamma'$ an isomorphism. Then, the natural action of Γ' on $\mathbb{H}$ induces an action of Γ by $(g, x) \mapsto \varphi(g)(x)$, $g \in \Gamma, x \in \mathbb{H}$. Since the elements of Γ' consist of computable real numbers, the action is by computable isometries of $\mathbb{H}$. Therefore, all the hypothesis of theorem 2.3.2 are satisfied and thus, Γ is computably quasi-isometric to $\mathbb{H}$. ☐

Corollary 2.3.7. *Let Γ, Γ' be cocompact Fuchsian groups, then $\Gamma \sim_{CQI} \Gamma'$.*

Proof. This is a direct consequence of the previous corollary and corollary 2.2.6. ☐

3 Medvedev degrees of SFTs over Fuchsian groups

The Medvedev degrees are a way of measuring the computational complexity of subsets of the Baire space $\mathbb{N}^{\mathbb{N}}$ analogous to the Turing degrees of subsets of $\mathbb{N}$. This tool is easily applicable to the study of symbolic dynamical systems, since a subshift is essentially a subspace of the Cantor space. The preprint [3] follows this line of research and computes the Medvedev degrees of G-subshifts for various classes of groups. In this chapter we will introduce the basic framework for this study and then we will see how corollary 2.3.7 can be applied to it.

3.1 Symbolic dynamics

Symbolic dynamical systems originated as a way to simplify the study of continuous dynamical systems, for example systems of differential equations modelling physical phenomena. This technique was first used by Jacques Hadamard in 1898 while studying geodesics on surfaces of negative curvature, see his paper [14]. The idea is to discretize time as well as the phase space, so the passage of time is modelled by iterates of a single transformation and the phase space is divided into a finite number of pieces, each one of them named with a "symbol". Then, an instantaneous state of the system is given by an infinite sequence of these symbols and the "tick" of time is just shifting this sequence one cell to the left. A reference for the classical theory is [20].

These systems gained considerable interest of their own and found multiple applications in computer science and other areas of mathematics. In this chapter we are interested in their relation with group theory. We can generalize the classical context by changing the underlying group from $\mathbb{Z}$ to any group G. Then the properties of these systems reflect properties of the group G and vice-versa.

Definition 3.1.1. Let A be a finite set, which we will call the alphabet, and let G be a group. The *full G-shift* is the set $A^G = \{x\colon G \to A\}$ equipped with the prodiscrete topology and with the left *shift* action $G \curvearrowright A^G$ by left multiplication given by

$$(gx)(h) = x(g^{-1}h) \quad \text{for every } g, h \in G \text{ and } x \in A^G.$$

The elements $x \in A^G$ are called *configurations*. For a finite set $F \subset G$, a *pattern* with support F is an element $p \in A^F$. We denote the cylinder generated by p by $U_p = \{x \in A^G : x|_F = p\}$ and note that the cylinders are a clopen base for the prodiscrete topology on A^G.

Definition 3.1.2. A *G-subshift* is a G-invariant and closed subset $X \subset A^G$. Equivalently, X is a G-subshift if there exists a set $\mathcal{F}$ of forbidden patterns such that

$$X = X_{\mathcal{F}} = \{x \in A^G : gx \notin U_p \text{ for every } g \in G, p \in \mathcal{F}\}.$$

Let $X \subseteq A^G$ and $Y \subseteq B^G$ be two subshifts. A map $\phi\colon X \to Y$ is called a *morphism* if it is continuous and G-equivariant. A morphism $\phi\colon X \to Y$ is a *topological factor map* if it is surjective and a *topological conjugacy* if it is bijective.

Theorem 3.1.3 (Curtis-Hedlund-Lyndon)**.** *A map $\phi\colon X \to Y$ is a morphism if and only if there is a finite set $F \subset G$ and $\Phi\colon A^F \to B$ such that $\phi(x)(g) = \Phi((g^{-1}x)|_F)$ for every $x \in X, g \in G$.*

Proof. The proof may be found in [6, Theorem 1.8.1]. □

Definition 3.1.4. A subshift X is of *finite type* (SFT) if there exists a finite set $\mathcal{F}$ of forbidden patterns for which $X = X_{\mathcal{F}}$. A subshift Y is *sofic* if there exists an SFT X and a topological factor map $\phi\colon X \to Y$.

We now define computability notions on subshifts. From now on G will always denote an infinite finitely generated group with decidable word problem. By theorem 1.2.8 there exists a computably unique bijective coding $\nu\colon \mathbb{N} \to G$ that makes the group operation computable. This coding induces a natural representation (as in definition 2.1.4) for the space $\mathbb{N}^G$:

$$\begin{aligned} \delta\colon \mathbb{N}^{\mathbb{N}} &\to \mathbb{N}^G \\ x &\mapsto (x_{\nu^{-1}(g)})_{g \in G}, \end{aligned}$$

so we may speak of computable functions on $\mathbb{N}^G$.

Definition 3.1.5. A partial function $F :\subseteq \mathbb{N}^G \to \mathbb{N}^G$ is said to be computable if the functional $(\delta^{-1} \circ F \circ \delta) :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ is computable.

Particularly, if A is a finite alphabet, we may always assume that $A \subseteq \mathbb{N}$, so we regard a subshift $X \subseteq A^G$ as a compact subspace of $\mathbb{N}^\mathbb{N}$.

It is easy to see that theorem 1.2.8 and example 1.1.35 imply that this definition does not depend on the particular coding $\nu \colon \mathbb{N} \to G$ we fix in the beginning, as long as it is bijective and makes the group operation computable.

We have the following important observation.

Proposition 3.1.6. *Let $X \subseteq A^G$ and $Y \subseteq B^G$ be G-subshifts and $\phi \colon X \to Y$ a morphism. Then ϕ is computable. In particular, two conjugate G-subshifts are computably homeomorphic.*

Proof. By the Curtis-Hedlund-Lyndon theorem 3.1.3, there is a finite set $F \subset G$ and $\Phi \colon A^F \to B$ such that $\phi(x)(g) = \Phi((g^{-1}x)|_F)$ for every $x \in X, g \in G$. Since the group operations are computable and Φ is a finite function, this clearly makes ϕ computable. □

3.2 Medvedev degrees

For comparison, Medvedev reducibility is a generalization of Turing reducibility, see the remarks following definition 1.1.27.

Definition 3.2.1. Let $P, Q \subseteq \mathbb{N}^\mathbb{N}$. P is *Medvedev reducible* to Q (written $P \leq Q$) if there exists a partial computable functional $F :\subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$ such that $Q \subseteq \mathrm{dom}(F)$ and $F(Q) \subseteq P$.

Intuitively, one regards $P \subseteq \mathbb{N}^\mathbb{N}$ as a problem and each $x \in P$ as a solution to the problem. Then, $P \leq Q$ means there is a uniformly computable method to get a solution to P from any solution to Q.

As in Turing reducibility, we say that P is *Medvedev equivalent* to Q (written $P \equiv Q$) if $P \leq Q$ and $Q \leq P$. The equivalence class of P is called the *Medvedev degree* of P and we denote it by $m(P)$. The set of all Medvedev degrees $\mathfrak{M}$ has a partial order well-defined by $m(P) \leq m(Q)$ iff $P \leq Q$.

For example, $m(\emptyset) \geq m(Q)$ for all Q, since any functional F satistfies $F(\emptyset) \subseteq P$. If P contains a computable element $x \in P$, then $m(P) \leq m(Q)$ for all Q, since we can take the computable functional that always outputs x. Thus, we can see that the partial order $(\mathfrak{M}, \leq)$ has a greatest degree (denoted by $1_\mathfrak{M}$) given by the empty set, and a least degree (denoted by $0_\mathfrak{M}$) given by any set with a computable element. Moreover, $\mathfrak{M}$ possesses some additional structure of a *lattice*, which we now describe.

Given $x, y \in \mathbb{N}^\mathbb{N}$, denote by $x \oplus y$ the element of $\mathbb{N}^\mathbb{N}$ defined by

$$(x \oplus y)(2n) = x(n),$$
$$(x \oplus y)(2n + 1) = y(n),$$

for all $n \in \mathbb{N}$. Similarly, for strings $\sigma, \tau \in \mathbb{N}^{<\mathbb{N}}$ with $l(\sigma) = l(\tau) = s$ denote $\sigma \oplus \tau$ the string of length $2s$ defined by

$$(\sigma \oplus \tau)(2n) = \sigma(n),$$
$$(\sigma \oplus \tau)(2n + 1) = \tau(n),$$

for all $n < s$. It is clear that $\oplus$ is a computable operation.

Given $P, Q \subseteq \mathbb{N}^\mathbb{N}$, we write

$$P \vee Q = \{x \oplus y : x \in P \text{ and } y \in Q\},$$
$$P \wedge Q = \{0x : x \in P\} \cup \{1x : x \in Q\},$$

where $0x$ denotes the sequence that starts with 0 and then continues as x, similarly for $1x$.

Proposition 3.2.2. *$(\mathfrak{M}, \leq)$ is a distributive lattice, that is to say, for every $P, Q, R \subseteq \mathbb{N}^\mathbb{N}$,*

1. *$\{m(P), m(Q)\}$ has a least upper bound (join) given by $m(P) \vee m(Q) = m(P \vee Q)$,*

2. *$\{m(P), m(Q)\}$ has a greatest lower bound (meet) given by $m(P) \wedge m(Q) = m(P \wedge Q)$,*

3. *$m(P) \vee (m(Q) \wedge m(R)) = (m(P) \vee m(Q)) \wedge (m(P) \vee m(R))$,*

4. $m(P) \wedge (m(Q) \vee m(R)) = (m(P) \wedge m(Q)) \vee (m(P) \wedge m(R))$.

Proof. It is direct from the definitions, we will prove (1) as an example.

If $x \in P \vee Q$, then the elements $y, z \in \mathbb{N}^{\mathbb{N}}$ defined by $y(n) = x(2n)$ and $z(n) = x(2n+1)$, for all $n \in \mathbb{N}$, belong to P and Q respectively. Moreover we may uniformly compute them from x, so $P \leq P \vee Q$ and $Q \leq P \vee Q$.

If $R \subseteq \mathbb{N}^{\mathbb{N}}$ is such that $P, Q \leq R$, then there exist computable functionals F and G such that $F(R) \subseteq P$ and $G(R) \subseteq Q$. Define $H\colon \operatorname{dom}(F) \cap \operatorname{dom}(G) \to \mathbb{N}^{\mathbb{N}}$ by $H(x) = F(x) \oplus G(x)$ for all $n \in \mathbb{N}$, then, clearly $H(R) \subseteq P \vee Q$ and H is computable, so $P \vee Q \leq R$. □

Intuitively, $m(P) \vee m(Q)$ is the difficulty of solving P and Q simultaneously, and $m(P) \wedge m(Q)$ is the difficulty of solving at least one of them. The empty set represents the problem with no solutions, so it has the greatest degree of difficulty. Lastly, a problem with a computable solution is solvable by a regular Turing machine, so we regard it as "easy".

The Medvedev degrees of Π^0_1 subsets of $2^{\mathbb{N}}$ (see definition 1.1.29) will be the most relevant in this work. We denote

$$\mathcal{P} = \{m(P) : P \subseteq 2^{\mathbb{N}} \text{ is a nonempty } \Pi^0_1 \text{ set}\}.$$

Proposition 3.2.3. *$\mathcal{P}$ is a sublattice of $\mathfrak{M}$.*

Proof. Let $P, Q \subseteq 2^{\mathbb{N}}$ be nonempty Π^0_1 sets, then obviously $P \vee Q \in 2^{\mathbb{N}}$ and $P \wedge Q \in 2^{\mathbb{N}}$ are nonempty. We show that they are Π^0_1. By hypothesis, there are c.e. sets $I, J \subseteq \mathbb{N}^{<\mathbb{N}}$ such that

$$P = \bigcap_{\sigma \in I} (\mathbb{N}^{\mathbb{N}} \setminus U_\sigma) \text{ and } Q = \bigcap_{\tau \in J} (\mathbb{N}^{\mathbb{N}} \setminus U_\tau).$$

Define

$$K = \{\sigma \oplus \tau : \sigma, \tau \in \mathbb{N}^{<\mathbb{N}},\, l(\sigma) = l(\tau) \text{ and } (\sigma \in I \text{ or } \tau \in J)\},$$

then K is c.e. because I, J are c.e and $\oplus$ is computable. Moreover, $P \vee Q = \bigcap_{\rho \in K} (\mathbb{N}^{\mathbb{N}} \setminus U_\rho)$. Indeed, if $z \in P \vee Q$, then $z = x \oplus y$ for $x \in P$ and $y \in Q$. From this we have that $\sigma \oplus \tau \prec z$ implies $\sigma \prec x$ and $\tau \prec y$ for any σ, τ. In particular, $\rho \not\prec z$ for any $\rho \in K$. Conversely, if $z \notin P \vee Q$, then $z = x \oplus y$ where $x \notin P$ or $y \notin Q$. Suppose $x \notin P$, then $\sigma \prec x$ for some $\sigma \in I$. Take $\tau = y|l(\sigma)$. We have $\sigma \oplus \tau \prec z$ and $\sigma \oplus \tau \in K$. Similarly if $y \notin Q$. This shows that $P \vee Q$ is Π^0_1.

Define

$$L = \{0\sigma : \sigma \in I\} \cup \{1\tau : \tau \in J\},$$

then L is easily seen to be c.e. and as before we have $P \wedge Q = \bigcap_{\rho \in L} (\mathbb{N}^{\mathbb{N}} \setminus U_\rho)$, which shows that $P \wedge Q$ is Π^0_1. □

Of course, $0_{\mathfrak{M}} \in \mathcal{P}$ since a singleton $\{x\}$ with x computable is a Π^0_1 set. It turns out that $\mathcal{P}$ also has a greatest element but this is much harder to prove and it will not be relevant for us. See [15] for more details.

We now make the connection with subshifts. Let G be an infinite finitely generated group with solvable word problem and let $\delta\colon \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^G$ be the representation as in definition 3.1.5.

Definition 3.2.4. We define the Medvedev degree of a subshift $X \subseteq A^G$ as $m(X) = m(\delta^{-1}(X))$.

Again, this definition does not depend on the particular coding for G. What makes Medvedev degrees interesting in this context is that they are a conjugacy invariant.

Proposition 3.2.5. *Let $X \subseteq A^G$ and $Y \subseteq B^G$ be G-subshifts and $\phi\colon X \to Y$ a morphism. Then $m(X) \geq m(Y)$. In particular, two conjugate G-subshifts have the same Medvedev degree.*

Proof. By proposition 3.1.6, ϕ is computable, thus $m(X) \geq m(Y)$ by definition. □

Let G be an infinite finitely generated group with solvable word problem. We denote

$$\mathfrak{M}_{\mathtt{SFT}}(G) = \{m(X) : X \text{ is a nonempty } G\text{-SFT}\}.$$

Proposition 3.2.6. *$\mathfrak{M}_{\mathtt{SFT}}(G) \subseteq \mathcal{P}$ and it is an invariant of group isomorphism.*

Proof. Let $X \subseteq A^G$ be a nonempty G-SFT, we show that it is computably homeomorphic to a Π_1^0 set $Y \subseteq 2^{\mathbb{N}}$. By definition (see section 3.1), there exists a finite set $\mathcal{F}$ of patterns, such that $X = X_{\mathcal{F}}$, i.e.,

$$X = \bigcap_{g \in G} \bigcap_{p \in \mathcal{F}} (A^G \setminus g^{-1}(U_p)).$$

Note that, identifying G with $\mathbb{N}$, a pattern $p \in \mathcal{F}$ is just a function $p \colon F \subset \mathbb{N} \to A$, where F is finite, so it is almost a word over A. Since A and $\mathcal{F}$ are finite, it is easy to construct another finite set $\mathcal{F}'$ of patterns such that $X = X_{\mathcal{F}'}$ and every $p \in \mathcal{F}'$ is a word of fixed length $p \colon s \to A$. Let $C = \bigcap_{p \in \mathcal{F}'} (A^G \setminus U_p)$, then, since every finite set is c.e., C is Π_1^0. We have

$$X = \bigcap_{g \in G} g^{-1}(C)$$

and it is not hard to see that, since the group operation is computable, the family $\{g^{-1}(C)\}_{g \in G}$ is uniformly closed. Thus, by proposition 1.1.31, X is Π_1^0. Recall from example 1.1.36 that we have a computable homeomorphism $F \colon A^{\mathbb{N}} \to 2^{\mathbb{N}}$, so X is computably homeomorphic to $F(X) \subseteq 2^{\mathbb{N}}$ which is Π_1^0. Therefore $m(X) \in \mathcal{P}$.

We now prove that $\mathfrak{M}_{\mathsf{SFT}}(G)$ is an invariant under group isomorphism. Let G and H be f.g. groups with solvable word problem and $\varphi \colon G \to H$ an isomorphism. By corollary 1.2.9, φ is computable, so it induces a computable homeomorphism $\Phi \colon \mathbb{N}^G \to \mathbb{N}^H$. Thus, if X is a non-empty G-SFT, then $\Phi(X)$ is a non-empty H-SFT computably homeomorphic to X, particularly with the same Medvedev degree. We conclude $\mathfrak{M}_{\mathsf{SFT}}(G) \subseteq \mathfrak{M}_{\mathsf{SFT}}(H)$. Similarly $\mathfrak{M}_{\mathsf{SFT}}(H) \subseteq \mathfrak{M}_{\mathsf{SFT}}(G)$. □

As an example, we have the following result regarding $\mathbb{Z}^2$, observed by Simpson in [27].

Theorem 3.2.7. *$\mathfrak{M}_{\mathsf{SFT}}(\mathbb{Z}^2) = \mathcal{P}$.*

The preprint [3] studies how $\mathfrak{M}_{\mathsf{SFT}}(G)$ behaves under various relations in group theory, as in subgroups, quotients, conmensurability, translation-like actions and quasi-isometries. The authors proved the following theorem.

Theorem 3.2.8. *Let G and H be two computably quasi-isometric finitely presented groups with solvable word problem. Then $\mathfrak{M}_{\mathsf{SFT}}(G) = \mathfrak{M}_{\mathsf{SFT}}(H)$.*

The details of the proof are too long and cumbersome to include here. The idea is to construct an SFT QI over G whose elements encode all quasi-isometries $f \colon H \to G$, moreover, the computable points of QI correspond to the computable quasi-isometries from H to G. Then, given any SFT $X \subseteq A^H$, one can enrich QI to get a G-SFT $QI[X]$ such that $m(QI[X]) = m(QI) \vee m(X)$. Since G and H are computably quasi-isometric, we have $m(QI) = 0_{\mathfrak{M}}$, so $m(QI[X]) = m(X)$. Thus $\mathfrak{M}_{\mathsf{SFT}}(H) \subseteq \mathfrak{M}_{\mathsf{SFT}}(G)$. See also [8].

Corollary 3.2.9. *Let G, H be cocompact Fuchsian groups, then $\mathfrak{M}_{\mathsf{SFT}}(G) = \mathfrak{M}_{\mathsf{SFT}}(H)$.*

Proof. Direct from the previous theorem and corollary 2.3.7. □

4 Further questions

In section 2.2 we showed the existence of a pair of computable metric spaces $\mathbb{N}$ and T such that $\mathbb{N}$ quasi-isometrically embeds in T but not computably. We note that this does not directly imply the existence of a pair of computable metric spaces that are quasi-isometric but not computably, because the image of $\mathbb{N}$ in T is not a computable subset of T.

Question 4.1. Does there exist a pair of computable metric spaces X,Y such that $X \sim_{QI} Y$ but not $X \sim_{CQI} Y$?

We conjecture a positive answer for this question. However, more sophisticated techniques are probably needed, such as priority constructions from computability theory. Furthermore, it should be possible to define a space of "computable ends" for arbitrary computable metric spaces which is invariant under computable quasi-isometries.

In the spirit of [23], we say that a computable metric space X is computably quasi-isometrically categorical if $X \sim_{QI} Y$ implies $X \sim_{CQI} Y$ for all computable metric spaces Y.

Question 4.2. Does there exist a computably quasi-isometrically categorical metric space?

Concerning groups, recall from corollary 1.2.9 that finitely generated groups are computably categorical, it is natural to ask if the same happens with quasi-isometries.

Question 4.3. Does there exist a pair G, H of finitely generated groups with solvable word problem such that $G \sim_{QI} H$ but not $G \sim_{CQI} H$?

A positive answer to this question would be very interesting, since it would mean that groups have a finer computable large-scale geometry worth studying.

The techniques used in section 2.3 seem easy to generalize to other discrete subgroups of matrix groups, such as Kleinian groups. Moreover, it seems possible to develop a theory of computable discrete subgroups of computable topological groups, which could possibly answer question 4.3.

Finally, in section 3.2 we showed that SFTs over cocompact Fuchsian groups share the same class of Medvedev degrees, however this does not give any explicit description of it. We claim that, using similar techniques to the ones used for the group $\mathbb{Z}^2$, it is possible to show that this class is in fact the class of all Π_1^0 Medvedev degrees. This is ongoing work with Barbieri and Carrasco-Vargas that will appear in an upcoming article.

References

[1] C. Ash and J. Knight. *Computable Structures and the Hyperarithmetical Hierarchy*. Studies in Logic and the Foundations of Mathematics. Elsevier Science, 2000.

[2] A. Baker. *Transcendental number theory*. Cambridge Mathematical Library. Cambridge University Press, Cambridge, 2022. With an introduction by David Masser, Reprint of the 1975 original [0422171].

[3] S. Barbieri and N. Carrasco-Vargas. Medvedev degrees of subshifts on groups. *arXiv:2406.12777*, 2024.

[4] S. Barbieri, N. Carrasco-Vargas, and C. Rojas. Effective dynamical systems beyond dimension zero and factors of sfts. *Ergodic Theory and Dynamical Systems*, 45(5):1329–1369, 2025.

[5] V. Brattka and P. Hertling. *Handbook of Computability and Complexity in Analysis*. Theory and Applications of Computability. Springer International Publishing, 2021.

[6] T. Ceccherini-Silberstein and M. Coornaert. *Cellular Automata and Groups*. Springer Monographs in Mathematics. Springer Berlin Heidelberg, 2010.

[7] K. F. Celine, Z. Gao, S. Jain, R. Lou, F. Stephan, and G. Wu. Quasi-isometric reductions between infinite strings. *arXiv:2407.14105*, 2025.

[8] D. B. Cohen. The large scale geometry of strongly aperiodic subshifts of finite type. *Advances in Mathematics*, 308:599–626, 2017.

[9] R. Downey and A. Melnikov. Computable structure theory: A unified approach. Draft retrieved from: https://homepages.ecs.vuw.ac.nz/~melnikal/maindoc.pdf.

[10] T. Dymarz. Bilipschitz equivalence is not equivalent to quasi-isometric equivalence for finitely generated groups. *Duke Mathematical Journal*, 154(3):509–526, 2010.

[11] M. Einsiedler and T. Ward. *Ergodic theory with a view towards number theory*, volume 259 of *Graduate Texts in Mathematics*. Springer-Verlag London, Ltd., London, 2011.

[12] V. Gregoriades, T. Kispéter, and A. Pauly. A comparison of concepts from computable analysis and effective descriptive set theory. *Mathematical Structures in Computer Science*, 27(8):1414–1436, 2017.

[13] M. Gromov. Hyperbolic groups. In *Essays in Group Theory*, volume 8 of *Mathematical Sciences Research Institute Publications*, pages 75–263. Springer, 1987.

[14] J. Hadamard. Les surfaces à courbures opposées et leurs lignes géodésiques. *Journal de Mathématiques Pures et Appliquées*, 4:27–74, 1898.

[15] P. G. Hinman. A survey of Mučnik and Medvedev degrees. *The Bulletin of Symbolic Logic*, 18(2):161–229, 2012.

[16] S. Katok. *Fuchsian Groups*. Chicago Lectures in Mathematics. University of Chicago Press, 1992.

[17] B. Khoussainov and T. Takisaka. Infinite strings and their large scale properties. *The Journal of Symbolic Logic*, 87(2):585–625, 2022.

[18] J. Lee. *Introduction to Smooth Manifolds*. Graduate Texts in Mathematics. Springer, 2003.

[19] H. Lewis and C. Papadimitriou. *Elements of the Theory of Computation*. Prentice-Hall International editions. Prentice-Hall, 1998.

[20] D. Lind and B. Marcus. *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, 1995.

[21] C. Löh. *Geometric Group Theory: An Introduction*. Universitext. Springer International Publishing, 2017.

[22] R. Lyndon and P. Schupp. *Combinatorial Group Theory.* Classics in Mathematics. Springer Berlin Heidelberg, 2001.

[23] A. G. Melnikov. Computably isometric spaces. *The Journal of Symbolic Logic*, 78(4):1055–1085, 2013.

[24] A. Montalbán. *Computable Structure Theory: Within the Arithmetic.* Perspectives in Logic. Cambridge University Press, 2021.

[25] J. Rotman. *An Introduction to the Theory of Groups.* Graduate Texts in Mathematics. Springer New York, 1999.

[26] G. Shimura. *Introduction to the Arithmetic Theory of Automorphic Functions.* Kanô memorial lectures. Princeton University Press, 1971.

[27] S. G. Simpson. Medvedev degrees of two-dimensional subshifts of finite type. *Ergodic Theory and Dynamical Systems*, 34(2):679–688, 2012.

[28] R. Soare. *Recursively Enumerable Sets and Degrees: A Study of Computable Functions and Computably Generated Sets.* Perspectives in Mathematical Logic. Springer Berlin Heidelberg, 1999.

[29] A. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42(1):230–265, 1936.

[30] K. Weihrauch. *Computable Analysis: An Introduction.* Texts in Theoretical Computer Science. An EATCS Series. Springer Berlin Heidelberg, 2000.

[31] A. Weil. On discrete subgroups of lie groups. *Annals of Mathematics*, 72(2):369–384, 1960.