UNIVERSITY OF TAMPERE
Master's Thesis

Kirsti Laurila

On Recurrence Time

Department of mathematics, statistics and philosophy
Mathematics
February 2005

Tampereen yliopisto
Matematiikan, tilastotieteen ja filosofian laitos

LAURILA, KIRSTI: Rekursioajasta
Pro gradu -tutkielma, 59s
Matematiikka
Helmikuu 2005

Tiivistelmä

Tiedon määrä maailmassa lisääntyy koko ajan, minkä vuoksi tiedon tiivistämiseen tarvitaan tehokkaita menetelmiä. Menetelmissä käytetään monia erilaisia algoritmeja, joista niin sanotut Lempel-Ziv algoritmit liittyvät läheisesti merkkijonon rekursioaikaan. Merkkijonon rekursioaika on merkkijonon ensimmäisen ja toisen esiintymän välissä olevien merkkien määrä lähtien merkkijonon alusta. Rekursioajalla on monia matemaattisia ominaisuuksia, joita tutkielmassa tarkastellaan. Erityisesti todistetaan Rekursioaika-lause, jonka perusteella rekursioaikaa voidaan käyttää tehokkaana apuna tiedon tiivistämisessä.

Kun tietoa tiivistetään, apuna on erilaisia koodeja, joten tutkielmassa tarkastellaan myös koodien ominaisuuksia ja koodien käyttämistä eri yhteyksissä. Lisäksi näiden ominaisuuksien tutkimisessa kokonaislukuvälien pakkaukset ovat tärkeitä apuvälineitä. Kokonaislukuvälin pakkaus on pakattavalla välillä olevien lukujen tarpeeksi suuri joukko.

Erityinen tiedon tiivistämisen sovellusala ovat biologiset merkkijonot, muun muassa DNA-sekvenssit. Tämän vuoksi tutkielmassa etsitään kokeellisesti rekursioaikoja DNA:lle käyttäen aineistona ihmisen kromosomi 22:n DNA-sekvenssiä. Lisäksi saatujen lauseiden perusteella lasketaan arvioita DNA-sekvenssien rekursioajoille. Lopuksi kokeellisia rekursioaikoja verrataan laskemalla saatuihin rekursioaikoihin ja huomataan näiden välillä vastaavuus.

University of Tampere
Department of mathematics, statistics and philosophy

LAURILA, KIRSTI: On Recurrence Time
Master's thesis, 59 pages
Mathematics
February 2005

Abstract

The amount of the data in the world enlarges all the time and therefore efficient methods are needed for data compression. There are many different algorithms to compress the data. One class of compression algorithms are the Lempel-Ziv algorithms that are closely connected to the recurrence time of the sequence. The recurrence time of the sequence is the number of the characters between the start at the sequence and its following occurrence. Recurrence time has many mathematical properties which are examined in the thesis. Especially the Recurrence time theorem is proved. This theorem gives the basis to use recurrence time as an efficient help in the data compression.

When compressing the data different codes are used. This is why the properties of the codes and the using the codes in different cases are also studied. Furthermore, to study these properties, the packings of intervals of integers are important tools. The packing of a interval of integers is a big enoug set of numbers inside the interval..

The special application field of data compression is biological sequences, among other things, DNA sequences. Thus in the thesis recurrence times of DNA-sequences are experimentally studied using the human chromosome 22 as a DNA-sequence. Besides, the recurrence times of DNA-sequences are estimated on the basis of the theorems proved in the thesis. Finally, the experimental recurrence times are compared with the calculated ones and in general, a good agreement is found.

Preface

This Master's thesis has been carried out in the Institute of Signal Processing at Tampere University of Technology. I am deeply grateful to my supervisor, Professor Ioan Tabus, for providing me this great opportunity of working with the subject which I am interested in, and for helping me in many ways during the project.

I want also to thank Professor Lauri Hella, the examiner of the thesis, for advantageous comments.

Further, I am obliged to Jukka Ilmonen for useful notes and to Aatu Koskensilta for remarks of the text and especially for correcting my English. My thanks also go to Marko Kanerva for borrowing his computer and to Kukka-Maaria Kemppainen for reading the Finnish abstract.

Above all, I wish to thank Leo, Pingu, my family and my friends for understanding and particularly for being part of my life. Without you there would be nothing!

"Black holes are God's way of dividing by zero."
-Unknown-

Tampere February 2004

Kirsti Laurila

Contents

1 Introduction

We were inspired to the idea of this thesis by Aaron D. Wyner's, Jakob Ziv's and Abraham J. Wyner's paper "On the Role of Pattern Matching in Information Theory" [23]. The paper is about pattern matching and data compression and it collects together some useful theorems which concern, among other things, the recurrence time which is the subject of the thesis. The paper introduces, for instance, Kac's lemma and the Recurrence time theorem both of which we prove in the thesis. We were not interested only in mathematical theorems but also in their applications, especially on biological sequences. Thus the thesis contains also a short part where we have applied the studied theorems with DNA sequences.

The structure of the thesis is the following.

First, in Chapter 2 we introduce certain definitions and theorems which are needed in the thesis. We have divided the chapter into four sections the most central of which are Sections 2.2 and 2.3. They cover mathematical statistics and information theory. Especially important is the concept of entropy in Chapter 2.3.

Secondly, in Chapter 3 we start the proper theme of the thesis by studying codes and codings. Inter alia, we give examples of codes, we prove a famous theorem used in coding theory, Kraft's inequality, and we also prove that there are good, but not too good, codes.

Chapter 4 is about packings. Although this chapter is not as important as the previous one, it contains many interesting results such as Packing lemma. This lemma we need in Chapter 5.

Next chapter (Chapter 5) is the main chapter of the thesis. There we introduce the concept of recurrence time which is the time it takes for a sequence to reappear in a longer sequence. In this chapter we prove the Recurrence

time theorem (theorem 5.1). In the proof we need several theorems proven earlier in the thesis. In the end of the chapter we also prove some other theorems in which recurrence time plays an important role.

After Chapters 2-5 which contain the theoretical part of the thesis we come to Chapter 6 where we apply the Kac's lemma which we have presented in Chapter 5. We apply this lemma to a DNA sequence, the human chromosome 22.

Finally, in Chapter 7 we summarize the thesis and discuss some further topics related to recurrence time.

In the thesis we have widely used Paul C. Shield's book "Ergodic Theory and Discrete Sample Paths" as a source, but also several other books, articles and papers have been consulted. We have tried to present the proofs of the thesis in detail so that the steps follow each other clearly. In some proofs we have only "written them open" whilst some proofs differ from that of the source significantly. Some superfluous details we have omitted, too. There are also some examples, especially in Chapters 3 and 4, but in Chapter 5 we considered it useless to include them since they could not be simple enough to clarify the theorems of the chapter. We hope that Chapter 6 helps to understand the importance and usefulness of the theorems of Chapter 5 in different kinds of applications, too.

We assume the reader knows mathematics and especially analysis and statistics. In individual proofs some concepts of graph-theory are also required. Knowledge of measure theory may help in understanding the proofs, too. However, it is possible to understand the thesis with minor knowledge in mathematics if the proofs are skipped.

2 Preliminaries

In this chapter we give some definitions and theorems which are needed later. We have confined ourselves to just giving references to the proofs because some proofs are very long, and are not essential for the thesis.

2.1 General

First we define some basic concepts of analysis, such as limes supremum and the Landau symbols which are needed through the thesis. We also give a definition of the L^1-norm. The source of this section is [18].

Definition 2.1. *Let $a_1, a_2, \ldots$ be a sequence of numbers.*

a) ***Limes supremum*** *of $a_1, a_2, \ldots$, which is denoted by $\limsup_{i\to\infty} a_i$, is*

$$\limsup_{i\to\infty} a_i = \lim_{i\to\infty} A_i, \quad \text{where } A_i = \sup_{k>i} a_k \text{ (if such exists).}$$

b) ***Limes infimum*** *of $a_1, a_2, \ldots$, which is denoted by $\liminf_{i\to\infty} a_i$, is*

$$\liminf_{i\to\infty} a_i = \lim_{i\to\infty} A_i, \quad \text{where } A_i = \inf_{k>i} a_k \text{ (if such exists).}$$

Definition 2.2. *Let f and g be functions. We define the Landau symbols O and o by setting:*

a) f is ***Big o*** *of g if there exists positive constants $C \in \mathbb{R}$ and $n_0 \in \mathbb{R}$ such that $|f(n)| \leq C|g(n)|$, for all $n \geq n_0$. We write this as $f(n) = O(g(n))$.*

b) f is ***little o*** *of g if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$. This we write as $f(n) = o(g(n))$.*

Definition 2.3. *Let* $\boldsymbol{x}$ *be a vector of length* n. ***The*** $\boldsymbol{L^1}$***-norm*** *of* $\boldsymbol{x}$ *is*

$$\|\boldsymbol{x}\|_1 = \sum_{i=1}^{n} |x_i|.$$

Definition 2.4. *Let* $\|\cdot\|$ *be a norm. We say that* $\boldsymbol{x} = x_1, x_2, \ldots$ ***converges in norm*** *to* b *if*

$$\lim_{n\to\infty} \|x_n - b\| = 0.$$

2.2 Statistics

In this section we define some basic concepts and give theorems of statistics and sequences. Among other things, we give the definitions of a measure, a probability space, a stochastic process and a source sequence. We also define stationarity and ergodicity which are very important in the thesis since in many theorems the source sequence is assumed to have these properties. The main sources of this subchapter are [11], [15], [16] and [17].

Definition 2.5. *Let* Ω *be a set. A nonempty collection* Σ *of subsets of* Ω *is* ***a*** $\boldsymbol{\sigma}$***-algebra*** *if the following three conditions are satisfied:*

1. $\Omega \in \Sigma$.
2. *If* $S \in \Sigma$, *then* $\bar{S} \in \Sigma$. *(Here* $\bar{S} = \{\omega \in \Omega : \omega \notin S\}$ *is the complement of* S*.)*
3. *If* $S_i \in \Sigma$ *for all* $i \in \mathbb{Z}_+$, *then* $\bigcup_{i=1}^{\infty} S_i \in \Sigma$.

The smallest σ*-algebra containing the set* S *is called the* σ*-algebra* ***generated by*** S *and it is denoted by* $\sigma(S)$.

Definition 2.6. ***A Borel*** $\boldsymbol{\sigma}$***-algebra*** *is a* σ*-algebra generated by a collection of open or closed sets in a topology.*

Definition 2.7. *Let P be a function which maps the sets of some family of sets $\mathcal{C}$ to $\bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$. We say that P is* ***σ-additive*** *if*

i) $P(\emptyset) = 0$.

ii) If when the subsets $A_1, A_2, \ldots \in \mathcal{C}$ are pairwise disjoint and $\bigcup_{i=1}^{\infty} A_i \in \mathcal{C}$, then $P(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} P(A_i)$.

Definition 2.8. *A non-negative, σ-additive function is* ***a measure****. If Σ is a Borel σ-algebra, then the measure $P : \Sigma \to \bar{\mathbb{R}}$ is* ***a Borel measure.***

Definition 2.9. ***A probability measure*** *is a measure P which is defined on a σ-algebra Σ and for which $P(\Omega) = 1$.*

Definition 2.10. *A triplet (Ω, Σ, P) is* ***a probability space*** *if Σ is the σ-algebra generated by the set Ω and if P is a probability measure.*

Definition 2.11. *Let (Ω, Σ, P) be a probability space and let X be a real-valued function on the set Ω. Function $X : \Omega \to \mathbb{R}$ is* ***a random variable*** *if when $x \in \mathbb{R}$, then $\{\omega \in \Omega : X(\omega) \leq x\} \in \Sigma$. A vector-valued function $(X_1, X_2, \ldots, X_n) : \Omega \to \mathbb{R}^n$ is* ***a random vector*** *if when $x_1, x_2, \ldots, x_n \in \mathbb{R}$, always $\{\omega \in \Omega : X_1(\omega) \leq x_1, X_2(\omega) \leq x_2, \ldots, X_n(\omega) \leq x_n\} \in \Sigma$.*

Definition 2.12. *A sequence $\{X_n\} = X_1, X_2, \ldots$ of random variables defined on a probability space (Ω, Σ, P) is* ***a (stochastic) process****. If $X_1, X_2, \ldots X_k$ is finite, then the process is* ***discrete-time****.*

Remark 1. We usually write $X = x$ instead of $\{\omega \in \Omega : X(\omega) = x\}$.

Definition 2.13. *Let $\{X_n\}$ be a process. If all random variables X_i of the process get values on $\mathcal{A}$, we call $\mathcal{A}$ as* ***an alphabet*** *of the process. The (finite) number of elements in $\mathcal{A}$ is denoted by $|\mathcal{A}|$.*

Definition 2.14. *Let $a_i \in \mathcal{A}$ for all $m \leq i \leq n$. A sequence $a_m, a_{m+1}, \ldots, a_n$ is denoted by a_m^n. The set of all sequences a_m^n is marked with $\mathcal{A}_m^n$ and if $m = 1$, with $\mathcal{A}^n$. The set of all infinite sequences $\boldsymbol{a}$ is $\mathcal{A}^\infty$. **The cylinder set** determined by a_m^n is the set $[a_m^n] = \{\boldsymbol{x} \in \mathcal{A}^\infty : x_i = a_i, m \leq i \leq n\}$.*

Remark 2. The sequence $A = a_1, a_2, \ldots, a_n$ can also be described as **a concatenation** of blocks $v(1), v(2), \ldots, v(k)$, where $v(i) = a_t, \ldots, a_{t+s}$ and $t, s \in \{1, \ldots, n\}$ so that

$$A = v(1)v(2)\ldots v(k).$$

Definition 2.15. ***A source*** *$S = \{X_n\}$ is a (discrete-time) stochastic process.*

Definition 2.16. ***A distribution*** *of a (discrete) random variable X is defined as a set of numbers with a probability function $P_X(x_j) = P(\{\omega \in \Omega : X(\omega) = x_j\})$, for $x_j \in \mathbb{R}$. The probability of an event $E \in X(\Omega)$ is $P(E) = \sum_{x_j \in E} P_X(x_j)$. **The cumulative distribution function** F_X of X is $F_X(x) = P(X \leq x)$, where $x \in \mathbb{R}$.*

Definition 2.17. *Let $\{X_n\}$ be a process. **The kth order joint distribution** of the process is a measure P_k on $\mathcal{A}^k$ defined by*

$$P_k(a_1^k) = P(X_1^k = a_1^k) = P(X_1 = a_1, X_2 = a_2, \ldots, X_k = a_k), \text{ where } a_1^k \in \mathcal{A}^k.$$

Remark 3. We frequently write $P(X = a)$ as $p(a)$ or $P(a)$, and $P(X_1^k = a_1^k)$ as $p(a_1^k)$ or $P(a_1^k)$.

Definition 2.18. *Let $\{X_k\}$ be a process (source). We say that the process (source) is **stationary** if for all m, n and a_m^n, it holds that*

$$P(X_m^n = a_m^n) = P(X_{m+1}^{n+1} = a_m^n).$$

Remark 4. If the process is stationary, then for all $m \leq i \leq n$ and $k \in \mathbb{N}$,

$$P(X_i = a_i) = P(X_{i+1} = a_i) = P(X_{(i+1)+1} = a_i) = \cdots = P(X_{i+k} = a_i).$$

In other words stationarity means that whatever is the starting point of the process, the probability law is the same.

Definition 2.19. *Let (Ω, Σ, P) be a probability space and $A \in \Sigma$ an event with positive probability.* ***The conditional probability*** *$P(\cdot\,|A) : \Sigma \to \mathbb{R}$ is a function defined as*

$$P(B \,|\, A) = \frac{P(A \cap B)}{P(A)}, \text{ where } B \in \Sigma.$$

Definition 2.20. *Let (Ω, Σ, P) be a probability space and $A \in \Sigma, \quad B \in \Sigma$ events. We say that A and B are* ***independent*** *if*

$$P(A \cap B) = P(A)P(B).$$

If random variables $X_1, X_2, \ldots, X_n$ are all independent and they have the same probability distribution, we say that $X_1, X_2, \ldots, X_n$ are ***identically independently distributed*** *and we abbreviate this i.i.d.*

Definition 2.21. *Let (Ω, Σ, P) be a probability space, X a (discrete) random variable and g a function which maps X to $\mathbb{R}$. If $g(X)$ is discrete and if $\sum_{i=1}^{\infty} |g(x_i)|P(X = x_i) < \infty$, then* ***the expected value*** *of the random variable X is*

$$E[g(X)] = \sum_{i=1}^{\infty} g(x_i)P(X = x_i).$$

Definition 2.22. *Let $X_1, X_2, \ldots$ and X be random variables in a probability space (Ω, Σ, P).*

a) If for any $\varepsilon > 0$, it holds that

$$P\left(|X_n - X| \geq \varepsilon\right) \to 0, \text{ when } n \to \infty,$$

we say that the sequence $\{X_n\}$ ***converges in probability*** *to the random variable X. We denote this by $X_n \xrightarrow{p} X$.*

b) The sequence $\{X_n\}$ ***converges almost surely*** *to the random variable* X *if*

$$P\left(\{\omega \in \Omega \,:\, X_n(\omega) \to X\left(\omega\right)\}\right) = 1, \text{ when } n \to \infty,$$

that is $P(\lim_{n\to\infty} X_n = X) = 1$. *This we write as* $X_n \xrightarrow{a.s.} X$.

Remark 5. The almost sure convergence is also called convergence with probability one.

Remark 6. Let $f_1, f_2, \ldots$ and f be measurable functions (see [18]). The following are equivalent (Cf. [17, page 11].)

a) For $\varepsilon > 0$ there exists an integer N and a set G for which $P(G) \geq 1-\varepsilon$ such that for any $x \in G$ and any $n \geq N$, it holds that $|f_n(x)-f(x)| < \varepsilon$.

b) Almost surely $f_n \to f$.

c) For every $\varepsilon > 0$, $|f_n(x) - f(x)| < \varepsilon$ eventually, almost surely.

Definition 2.23. *a)* ***The (left) shift transformation*** T *is a function* $T : \mathcal{A}^\infty \to \mathcal{A}^\infty$ *for which*

$$(T\boldsymbol{a})_n = a_{n+1}, \text{ for all } \boldsymbol{a} \in \mathcal{A}^\infty \text{ and } n \in \mathbb{Z}_+.$$

b) ***The set transformation*** $T^{-1} : \mathcal{P}(\mathcal{A}^\infty) \to \mathcal{P}(\mathcal{A}^\infty)$ *is a function for which*

$$T^{-1}B = \{\boldsymbol{a} \in \mathcal{A}^\infty \,:\, T\boldsymbol{a} \in B\}, \quad \text{where } B \subseteq \mathcal{A}^\infty.$$

Definition 2.24. *Let* $\boldsymbol{a} \in \mathcal{A}^\infty$ *and* $n \geq 1$. *The* ***coordinate function*** $\widehat{X}_n : \mathcal{A}^\infty \to \mathcal{A}$ *is*

$$\widehat{X}_n(\boldsymbol{a}) = a_n.$$

Theorem 2.1 (Kolmogorov representation theorem)**.** *Let $\{X_n\}$ be a process with a finite alphabet $\mathcal{A}$. There exists a unique Borel measure P on $\mathcal{A}^\infty$ for which the sequence of coordinate functions $\{\widehat{X}_n\}$ has the same distribution as $\{X_n\}$.*

Proof. See [17, pages 2-3]. □

Remark 7. We call the sequence of coordinate functions $\{\widehat{X}_n\}$ on a probability space $(\mathcal{A}^\infty, \Sigma, P)$ the **Kolmogorov representation of the process** $\{X_n\}$ and the measure P the **Kolmogorov measure of the process.** If there is no danger of misunderstanding the Kolmogorov measure of the process, P is called simply a process or a measure of the process.

Definition 2.25. *Let (Ω, Σ, P) be a probability space, $B \in \Sigma$ an event and $T : \Omega \to \Omega$ a shift transformation. The transformation T is **measurable** if $T^{-1}B \in \Sigma$. If also $P(T^{-1}B) = P(B)$, T is **measure-preserving**.*

Definition 2.26. *Let T be a measure-preserving transformation. The transformation T is **ergodic** if*

$$\textit{always when} \quad T^{-1}B = B, \ \textit{then } P(B) = 0 \ \textit{or } P(B) = 1.$$

If the shift transformation is ergodic in the Kolmogorov representation of the process relative to the Kolmogorov measure, then stationary source is ergodic.

Definition 2.27. *Let $f : \mathcal{A}^\infty \to [0, 1]$ be a measurable function and T a shift transformation. Function f is **sub-invariant** if*

$$f(T\boldsymbol{a}) \leq f(\boldsymbol{a}), \ \textit{for all } a \in \mathcal{A}^\infty.$$

Lemma 2.1 (Subinvariance lemma)**.** *Let $f : \mathcal{A}^\infty \to [0, 1]$ be a measurable function and P an ergodic measure. If f is sub-invariant, then $f(\boldsymbol{a})$ is a constant almost surely.*

Proof. See [3, page 24]. □

Theorem 2.2 (Birkhoff's ergodic theorem)**.** *Let T be a measure-preserving transformation on a probability space (Ω, Σ, P) and let f be an integrable function. Now*

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} f(T^{i-1}\boldsymbol{a}) = \int f \mathrm{d}P, \text{ almost surely,} \tag{2.1}$$

and the convergence is in L^1-norm.

Proof. See [17, pages 36-39]. □

Remark 8. Let I_A be **the indicator function** of A, that is

$$I_A(x) = \begin{cases} 1, & \text{if } \boldsymbol{a} \in A \\ 0, & \text{if } \boldsymbol{a} \notin A. \end{cases}$$

Now if we take $f = I_A$ in (2.1), then we get

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} I_A(T^{i-1}\boldsymbol{a}) = P(A), \text{ almost surely.}$$

(This special case of the Birkhoff's ergodic theorem has been proven in [5, page 14], but only in probability, not almost surely.)

Theorem 2.3 (Markov's inequality)**.** *Let X be a random variable taking only non-negative values. If $a > 0$, then*

$$P(X \geq a) \leq \frac{E[X]}{a}.$$

Proof. See [16, page 93]. □

Theorem 2.4 (Borel-Cantelli lemma)**.** *Let (Ω, Σ, P) be a probability space and let $x \in C$. If $\{C_n\}$ is a sequence of measurable sets, such that $\sum_{n=1}^{\infty} P(C_n) < \infty$, then $x \notin C_n$, eventually, almost surely.*

Proof. See [15, pages 4-5]. □

Remark 9. Borel-Cantelli lemma can also be stated as follows;
Let $\{X_n\}$ *be a sequence of random variables on a probability space* (Ω, Σ, P). *If for all* $\varepsilon > 0$ *it is true that* $\sum_{n=1}^{\infty} P(|X_n - X| \geq \varepsilon) < \infty$, *then* $X_n \xrightarrow{a.s.} X$.

2.3 Information theory

In this section we give some definitions and theorems of information theory. The main concept is entropy, around which the theorems of the chapter are based on. The entropy measures the uncertainty of a random variable and it is one of the most central concepts in information theory. In this section the main source is [6].

Remark 10. If the base of a logarithm is not marked, then it is 2.

Remark 11. If $p(x) = 0$, we set $\log p(x) = 0$.

Definition 2.28. *Let* X *be a discrete random variable taking values in* $\mathcal{A}$ *and let* $p(a)$ *be its probability function. The* ***entropy*** *of* X *is*

$$H(X) = -\sum_{a \in \mathcal{A}} p(a) \log p(a).$$

The nth-order-per-letter entropy *of the sequence* X_1^n *is*

$$H_n(X_1^n) = -\frac{1}{n} \sum_{a_1^n \in \mathcal{A}^n} p(a_1^n) \log p(a_1^n).$$

The process entropy *of a process* $\{X_n\}$ *is*

$$H(\{X_n\}) = \limsup_{n \to \infty} H_n(X_1^n).$$

Remark 12. The entropy of X can also be defined by

$$H(X) = E\left[\log \frac{1}{P(X)}\right].$$

Definition 2.29. *Let X and Y be discrete random variables taking values in $\mathcal{A}$ and let $p(a,b)$ be their joint probability function.* ***The joint entropy*** *of X and Y is*

$$H(X,Y) = -\sum_{a\in\mathcal{A}}\sum_{b\in\mathcal{A}} p(a,b)\log p(a,b).$$

Definition 2.30. *Let X and Y be discrete random variables taking values in $\mathcal{A}$ and let $p(a,b)$ be their joint probability distribution function.* ***The conditional entropy*** *$H(Y\,|\,X)$ is*

$$H(Y\,|\,X) = -\sum_{a\in\mathcal{A}}\sum_{b\in\mathcal{A}} p(a,b)\log p(b\,|\,a).$$

Theorem 2.5. *If X and Y are discrete random variables, then*

$$H(X,Y) = H(X) + H(Y|X).$$

Proof. See [6, page 16]. □

Theorem 2.6. *If X and Y are independent random variables, then entropy is additive, that is*

$$H(X,Y) = H(X) + H(Y).$$

Proof. See [6, page 28]. □

Definition 2.31. *Let P be a Kolmogorov measure for an ergodic process $\{X_n\}$.* ***The entropy rate*** *of the process is*

$$H(P) = \lim_{n\to\infty}\frac{1}{n}\log\frac{1}{P(a_1^n)},\ \textit{where } a_1^n \in \mathcal{A}^n.$$

Theorem 2.7. *Let P be a Kolmogorov measure for an ergodic process $\{X_n\}$. Now the entropy rate and the entropy of the process are the same i.e.*

$$H(\{X_n\}) = H(P).$$

Proof. See [17, page 61]. □

Theorem 2.8. *If a process is stationary, then the entropy rate* $H(\{X_n\})$ *is*

$$H(\{X_n\}) = \limsup_{n\to\infty} H(X_n \mid X_{n-1}, X_{n-2}, \ldots, X_1).$$

Proof. See [6, page 65]. □

Theorem 2.9 (Asymptotic equipartition property, AEP)**.** *Let* $X_1, X_2, \ldots$ *be i.i.d. random variables getting values on a probability distribution* $p(a)$*, where* $a \in \mathcal{A}$*. Now*

$$\lim_{n\to\infty} -\frac{1}{n} \log p_n(a_1^n) = H(X), \quad \textit{in probability.}$$

Proof. See [6, page 51]. □

Theorem 2.10 (Entropy theorem)**.** *Let* $\{X_n\}$ *be a stationary, ergodic process with alphabet* $\mathcal{A}$ *for which* $|\mathcal{A}| < \infty$*. If* $H(P)$ *is the entropy rate of the process, then*

$$\lim_{n\to\infty} -\frac{1}{n} \log p_n(a_1^n) = H(P), \quad \textit{almost surely.}$$

Proof. See [6, pages 475-476]. □

Remark 13. The Entropy theorem is stronger result than AEP since it assures an almost sure convergence and thus AEP can be proven as a corollary of it. The Entropy theorem is known also as the Shannon- McMillan-Breiman theorem. In some books the Entropy theorem is also called as the Asymptotic equipartition property.

2.4 Markov chains and k-types

In this section we define Markov chains and k-types and give some useful theorems for later use. Markov chains are stochastic processes and are used in numerous fields of mathematics. For instance, in mathematical theories of biology Markov chains have proved useful. k-types are empirical distributions, and they are needed in Chapter 3.3. References for this section have mainly been [12] and [17].

Definition 2.32. *Let $\{X_n\}$ be a stochastic process, where X_n takes values in a finite $\mathcal{A}$. The process is* ***a (finite) Markov process*** *if*

$$P(X_n = a_n \,|\, X_1^{n-1} = a_1^{n-1}) = P(X_n = a_n \,|\, X_{n-1} = a_{n-1}).$$

In other words, the probability of X_n is dependent only on the preceding X_{n-1} not on the others.

Definition 2.33. *Let $\{X_n\}$ be a stochastic process, where X_n takes values in finite $\mathcal{A}$. The process is* ***a k-th order Markov process*** *if X_n is dependent only of k precedent X_i, i.e.*

$$P(X_n = a_n \,|\, X_1^{n-1} = a_1^{n-1}) = P(X_n = a_n \,|\, X_{n-k}^{n-1} = a_{n-k}^{n-1}).$$

Definition 2.34. *Let $\{X_n\}$ be a Markov process.* ***The nth step transition probabilities*** *$p_{a_i a_j}(n)$ of the process are*

$$p_{a_i a_j}(n) = P(X_n = a_j \,|\, X_{n-1} = a_i), \text{ where } i, j \in |\mathcal{A}|.$$

Definition 2.35. *Let $\{X_n\}$ be a Markov process and $p_{a_i a_j}(n)$ be its nth step transition probabilities. We say that the process is* ***a Markov chain*** *if the transition probabilities do not depend on n, i.e.*

$$p_{a_i a_j}(n) = p_{a_i a_j}(n+1), \text{ for all } n \in \mathbb{Z}_+.$$

The matrix M with entries $p_{a_i a_j}$ is called ***the transition matrix*** *of Markov chain.*

Definition 2.36. *Let $\{X_n\}$ be a Markov process.* ***The initial probability vector*** *of the process is the vector π with components $\pi_i = P(X_0 = a_i)$.*

Theorem 2.11. *Let $\{X_n\}$ be a Markov chain with the transition matrix M and the initial probability vector π. The entropy of the chain is then*

$$H(\{X_n\}) = H(X_2 \,|\, X_1) = -\sum_{i,j} \pi_i M_{ij} \log M_{ij}.$$

Proof. See [6, pages 64-66]. □

Remark 14. The entropy of a kth order Markov chain is

$$H(\{X_n\}) = H(X_{k+1} \,|\, X_1^k).$$

Definition 2.37. *Let $a_1^n \in \mathcal{A}^n$ and let p_k be a probability distribution on $\mathcal{A}^k$, such that for $x_1^k \in \mathcal{A}^k$*

$$p_k(x_1^k \,|\, a_1^n) = \frac{|\{i \in [1, n-k+1] \,:\, a_i^{i+k-1} = x_1^k\}|}{n-k+1}.$$

We see that $P_k(x_1^k \,|\, a_1^n)$ is the relative frequency of each k-block in a_1^n. We say that p_k is the ***k-type*** *of a_1^n.*

Definition 2.38. *Let $a_1^n \in \mathcal{A}^n$ and $b_1^n \in \mathcal{A}^n$ and let $p_k(\cdot \,|\, a_1^n)$ and $p_k(\cdot \,|\, b_1^n)$ be their k-types. We say that a_1^n and b_1^n are* ***k-type equivalent*** *if $p_k(\cdot \,|\, a_1^n) = p_k(\cdot \,|\, b_1^n)$. We call the k-type equivalence classes $\mathcal{T}_{p_k}(a_1^n) = \mathcal{T}_{p_k}$ as* ***k-type classes.***

Theorem 2.12. *The number of possible k-types is at most $(n-k+2)^{|\mathcal{A}|^k}$.*

Proof. See [17, pages 64]. □

Definition 2.39. ***The empirical $(k-1)$st order Markov entropy*** *$\widehat{H}^{(k-1)}$ of a_1^n is*

$$\widehat{H}^{(k-1)} = -\sum_{x_1^k \in \mathcal{A}^k} p_k(x_1^k \,|\, a_1^n) \log \widehat{p}(x_k \,|\, x_1^{k-1}),$$

where

$$\widehat{p}(x_k \mid x_1^{k-1}) = \frac{p_k(x_1^k \mid a_1^n)}{\sum_{b_k \in \mathcal{A}} p_k(x_1^{k-1} b_k \mid a_1^n)}.$$

Remark 15. The empirical $(k-1)$st order Markov entropy is equal to the entropy of a Markov chain in Theorem 2.11.

Theorem 2.13. *Number of k-type classes have an upper bound*

$$|\mathcal{T}_{p_k}| \leq (n-k)2^{(n-k)\widehat{H}^{(k-1)}}.$$

Proof. See [17, pages 65]. □

3 Coding

The subject of this chapter is coding. The idea of codes is to represent symbols (or words) of the source alphabet in symbols of another system. Usually the system is the binary system which consists of the symbols 0 and 1. In the thesis we concentrate mainly on binary codes although we prove some more general theorems. We denote by $\mathcal{B}^* = \{0,1\}^*$ the set of all finite-length binary words.

A problem in coding is how to create a code which is unambiguous and which uses as small amount of bits as possible. This is an important question even though nowadays computers become more and more faster and the sizes of their memories grow fast. Thus this chapter is also of independent interest although it is included in the thesis to get some tools for later use.

3.1 Code

In this section we give some definitions for basic concepts such as a code and the length of a codeword. We also prove that there exists a coding of positive

integers the length of codewords of which satisfies a specific formula. This coding is called an Elias code.

We start by defining the concepts of a code and of a prefix code, of which we give two examples.

Definition 3.1. ***A code*** *C for a random variable X taking values in $\mathcal{A}^n$ is a mapping $C : \mathcal{A}^n \to \mathcal{B}^*$.* ***The codeword of*** *a_1^n is $C(a_1^n)$. If the mapping C is one-to-one, then the code is said to be* ***non-singular****.*

Definition 3.2. *A nonempty word u is* ***a prefix*** *of a word v if there exists a non-empty w, such that $v = uw$.* ***A prefix code*** *is a non-singular code for which no codeword is a prefix of another.*

Example 3.1. a) Let $\mathcal{A} = \{A, C, G, T\}$. The function $C_1 : \mathcal{A} \to \{0,1\}^2$, with

$$C_1(A) = 00 \qquad C_1(C) = 01 \qquad C_1(G) = 10 \qquad C_1(T) = 11$$

is a prefix code.

b) Let $\mathcal{A} = \{$penguins, are, birds, that, cannot, fly $\}$. The function $C_2 : \mathcal{A} \to \mathcal{B}^*$, with

$$\begin{array}{lll} C_2(\text{penguins}) = 10100 & C_2(\text{are}) = 0 & C_2(\text{birds}) = 10101 \\ C_2(\text{that}) = 11 & C_2(\text{cannot}) = 100 & C_2(\text{fly}) = 1011 \end{array}$$

is a prefix code.

We continue by defining the concepts of a code sequence and length of code.

Definition 3.3. ***A code sequence*** *is a sequence $\{C_n : n \in \mathbb{Z}_+\}$, where C_n is a code $C_n : \mathcal{A}^n \to \mathcal{B}^*$. If each C_n is non-singular, then the sequence $\{C_n\}$ is* ***faithful****.*

Definition 3.4. ***A length function*** $\mathcal{L}$ *of a code* C *is a function which maps the codewords of* C *to their lengths i.e.*

$$\mathcal{L}(C(a)) = \text{ the length of the codeword } C(a).$$

The denotation $\mathcal{L}(C(a))$ *is usually abbreviated with* $\mathcal{L}(a)$. ***The expected length*** *of the code* C *is*

$$\mathcal{L}(C) = \sum_{a_1^n \in \mathcal{A}^n} p(a_1^n)\mathcal{L}(C(a_1^n)).$$

Now we can define a certain property of integer codings, namely that of being an Elias code, which roughly means that the length of the codewords are sufficiently small. We then continue by proving that a code having this property can be constructed.

Definition 3.5. *A prefix code* $\mathcal{E} : \mathbb{Z}_+ \to \mathcal{B}^*$, *is called* ***an Elias code*** *if*

$$\mathcal{L}(\mathcal{E}(n)) = \log n + o(\log n).$$

Lemma 3.1. *There exists an Elias code.*

Proof. Cf. [17, page 75].

We start the proof by defining the codeword $\mathcal{E}(n)$ as a concatenation of three sequences. First we let $w(n)$ to be the binary representation of n and if l_1 is the length of $w(n)$, then the block $v(n)$ is the binary representation of l_1. Also, if l_2 is the length of $v(n)$, then $u(n)$ is a sequence of l_2 consecutive 0-bits. Now we let

$$\mathcal{E}(n) = u(n)v(n)w(n).$$

(See Example 3.2 to see some codewords of integers.)

We first show that this coding is a prefix code. Let

$$\mathcal{E}(n)W = u(n)v(n)w(n)W = u(m)v(m)w(m) = \mathcal{E}(m).$$

Since $v(n)$ and $v(m)$ are binary representations, they both start with 1. Also, as $u(n)$ is a sequence of 0s, the length of $u(n)$ and $u(m)$ must be the same. So, in order to the equality of $\mathcal{E}(n)W$ and $\mathcal{E}(m)$ would be achieved, it must be that $u(n) = u(m)$. This again means that $v(n)$ and $v(m)$ have the same lengths (since they have the same length as $u(n)$ and $u(m)$), and because of that $v(n) = v(m)$, or otherwise the assumption $\mathcal{E}(n)W = \mathcal{E}(m)$ would not hold. This further leads to equality of lengths of $w(n)$ and $w(m)$ and thus W is empty. Clearly $w(n) = w(m)$, and specifically $n = m$ which shows that this code is a prefix code.

We still have to prove that $\mathcal{L}(\mathcal{E}(n)) = \log n + o(\log n)$. First we consider the length of $w(n)$. Since n is simply the binary representation of n which is $a_r a_{r-1} \dots a_0$ where $n = a_r 2^r + a_{r-1} 2^{r-1} + \dots + a_0$ and since $2^r < n \leq 2^{r+1}$ it follows that $r < \log(n+1) \leq r+1$ and it is clear that $\lceil \log(n+1) \rceil (= \lfloor \log n + 1 \rfloor)$ bits are needed in the coding. For each of the codewords $u(n)$ and $v(n)$ the number of bits that are needed is $\lceil \log(\lceil \log(n+1) \rceil + 1) \rceil$ since $v(n)$ is the binary representation of $\lceil \log(n+1) \rceil$ and $u(n)$ has the same length. Now it follows that

$$\begin{aligned} \mathcal{L}(\mathcal{E}(n)) &= \lceil \log(n+1) \rceil + 2 \lceil \log(\lceil \log(n+1) \rceil + 1) \rceil \qquad (3.2) \\ &= \log n + o(\log n) + 2 \lceil \log(\log n + o(\log n)) \rceil \\ &= \log n + o(\log n) + 2 \lceil \log \log n + o(\log \log n) \rceil \\ &= \log(n) + o(\log n) + 2 \log \log n + 2o(\log \log n) \\ &= \log(n) + o(\log n). \end{aligned}$$

This completes the proof of Lemma 3.1. □

In the next example we present some codewords of an Elias code.

Example 3.2. For the Elias code presented in the proof of Lemma 3.1 the following are a couple of its codewords

$$\begin{aligned}\mathcal{E}(5) &= 0011101,\\ \mathcal{E}(10) &= 0001001010,\\ \mathcal{E}(15) &= 0001001111,\\ \mathcal{E}(21) &= 00010110101.\end{aligned}$$

The next lemma gives another equality of the codeword length of the previous coding. This lemma is needed in the proof of Theorem 5.2.

Lemma 3.2. *There is a prefix code $C : \mathbb{Z}_+ \to \mathcal{B}^*$ such that for $L \geq 4$, $L \in \mathbb{Z}_+$*

$$\mathcal{L}(C(L)) = \log L + O(\log\log L).$$

Proof. Cf. [22]. Let C be the Elias code presented in the proof of Lemma 3.1. Now if $L \geq 4$, then we get from the equality (3.2)

$$\begin{aligned}\mathcal{L}(C(L)) &= \lceil\log(L+1)\rceil + 2\lceil\log(\lceil\log(L+1)\rceil + 1)\rceil\\ &\leq \log L + 2 + 2(\log\log L + 2)\\ &\leq \log L + 8\log\log L\end{aligned}$$

and thus

$$\mathcal{L}(C(L)) = \log L + O(\log\log L).$$

□

Remark 16. For the coding C presented in the proof of Lemma 3.2 it holds that for small $L \in \mathbb{Z}_+$

$$\mathcal{L}(C(L)) \leq \log(L+1) + O(\log\log(L+2)).$$

See [22].

In the next example we construct a prefix code the codelength of which is close to the entropy.

Example 3.3. (Exercise 5.12, [6].) We have a random variable X taking values in $\mathcal{A}$ with $|\mathcal{A}| = m$, the values having the probabilities $p_1, p_2, \ldots, p_m$ ordered so that $p_1 \geq p_2 \geq \cdots \geq p_m$ (if the elements of $\mathcal{A}$ are not integers we map them to integers). We now build a code C such that the codeword of each $k \in \mathcal{A}$ is the binary representation of $0 \leq F_k = \sum_{i=1}^{k-1} p_i < 1$ rounded off to $l_k = \lceil -\log p_k \rceil$ bits. For example, if X takes values in $\{1, 2, 3, 4, 5\}$ with probabilities $0,672; 0,213; 0,054; 0,0305; 0,0305$, we get the codewords as follows

k	p_k	F_k	l_k	binary repr.	C(k)
1	0.672	0	1	0	0
2	0.213	0.672	3	0.1010110...	101
3	0.054	0.885	5	0.1110001...	11100
4	0.0305	0.939	6	0.1111000...	111100
5	0.0305	0.9695	6	0.1111100...	111110.

We show that the code defined in this way is always a prefix code and that $H(X) \leq \mathcal{L}(C) \leq H(X) + 1$.

The latter inequality is true since

$$\begin{aligned} H(X) &= \sum_{a \in \mathcal{A}} -p(a) \log p(a) \\ &\leq \sum_{a \in \mathcal{A}} p(a) \lceil -\log p(a) \rceil = \mathcal{L}(C) \\ &\leq \sum_{a \in \mathcal{A}} p(a)(-\log p(a) + 1) \\ &= \sum_{a \in \mathcal{A}} p(a)(-\log p(a)) + \sum_{a \in \mathcal{A}} p(a) = H(X) + 1. \end{aligned}$$

We assume now that C is not a prefix code. In that case there are $l, k \in \mathcal{A}$ with $l < k$ such that $C(l)K = c_1 c_2 \ldots c_r K = c_1 c_2 \ldots c_r c_{r+1} \ldots c_t = C(k)$. We

mark by $\widetilde{F}(i)$ the binary representation of $C(i)$ which clearly is $\leq F(i)$. Now as $C(l)$ is a prefix of $C(k)$, and since $F(l)$ is rounded off to $r = l_l$ bits it has to be that $F(k) - \widetilde{F}(l) < \frac{1}{2^{l_l}}$. Hence we get

$$F(k)-F(l) \leq F(k)-\widetilde{F}(l) < \frac{1}{2^{l_l}} = \frac{1}{2^{\lceil -\log p_l \rceil}} \leq \frac{1}{2^{-\log p_l}} = p_l = F(l+1)-F(l).$$

As a result we get $F(k) < F(l+1)$ which is impossible since $k > l$ and so our assumption is false and thus C is a prefix code.

3.2 Code inequalities

Here we prove two inequalities related to lengths of codewords. The first inequality is the very famous Kraft inequality which gives a sufficient and necessary condition for a code to be a prefix code. We also show by way of an example how the Kraft inequality can be used to recognize non-prefix codes. The second inequality, due to Barron, gives an almost sure result for code lengths. We need these inequalities later. First we, however, tell how we can present in a clarifying way a prefix code with its code tree [6, page 82].

The code $C : \mathcal{A} \to \mathcal{D}^*$ with $|\mathcal{D}| = D$ can be represented by a D-ary tree. Every edge of the tree represents a letter of the code alphabet and each node has maximum D children, every edge having different letters as a "name". Each codeword is obtained at a leaf of the tree by starting from the root. The codeword is created by collecting the name of an edge from each level of the tree until the codeword is obtained. If each leaf represents a codeword, then the code is prefix code. An example of the code tree can be seen in Figure 1.

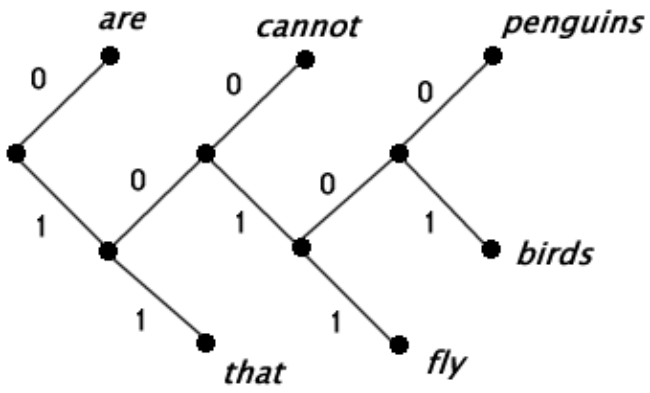

Figure 1: The code tree of the code of the example 3.1 b)

Theorem 3.1 (Kraft inequality)**.** *Let C be a code over an alphabet $\mathcal{D}$ with codeword lengths $l_1, l_2, \ldots, l_n$. The code is a prefix code if and only if the codeword lengths satisfy the inequality*

$$\sum_{i=1}^{n} D^{-l_i} \leq 1, \text{ where } D = |\mathcal{D}|.$$

Proof. Cf. [6, pages 82-83], [1].

(The "only if" part) We assume first that there exist a code C, with $|\mathcal{D}|$=D and that T is its code tree. We now prove an equality after which the inequality follows immediately.

We prove by induction that;

Proposition:
If T is a complete D-ary tree with height h, number of leaves M and length of paths from root to leaves $l_1, l_2, \ldots, l_M$, then $\sum_{i=1}^{M} D^{-l_i} = 1$.

1. If $h = 1$, then it is clear that $\sum_{i=1}^{D} D^{-1} = D \cdot D^{-1} = 1$.

2. The induction hypothesis is that the proposition is true when $h = t$.
   We have to show that the proposition is true when $h = t + 1$.
   Let T be a full D-ary tree with heigth $t + 1$ and leaves $v_1, v_2, \ldots, v_M$. Let now $v_{k+1}, \ldots, v_M$ be those leaves for which the length of path from root to leaf is $t + 1$ (i.e. $l_{k+1} = l_{k+2} = \cdots = l_M = t + 1$). Let T' be a subtree of T with heigth t and which is obtained from T by removing $v_{k+1}, v_{k+2}, \ldots, v_M$. Now T' has $k + s$ leaves, where $s = \frac{M-k}{D}$ (this is always an integer since T' is complete, too) and these s branches have the lengths $l_{k+1} - 1$. Using the induction hypothesis we get

$$D^{-l_1} + D^{-l_2} + \cdots + D^{-l_k} + s\, D^{-(l_{k+1}-1)} = 1. \tag{3.3}$$

   Since T has $M = k + s * D$ leaves, we get for T

$$\begin{aligned} \sum_{i=1}^{M} D^{-l_i} &= D^{-l_1} + D^{-l_2} + \cdots + D^{-l_k} + D\, s\, D^{-(l_{k+1})} \\ &= D^{-l_1} + D^{-l_2} + \cdots + D^{-l_k} + D\, s\, D^{-(l_{k+1}-1)} \tfrac{1}{D} \\ &\overset{(3.3)}{=} 1 \end{aligned}$$

3. By the principle of induction, the proposition is true.

Now the Kraft inequality follows, since the tree mentioned in the proposition was required to be complete and a code tree T_C is a subtree of some complete D-ary tree T_D, just having less leaves than T_D and thus less paths from root to leaf, which ensures that

$$\sum_{i=1}^{n} D^{-l_i} \leq 1.$$

(The "if" part) We assume then that the codeword lengths $l_1, l_2, \ldots, l_n$ satisfy the inequality

$$\sum_{i=1}^{n} D^{-l_i} \leq 1.$$

We define n_j to be the number of those codewords the length of which is equal to j and L to be the maximum of lengths of codewords, i.e. $L = \max_i l_i$. Since the inequality

$$\sum_{i=1}^{n} D^{-l_i} \leq 1$$

holds, we get the inequalities

$$\sum_{j=1}^{L} n_j D^{-j} \leq 1 \text{ and } \sum_{j=1}^{L} n_j D^{L-j} \leq D^L.$$

Now by rearranging the terms of the last inequality we get

$$n_L \leq D^L - n_1 D^{L-1} - n_2 D^{L-2} - \cdots - n_{L-1} D.$$

Next we just "drop" n_L away and divide the inequality by D. The result is

$$n_{L-1} \leq D^{L-1} - n_2 D^{L-2} - n_3 D^{L-3} - \cdots - n_{L-2} D. \tag{3.4}$$

We keep on dropping and dividing and get the inequalities

$$\begin{aligned} n_{L-2} &\leq D^{L-2} - n_3 D^{L-3} - n_4 D^{L-4} - \cdots - n_{L-3} D && (3.5)\\ &\vdots \quad \vdots \quad \vdots && (3.6)\\ n_2 &\leq D^2 - n_1 D && (3.7)\\ n_1 &\leq D. && (3.8) \end{aligned}$$

We have $n_1 \leq D$ words l_i of length 1. We code arbitrarily these words to n_i symbols of $\mathcal{D}$. After coding the words we have $D - n_1$ symbols (codewords) unused. We get now $D^2 - n_1 D$ codewords to code words of length 2 by concatenating one symbol of $\mathcal{D}$ after each codeword which was not used in the coding of words of length 1 and by doing this for each symbol of $\mathcal{D}$. This is sufficient to code the n_2 words of length 2 on the basis of inequality (3.7) and after this we have $D^2 - n_1 D - n_2$ codewords unused. We concatenate

again the symbols of $\mathcal{D}$ after these codewords and carry on doing this until we have coded all words. The inequalities (3.4)-(3.5) assure that there are always enough codewords to code words of length i. Since the coding of words uses only those codewords which are not prefixes of shorter words, the code is a prefix code. □

Remark 17. The Kraft inequality can be proven also for a countable infinite set of codewords. See [6, page 84].

Example 3.4. A binary code whose codeword lengths are 2, 2, 3, 3, 4, 4, 5, 5, 8, 8, 8, 8, 8, 8 **may** be a prefix code since

$$2 \cdot \frac{1}{2^2} + 2 \cdot \frac{1}{2^3} + 2 \cdot \frac{1}{2^4} + 2 \cdot \frac{1}{2^5} + 6 \cdot \frac{1}{2^8} \approx 0,94.$$

A binary code whose codeword lengths are 2, 2, 3, 3, 3, 4, 4, 5 can **never** be a prefix code by Kraft inequality since

$$2 \cdot \frac{1}{2^2} + 3 \cdot \frac{1}{2^3} + 2 \cdot \frac{1}{2^4} + \cdot \frac{1}{2^5} \approx 1,03.$$

Theorem 3.2 (Barron inequality)**.** *Let $C : \mathcal{A}^n \to \mathcal{B}^*$ be a prefix code and P a Borel probability measure on $\mathcal{A}^\infty$. Let $\{\alpha_n\}$ be a sequence of positive numbers such that $\sum_{n=1}^{\infty} 2^{-\alpha_n} < \infty$. Now eventually, almost surely*

$$\mathcal{L}(a_1^n) + \log P(a_1^n) \geq -\alpha_n.$$

Proof. Cf. [17, page 125].

If $\mathcal{L}(a_1^n) + \log P(a_1^n) < -\alpha_n$, then $P(a_1^n) < 2^{-\mathcal{L}(a_1^n)} 2^{-\alpha_n}$. We define now for each n the set

$$B_n = \{a_1^n \,:\, P(a_1^n) < 2^{-\mathcal{L}(a_1^n)} 2^{-\alpha_n}\} = \{a_1^n \,:\, \mathcal{L}(a_1^n) + \log P(a_1^n) < -\alpha_n\},$$

and show that eventually, almost surely $a_1^n \in B_n$. The measure of the set B_n is

$$P(B_n) = \sum_{a_1^n \in B_n} P(a_1^n) \leq \sum_{a_1^n \in B_n} 2^{-\mathcal{L}(a_1^n)} 2^{-\alpha_n} \leq 2^{-\alpha_n}.$$

The last inequality follows from Kraft inequality which says that

$$\sum_{a_1^n \in B_n} 2^{-\mathcal{L}(a_1^n)} \leq 1.$$

We also know that $\sum_{n=1}^{\infty} 2^{-\alpha_n} < \infty$, and thus $\sum_{n=1}^{\infty} P(B_n) < \infty$. Now we let $a_1^n \in \mathcal{A}^n$ and the Borel-Cantelli lemma tells us that eventually, almost surely $a_1^n \notin B_n$, which yields the result

$$\mathcal{L}(a_1^n) + \log P(a_1^n) \geq -\alpha_n, \text{eventually, almost surely.}$$

Hence we have proven the Barron inequality. □

3.3 Existence of codes

In this section we first define the concept of an universal coding.

Definition 3.6. *Let $\{C_n\}$ be a code sequence and P a Kolmogorov measure of ergodic process $\{X_n\}$ with an alphabet $\mathcal{A}$. The sequence $\{C_n\}$ is **universally asymptotically optimal** or **universal** if*

$$\limsup_{n\to\infty} \frac{\mathcal{L}(a_1^n)}{n} \leq H(P).$$

We now prove two theorems. The first one says that one can find universal codings. So it is possible to build codes that are good. However, the second theorem tells that in no code sequence there can be infinitely many codes of which the lengths of codewords are less than the entropy of the object they code, and thus we can say that there are not too good codes.

Theorem 3.3 (There are universal codes)**.** *Let P be a Kolmogorov measure of any ergodic process $\{X_n\}$. There exists a prefix code sequence $\{C_n\}$ such that*

$$\limsup_{n\to\infty} \frac{\mathcal{L}(a_1^n)}{n} \leq H(P), \text{ almost surely.}$$

Proof. Cf. [17, page 122-124].

We first define a prefix code sequence $\{C_n\}$ and then show that almost surely $\limsup_{n\to\infty} \frac{\mathcal{L}(a_1^n)}{n} \leq H(\{X\})$ for any ergodic measure P. This is sufficient to proof the theorem because by Theorem 2.7 we know that $H(\{X\}) = H(P)$.

First we give a definition of a specific k-type, **a circular k-type** $\widetilde{P}_k$, which is a measure on $\mathcal{A}^k$ defined by

$$\widetilde{P}_k(x_1^k \,|\, a_1^n) = \frac{|\{i \in [1,n] \,:\, \tilde{a}_i^{i+k-1} = x_1^k\}|}{n}, \quad \text{where } \tilde{a}_1^{n+k-1} = a_1^n a_1^{k-1}$$
$$\text{and } x_1^k \in \mathcal{A}^k, \text{ with some } k \leq n.$$

Since a circular k-type is just a special k-type, the bounds given in Theorems 2.12 and 2.13 are valid also for the number of circular k-types $\widetilde{N}(k,n)$ and the number of circular k-type classes $|\widetilde{T}_k(x_1^n)|$. We just have to remember that the length of the sequence $\tilde{a}$ is $n+k-1$ instead of n and the number of possible circular k-types is hence at most $(n+1)^{|\mathcal{A}|^k}$ and an upper bound for the number of circular k-type classes $\widetilde{T}_k$ is $(n-1)2^{(n-1)\widetilde{H}_{k-1,a_1^n}}$.

Now since

$$\widetilde{P}_{k-1}(x_1^{k-1} \,|\, a_1^n) = \sum_{x_k \in \mathcal{A}^k} \widetilde{P}_k(x_1^k \,|\, a_1^n),$$

the inequality

$$\begin{aligned} \widetilde{H}_{k-1,a_1^n} &= -\sum_{x_1^k \in \mathcal{A}^k} \widetilde{P}_{k-1}(x_1^{k-1} \,|\, a_1^n) \log \frac{\widetilde{P}_k(x_1^k \,|\, a_1^n)}{\sum_{b_k \in \mathcal{A}} \widetilde{P}_k(x_1^{k-1} b_k \,|\, a_1^n)} \\ &\leq -\sum_{x_1^{i+1} \in \mathcal{A}^{i+1}} \widetilde{P}_i(x_1^i \,|\, a_1^n) \log \frac{\widetilde{P}_{i+1}(x_1^{i+1} \,|\, a_1^n)}{\sum_{b_k \in \mathcal{A}} \widetilde{P}_k(x_1^i b_k \,|\, a_1^n)} = \widetilde{H}_{i,a_1^n} \end{aligned} \tag{3.9}$$

holds for all $1 \leq i \leq k-1$.

We first let $k = k(n) = \lfloor \frac{1}{2} \log_{|\mathcal{A}|} n \rfloor$ and then we construct the code C_n by using circular k-types so that C_n is comprised of two parts. So, a codeword

of a_1^n is $C_n(a_1^n) = b_1^m b_{m+1}^t$, where the first part b_1^m is a binary sequence (with fixed length) which tells the index of the circular k-type of a_1^n. The second part b_{m+1}^t is a binary sequence (with variable length) which represents the index of a_1^n in its circular k-type class when there is some enumeration of $\widetilde{T}_k(a_1^n)$. Now we get an upper bound for the total code length

$$\mathcal{L}(a_1^n) \leq \lceil \log \widetilde{N}(k,n) \rceil + \lceil \log |\widetilde{T}_k(a_1^n)| \rceil.$$

Again since $k \leq \frac{1}{2} \log_{|\mathcal{A}|} n$ and by Theorem 2.12 we get

$$\lceil \log \widetilde{N}(k,n) \rceil \leq \lceil \log (n+1)^{|\mathcal{A}|^k} \rceil \leq 1 + \log (n+1)^{|\mathcal{A}|^{\frac{1}{2}\log_{|\mathcal{A}|} n}} = 1 + \sqrt{n} \log(n+1).$$

Also, by Theorem 2.13 we get

$$\lceil \log |\widetilde{T}_k(a_1^n)| \rceil \leq \lceil \log ((n-1) 2^{(n-1)\widetilde{H}_{k-1,a_1^n}}) \rceil \leq 1 + (n-1)\widetilde{H}_{k-1,a_1^n} + \log(n-1).$$

As a result, we get

$$\begin{aligned} &\limsup_{n\to\infty} \frac{\mathcal{L}(a_1^n)}{n} \\ &\leq \limsup_{n\to\infty} \frac{2 + \sqrt{n}\log(n+1) + (n-1)\widetilde{H}_{k-1,a_1^n} + \log(n-1)}{n} \\ &= \limsup_{n\to\infty} \widetilde{H}_{k-1,a_1^n}. \end{aligned}$$

We still have to show that for any process $\limsup_{n\to\infty} H_{k-1,a_1^n} \leq H(\{X_m\})$ holds, almost surely.

Now let P be an ergodic measure of a process with entropy $H = H(\{X_n\})$. Let then $\epsilon > 0$ and choose K such that

$$H_{K-1} = H(X_K \,|\, X_1^{K-1}) \leq H + \epsilon,$$

where H_{K-1} is the entropy of the Markov chain of order $K-1$ defined by the conditional probability $P(a_1^K \,|\, a_1^{K-1}) = \frac{P(a_1^K)}{P(a_1^{K-1})}$. (We can always find this

H_{K-1} by Theorem 2.8.) Now, since $\frac{1}{n}\sum_{i=1}^{n} I_{\mathcal{A}^k}(T^{i-1}\tilde{a}) = \widetilde{P}(x_1^k \,|\, a_1^n)$, we can use the Birkhoff's ergodic theorem and thus for fixed K

$$\lim_{n\to\infty} \widetilde{P}(x_1^k \,|\, a_1^n) = P(a_1^k) \quad \text{almost surely.}$$

Further this equality of the probabilities leads straightforwardly to the equality of the entropies, too, i.e.

$$\lim_{n\to\infty} \widetilde{H}_{K-1,a_1^n} = H_{K-1} \text{ almost surely.}$$

Now this ensures that there exists a $N = N(a,\epsilon) \in \mathbb{Z}_+$ such that for $n \geq N$

$$\widetilde{H}_{K-1,a_1^n} \leq H + 2\epsilon.$$

Again if we take n a sufficiently large, $k(n) \geq K$ and thus by the inequality (3.9),

$$\widetilde{H}_{k(n)-1,a_1^n} \leq \widetilde{H}_{K-1,a_1^n} \leq H + 2\epsilon,$$

Now since ϵ is arbitrary, it holds that almost surely

$$\limsup_{n\to\infty} \widetilde{H}_{k(n),a_1^n} \leq H,$$

and this completes the proof of Theorem 3.3. □

Theorem 3.4 (Too-good codes do not exist)**.** *Let $\{C_n\}$ be a faithful code sequence and let P be an ergodic measure having entropy $H(P)$. Now*

$$\liminf_{n\to\infty} \frac{\mathcal{L}(a_1^n)}{n} \geq H(P), \ \textit{almost surely.}$$

Proof. Cf. [17, pages 76,125].

We have a faithful code sequence $\{C_n\}$. This can be converted to a prefix code sequence such that the asymptotic properties of the sequence are not disrupted. This can be done, for example, by using the so called Elias header

technique. In this technique the code sequence $\{C_n\}$ is converted to a prefix code $C : \mathcal{A}^* \to \mathcal{B}^*$, where
$C(a_1^n) = \mathcal{E}(n)C_n(a_1^n), \quad a_1^n \in \mathcal{A}^n$ and $n \in \mathbb{Z}_+$.

Let now $\{\alpha_n\}$ be a sequence of positive numbers $\alpha_n = 2\log_{|\mathcal{A}|} n$. Now

$$\sum_{n=1}^{\infty} 2^{-\alpha_n} = \sum_{n=1}^{\infty} 2^{-2\log_{|\mathcal{A}|} n} = \sum_{n=1}^{\infty} \left(\frac{1}{4}\right)^{\log_{|\mathcal{A}|} n} \leq \sum_{n=1}^{\infty} \left(\frac{1}{4}\right)^{n} < \infty.$$

The Barron inequality implies that $\mathcal{L}(a_1^n) + \log P(a_1^n) \geq -\alpha_n$ eventually, almost surely and this yields that eventually, almost surely

$$\liminf_{n\to\infty} \frac{\mathcal{L}(a_1^n)}{n} \geq \liminf_{n\to\infty} \frac{-\log P(a_1^n)}{n} - \liminf_{n\to\infty} \frac{\alpha_n}{n}.$$

On the other hand, $\liminf_{n\to\infty} \frac{\alpha_n}{n} = 0$ and $\liminf_{n\to\infty} \frac{-\log P(a_1^n)}{n} = H(P)$ for any ergodic measure P by the Entropy theorem. Herewith almost surely

$$\liminf_{n\to\infty} \frac{\mathcal{L}(a_1^n)}{n} \geq H(P)$$

and we have proven Theorem 3.3. □

4 Packing

This chapter deals with packings which are collections of subintervals of some interval of integers. In this chapter among other things we introduce the Packing lemma which we use in the proof of Theorem 5.1. In this chapter we adopt the convention $[n, m] = \{j \in \mathbb{Z}_+ : n \leq j \leq m\}$.

We start by giving definitions related to packings and covers.

Definition 4.1. *Let $m : \mathbb{Z}_+ \to \mathbb{Z}_+$ be a function satisfying $m(i) \geq i$. A collection $\mathcal{C} = \{C_i \in \mathbb{Z}_+ \,|\, i \in \mathbb{Z}_+\}$ subsets $C_i \subseteq \mathbb{Z}_+$ is a **strong cover** of $\mathbb{Z}_+$ if $C_i = [i, m(i)]$ for all $i \in \mathbb{Z}_+$.*

Definition 4.2. *Let L be an integer, $\mathcal{C}$ be a strong cover of $\mathbb{Z}_+$ and $[1, K] \subseteq \mathbb{Z}_+$ an interval such that $L \leq K$. The interval $[1, K]$ is* $(\boldsymbol{L, \delta})$***-strongly covered by*** $\boldsymbol{\mathcal{C}}$ *if*

$$\frac{|\{i \in [1, K] : m(i) - i + 1 > L\}|}{K} \leq \delta.$$

Example 4.1. Let $m : \mathbb{Z}_+ \to \mathbb{Z}_+$ be a function such that

$$m(i) = \begin{cases} i + 2, & \text{if } i \text{ is even and } i \leq 6 \\ i + 1, & \text{otherwise.} \end{cases}$$

Let $\mathcal{C}$ be a collection of sets $C_i = [i, m(i)]$. Now $\mathcal{C}$ is a strong cover of $\mathbb{Z}_+$. Further $[1, 10]$ is $(2, \frac{1}{3})$-strongly covered by $\mathcal{C}$ since $|\{i \in [1, 10] : m(i) - i + 1 > 2\}| = 3 \leq \frac{10}{3}$, but not $(1, \frac{1}{3})$-strongly covered by $\mathcal{C}$ since $|\{i \in [1, 10] : m(i) - i + 1 > 1\}| = 10 > \frac{10}{3}$.

Definition 4.3. *Let $\mathcal{C}'$ be a collection of subintervals C_i of the interval $[1, K]$. The collection $\mathcal{C}'$ is a* $\boldsymbol{\theta}$***-packing of*** $[1, K]$ *if*

i) If $i \neq j$, then $C_i \cap C_j = \emptyset$ for all $C_i, C_j \in \mathcal{C}'$, and

ii) $|\bigcup_i C_i| \geq \theta K$.

Example 4.2. The set $\{[1, 2], [5, 6], [9, 10]\}$ is $\frac{1}{2}$-packing of $[1, 10]$ since intervals are pairwise disjoint and their union is large enough.

Definition 4.4. *Let (Ω, Σ, P) be a probability space.* ***A stopping time*** *is a measurable function $\tau : \Omega \to \bar{\mathbb{Z}}_+ = \mathbb{Z}_+ \cup \{\infty\}$.*

Definition 4.5. *Let P be a stationary measure on $\mathcal{A}^\infty$. A stopping time τ is* $\boldsymbol{P}$***-almost surely finite*** *if*

$$P(\{\boldsymbol{a} : \tau(\boldsymbol{a}) = \infty\}) = 0.$$

We can now introduce the following lemma which presents a way to build a strong cover of $\mathbb{Z}_+$.

Lemma 4.1. *If P is a stationary measure on $\mathcal{A}^\infty$, T a measure-preserving transformation and τ a P-almost surely finite stopping time, then for each $n \in \mathbb{Z}_+$ and for almost every $\boldsymbol{a} \in \mathcal{A}^\infty$, it holds that $\tau(T^{n-1}\boldsymbol{a}) < \infty$ and the collection*

$$\mathcal{C}_\tau = \mathcal{C}(\boldsymbol{a}, \tau) = \{C_i \,:\, C_i = [n, \tau(T^{n-1}\boldsymbol{a}) + n - 1], n \in \mathbb{Z}_+\} \tag{4.10}$$

is almost surely a strong cover of $\mathbb{Z}_+$. (Cf. [17, page 40].)

Proof. Since $P(\{\boldsymbol{a} \,:\, \tau(\boldsymbol{a}) = \infty\}) = 0$, it is clear that for almost every $\boldsymbol{a} \in \mathcal{A}^\infty$, $\tau(\boldsymbol{a}) < \infty$ and since $T^{n-1}\boldsymbol{a} \in \mathcal{A}^\infty$ for all $n \in \mathbb{Z}_+$ it also holds for almost every $\boldsymbol{a} \in \mathcal{A}^\infty$ that $\tau(T^{n-1}\boldsymbol{a}) < \infty$. As $1 \leq \tau(T^{n-1}\boldsymbol{a}) < \infty$ it is clear that $m : \mathbb{Z}_+ \to \mathbb{Z}_+$ is a function satisfying $m(n) = \tau(T^{n-1}\boldsymbol{a}) + n - 1 \geq n$, and the intervals are $C_n = [n, m(n) = \tau(T^{n-1}\boldsymbol{a}) + n - 1]$ for all $n \in \mathbb{Z}_+$. Thus the collection $\mathcal{C}$ is a strong cover of $\mathbb{Z}_+$. □

Now we introduce and prove the very useful Packing lemma.

Lemma 4.2 (Packing lemma)**.** *Let $\mathcal{C}$ be a strong cover of $\mathbb{Z}_+$, let $\delta > 0$ be given and let $K > L/\delta$. If $[1, K]$ is (L, δ)-strongly covered by $\mathcal{C}$, then there is a subcollection $\mathcal{C}' \subset \mathcal{C}$ which is a $(1 - 2\delta)$-packing of $[1, K]$.*

Proof. Cf. [17, page 34].

We construct a subcollection $\mathcal{C}'$ of $\mathcal{C}$ by induction and then we show that it meets the conditions of a $(1 - 2\delta)$-packing. Let $m : \mathbb{Z}_+ \to \mathbb{Z}_+$ be the function that defines the strong cover $\mathcal{C} = C_i$. Now we let $\mathcal{C}'$ be a collection of intervals $[n_i, m(n_i)]$ of $[1, K]$ defined by

Step 0 Define $n_0 = 0$, and $m(n_0) = m(0) = 0$.

Step i If $m(n_{i-1}) \leq K - L$ and there exists $j \in [1 + m(n_{i-1}), K - L]$, for which $m(j) - j + 1 \leq L$, then define

$$n_i = \min \{j \in [1 + m(n_{i-1}), K - L] \, : \, m(j) - j + 1 \leq L\} .$$

Otherwise, stop.

We let now I be the number of last step where was defined new n_i, and let

$$\mathcal{C}' = \{C_{n_i} = [n_i, m(n_i)] \, : \, 1 \leq i \leq I\}.$$

Since $n_i > m(n_{i-1})$, the intervals C_{n_i} are disjoint, and condition *i)* of Definition 4.3 is satisfied. Furthermore, each $C_{n_i} \subseteq [1, K]$, since by the definition of $\mathcal{C}'$, for all i, $m(n_i) - n_i + 1 \leq L$ and this leads to the inequality chain $m(n_I) \leq L + n_I - 1 \leq L + K - L - 1 < K$.

We still have to show that $|\bigcup_i C_{n_i}| \geq (1 - 2\delta)K$. By the definition of n_i, we know that

$$\text{if } k \in [1, K - L] \text{ but } k \notin \bigcup_i C_{n_i}, \text{ then } m(k) - k + 1 > L.$$

On the other hand, we know that $[1, K]$ is (L, δ)-strongly-covered by $\mathcal{C}$ and thus

$$\left| k \in [1, K - L] \, : \, k \notin \bigcup_i C_{n_i} \right| < \delta K.$$

We also know that

$$\big|\,]K - L, K]\, \big| = L - 1 < \delta K.$$

Finally we have

$$\begin{aligned} \big|\bigcup_i C_{n_i}\big| &\geq \big|\,[1, K]\,\big| - \big|\,]K - L, K]\,\big| - \big|\,\{k \in [1, K - L] \, : \, k \notin \bigcup_i C_{n_i}\}\,\big| \\ &\geq K - \delta K - \delta K = (1 - 2\delta)K. \end{aligned}$$

This shows that condition *ii)* of definition of $(1-2\delta)$-packing also holds and thus the proof of the Packing lemma is complete. □

Remark 18. The Packing lemma defines a packing for which the length of each interval belonging to the packing is at most L.

The packing lemma has many variants. The next lemma and the following example make use of the stopping time and packing lemma.

Lemma 4.3 (The ergodic stopping-time packing lemma)**.** *Let P be an ergodic measure for a process and $\delta > 0$. If τ is a P-almost surely finite stopping time, then there is an $N = N(\delta, \boldsymbol{a})$ for almost every $\boldsymbol{a} \in \mathcal{A}^\infty$ such that if $n \geq N$, then there exists a set of intervals of collection*

$$\mathcal{C}_\tau = \mathcal{C}(\boldsymbol{a}, \tau) = \{C_i : C_i = [k, \tau(T^{k-1}\boldsymbol{a}) + k - 1], n \in \mathbb{Z}_+\}$$

which is a $(1-\delta)$-packing of $[1, n]$.

Proof. Cf. [17, pages 40-41].

By assumption, τ is almost surely finite that is $P(\{\boldsymbol{a} : \tau(\boldsymbol{a}) = \infty\}) = 0$, and this implies that it is also bounded (almost surely). Because of this, for fixed $\delta > 0$, there exists an $L \in \mathbb{Z}_+$ such that

$$P(\{\boldsymbol{a} \in \mathcal{A}^\infty : \tau(\boldsymbol{a}) > L\}) < \frac{\delta}{2}. \tag{4.11}$$

Now define the set

$$D = \{\boldsymbol{a} \in \mathcal{A}^\infty : \tau(\boldsymbol{a}) > L\}.$$

Let I_D be an indicator function of the set D. We know by the Birkhoff's ergodic theorem and the formula (4.11) that almost surely

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} I_D(T^{i-1}\boldsymbol{a}) \mathrm{d}P = P(D) < \frac{\delta}{2}.$$

Thus eventually, almost surely $\boldsymbol{a} \in G_n$ if G_n is the set defined by

$$G_n = \left\{ \boldsymbol{a} \in \mathcal{A}^\infty \,:\, \frac{1}{n} \sum_{i=1}^{n} I_D(T^{i-1}\boldsymbol{a}) < \frac{\delta}{2} \right\}.$$

We now assume that $\boldsymbol{a} \in G_n$. Let $N = N(\delta, \boldsymbol{a}) = \frac{2L}{\delta}$ and $n \geq N$. The definition of G_n leads to the fact $\sum_{i=1}^{n} I_D(T^{i-1}\boldsymbol{a}) < \frac{n\delta}{2}$ which means that there is at most $\frac{n\delta}{2}$ ks on interval $[1, n]$ such that $T^{k-1}\boldsymbol{a} \in D$ and again by the definition of D we can conclude that there is then at most $\frac{n\delta}{2}$ indices k on interval $[1, n]$ such that $\tau(T^{k-1}\boldsymbol{a}) > L$. Since $C_i = [k, \tau(T^{k-1}\boldsymbol{a}) + k - 1]$ it follows that

$$\frac{|\{k \in [1, n] \,:\, \tau(T^{k-1}\boldsymbol{a}) + k - 1 - k + 1 > L\}|}{n} \leq \frac{n\delta}{2},$$

and thus $[1, n]$ is $(L, \frac{\delta}{2})$-strongly covered by $\mathcal{C}$ which is also a strong cover of $\mathbb{Z}_+$ by Lemma 4.1. Since $n \geq \frac{2L}{\delta}$ there is a $(1 - \delta)$ -packing of $[1, n]$ by the Packing lemma. Hence we have proven Lemma 4.3. □

Example 4.3. (Exercise I.3.e.1 [17]) We say that a packing $\mathcal{C}'$ of $[1, n]$ is **separated** if there is at least one integer between any two intervals in $\mathcal{C}'$. We construct now a separated $(1 - 2\delta)$-packing of $[1, n]$. First, we let P be an ergodic measure on $\mathcal{A}^\infty$ and τ be an almost-surely finite stopping time and also $\tau(\boldsymbol{a}) \geq M > \frac{1}{\delta}$. We then define $\widetilde{\tau}(\boldsymbol{a}) = \tau(\boldsymbol{a}) + 1$. Since τ is an almost-surely stopping time, so is $\widetilde{\tau}$, too. We next define the collection

$$\widetilde{\mathcal{C}} = \{[n, \widetilde{\tau}(T^{n-1}\boldsymbol{a}) + n - 1] \,:\, n \in \mathbb{Z}_+\}.$$

By the Ergodic stopping-time packing lemma, for almost every $\boldsymbol{a} \in \mathcal{A}^\infty$, there is an N such that if $n \geq N$, then there is a $(1 - \delta)$-packing D of $[1, n]$ which consists of intervals in $\widetilde{\mathcal{C}}$. Let now

$$\mathcal{C}' = \{C_n = [n, \tau(T^{n-1}\boldsymbol{a}) + n - 1] \,:\, [n, \widetilde{\tau}(T^{n-1}\boldsymbol{a}) + n - 1] \in D\}.$$

Since $[n, \tau(T^{n-1}\boldsymbol{a}) + n - 1] \subset [n, \widetilde{\tau}(T^{n-1}\boldsymbol{a}) + n - 1]$, and D is a packing, the intervals of $\mathcal{C}'$ are disjoint. Also, since the length of each interval in D is

$\widetilde{\tau}(T^{n-1}\boldsymbol{a}) + n - 1 - n = \tau(T^{n-1}\boldsymbol{a}) \geq M > \frac{1}{\delta}$, there is at most δn intervals in D. Now it follows that $|\bigcup_{C_n \in C'} C_n| \geq (1-\delta)n - \delta n = (1-2\delta)n$. By the definition of $\widetilde{\tau}$, D and $\mathcal{C}'$ there is also always at least one integer between the intervals of $\mathcal{C}'$. Thus $\mathcal{C}'$ is a separated $(1-2\delta)$-packing of $[1, n]$.

5 Recurrence time

This chapter is the main chapter of the thesis since it deals with the recurrence time. In general we can say that recurrence time of $\boldsymbol{a}$ is the time needed until $\boldsymbol{a}$ reappears in the sequence. The source of the text in this chapter is mainly [4, pages 214-235].

The recurrence time is used, among other things, in data compression. Data compression is an important application since the amount of information grows rapidly all along. Roughly speaking we can divide the data compression techniques into two categories, statistical and dictionary techniques. The idea of statistical methods is that they code the most probable sequences with short codewords. Dictionary methods use some kind of dictionary from which the compressed text is looked for. One widely used dictionary compression technique is the Lempel-Ziv algorithm two variants of which Jakob Ziv and Abraham Lempel introduced in 1977 (see [24], LZ77) and in 1978 (see [25], LZ78). These algorithms have several different variants and they are very widely used. For example, the GIF-picture format uses the Lempel-Ziv algorithm [19].

Shortly, the basic idea of the Lempel-Ziv algorithm is the following (implementations may, however, deviate from this description significantly).
When going thorough the text which is compressed the text is scanned for blocks (strings) that have already appeared somewhere in the text. If such a block is found the recurrence time of block and the length of the block

is coded, not the text of the block itself. For example, if we have the text ADTAACDTACDTAC which is to be compressed, we first take the text ADTAAC and then, since the block DTA already appears in the text (ADTAAC), we write just the recurrence time 5 and the block length 3 to code instead of writing the block DTA. Then again looking the forward, the block CDTAC can be found in the preceding text, too and thus we code the recurrence time 4 and the block length 5. Thus in the whole we code ADTAAC(5,3)(4,5).

The preceding section of text used to find previous appearances a block of text, is often called the window or the training sequence. It can be shown that if the length of the window is infinite, then the LZ77 is optimal [23]. The optimality of the LZ78 algorithm is shown in [17, pages 131-132]. The LZ- compression techniques are an example of universal codings discussed in the Chapter 3.3.

5.1 Recurrence time theorem

In this chapter we prove the Recurrence time theorem, which is the main goal of this thesis. First we, however, give an exact definition of recurrence time.

Definition 5.1. ***The recurrence time*** R_n *of a sequence* a_1^n *in a window of length* N_0 *is a function* $R_n : \mathcal{A}^n \to \mathbb{Z}_+$,

$$R_n(a_1^n) = \begin{cases} \min\{m : a_1^n = a_{m+1}^{m+n}, 1 \leq m \leq N_0\}, & \textit{if there exists such } m, \\ N_0, & \textit{otherwise.} \end{cases}$$

Remark 19. Recurrence time is often also defined by $\min\{m : a_1^n = a_{-m+1}^{-m+n}\}$, but this does not cause any difference with our theorem. This definition is usually used if we have a window a_{-m}^0. We use this definition in our theorems of the Chapter 5.2.

Now we introduce and prove the Recurrence time theorem.

Theorem 5.1 (Recurrence time theorem)**.** *Let a source $S = \{X_n\}$ be stationary, ergodic and with finite alphabet with measure P. Then*

$$\lim_{n\to\infty} \frac{\log R_n(a_1^n)}{n} = H\{X\} \text{ almost surely.}$$

Proof. Cf. [14], [17, pages 154-158].

In the proof we use the convention $H = H\{X\}$. Let $\boldsymbol{a} \in \mathcal{A}^\infty$. We first define upper and lower limits

$$\bar{r}(\boldsymbol{a}) = \limsup_{n\to\infty} \frac{\log R_n(\boldsymbol{a})}{n}, \tag{5.12}$$

$$\underline{r}(\boldsymbol{a}) = \liminf_{n\to\infty} \frac{\log R_n(\boldsymbol{a})}{n}, \quad \text{with } R_n(\boldsymbol{a}) = R_n(a_1^n). \tag{5.13}$$

We see that $R_n(\boldsymbol{a})$ is sub-invariant since

$$\begin{aligned} R_{n-1}(T\boldsymbol{a}) &= \min\{m \,:\, Ta_1^{n-1} = a_2^n = a_{m+2}^{m+l+1}, 1 \le m \le N_0\} \\ &\le \min\{m \,:\, a_1^n = a_{m+1}^{m+l}, 1 \le m \le N_0\} \\ &= R_n(\boldsymbol{a}). \end{aligned}$$

This implies the sub-invariancy of both $\bar{r}(\boldsymbol{a})$ and $\underline{r}(\boldsymbol{a})$ and as a consequence of the Subinvariance lemma they are constant, almost everywhere. We denote these constants by $\bar{r}$ and $\underline{r}$. We show now that $\bar{r} \le H \le \underline{r}$ gives us the theorem, because from Definitions 5.12 and 5.13 it clearly follows that $\underline{r} \le \bar{r}$.

We prove first that $\bar{r} \le H$.

Let $\epsilon > 0$. We define D_n to be the set of those $\boldsymbol{a}$ for which the recurrence time $R_n(\boldsymbol{a}) > 2^{n(H+\epsilon)}$, i.e.

$$D_n = \left\{\boldsymbol{a} \in \mathcal{A}^\infty \,:\, R_n(\boldsymbol{a}) > 2^{n(H+\epsilon)}\right\} = \left\{\boldsymbol{a} \in \mathcal{A}^\infty \,:\, \frac{\log R_n(\boldsymbol{a})}{n} > H + \epsilon\right\}.$$

We show that $\boldsymbol{a} \notin D_n$ eventually, almost surely which yields that eventually, almost surely $\frac{\log R_n(\boldsymbol{a})}{n} \leq H + \epsilon$ and thus $\bar{r} \leq H$.

Let

$$T_n = \left\{\boldsymbol{a} \in \mathcal{A}^\infty \,:\, P(a_1^n) \geq 2^{-n(H+\frac{\epsilon}{2})}\right\} = \left\{\boldsymbol{a} \in \mathcal{A}^\infty \,:\, -\frac{\log P(a_1^n)}{n} \leq H + \frac{\epsilon}{2}\right\}.$$

This is the set of so called **entropy typical** sequences. We show that if $\boldsymbol{a} \in T_n$ eventually, almost surely, then $\boldsymbol{a} \notin D_n \cap T_n$ eventually, almost surely. This is sufficient, since the Entropy theorem tells us that $\lim_{n\to\infty} -\frac{\log P(a_1^n)}{n} \leq H + \frac{\epsilon}{2}$, almost surely and thus $\boldsymbol{a} \in T_n$ eventually, almost surely.

Fix an $a_1^n \in \mathcal{A}^n$. We consider only those $\boldsymbol{a} \in D_n$ for which $\boldsymbol{a} \in [a_1^n]$. We denote this set by $D_n(a_1^n) = D_n \cap [a_1^n]$. We now let $\boldsymbol{a} \in D_n(a_1^n)$. The definition of D_n implies that it takes at least $2^{n(H+\epsilon)}$ elements in $\boldsymbol{a}$ before a_1^n reappears. Hence with a shift transformation T, it is true that $(T^j\boldsymbol{a})_1^n \neq a_1^n$ i.e. $T^j\boldsymbol{a} \notin [a_1^n]$, when $1 \leq j \leq 2^{n(H+\epsilon)} - 1$,.

As a consequence, the sets $D_n(a_1^n), T^{-1}D_n(a_1^n), \ldots, T^{-2^{n(H+\epsilon)}-1}D_n(a_1^n)$ are all disjoint. For this reason and the fact that these sets have the same measure it must be that

$$P(D_n(a_1^n)) \leq \frac{1}{2^{n(H+\epsilon)}}.$$

On the other hand, the cardinality of the projection of $D_n(a_1^n) \cap T_n(a_1^n)$ onto $\mathcal{A}^n$ cannot be greater than the cardinality of the projection of $T_n(a_1^n)$ which is at most $2^{n(H+\epsilon/2)}$ by the definition of T_n.

On account of these facts $P(D_n \cap T_n) \leq 2^{-n(H+\epsilon)}2^{n(H+\epsilon/2)} = 2^{-n\epsilon/2}$.

Now we see that

$$\sum_{n=1}^{\infty} P(D_n \cap T_n) \leq \frac{2^{-\epsilon/2}}{1 - 2^{-\epsilon/2}} < \infty,$$

and due to the Borell-Cantelli lemma $\boldsymbol{a} \notin D_n \cap T_n$ eventually, almost surely. This concludes the proof of $\bar{r} \leq H$.

Next we prove that $\underline{r} \geq H$. We assume that $\underline{r} < H - \epsilon$, where $\epsilon > 0$ is arbitrary.

We derive a contradiction by defining first the concept "too-soon-recurrent" and then showing that if our assumption holds, then our sequence x_1^n is too-soon-recurrent almost surely and thus we can construct a code which turn out to be too good.

We say that $a_s^t \subseteq a_1^n$ **recurs too soon** in a_1^n if there exists $k \in [1, 2^{H(t-s+1)}[$ such that $a_s^t = a_{s+k}^{t+k}$ with $s + k \leq n$. If a_s^t recurs too soon in a_1^n, then we call the smallest k for which $a_s^t = a_{s+k}^{t+k}$ **the distance from $\boldsymbol{a_1^n}$ to its next occurrence** in a_1^n.
We let $a_1^n = u_1 V(1) u_2 V(2) \dots u_J V(J) u_{J+1}$ be the concatenation of a_1^n and $m \in \mathbb{Z}_+$, $n \geq m$ and $\delta > 0$. We say that the concatenation is **$(\boldsymbol{\delta}, \boldsymbol{m})$-too soon recurrent** of a_1^n if

i) Each $V(j)$ recurs too soon in a_1^n and $|V(j)| \geq m$.

ii) The sum of lengths of the filler words u_j is at most $2\delta m$ i.e. $\sum_{j=1}^{J+1} |u_j| \leq 2\delta m$.

We now prove that under our assumption a_1^n is (δ, m)-too soon recurrent almost surely.

First we fix m and δ, and define the set $G(n)$ by setting

$$G(n) = \{a_1^n \in \mathcal{A}^n \,:\, a_1^n \text{ has a } (\delta, m)\text{-too soon recurrent representation}\}.$$

Next we define for all $n \in \mathbb{Z}_+$ the set

$$B_n = \left\{\boldsymbol{a} \in \mathcal{A}^\infty \,:\, R_n(\boldsymbol{a}) \leq 2^{n(H-\epsilon)}\right\} = \left\{\boldsymbol{a} \in \mathcal{A}^\infty \,:\, \frac{\log R_n(\boldsymbol{a})}{n} \leq H - \epsilon\right\}.$$

Now, since $\underline{r} = \liminf_{n\to\infty} \frac{1}{n} \log R_n(\boldsymbol{a})$, there exists an M such that the measure of the set $\mathcal{B} = \cup_{n=m}^{M} B_n$ exceeds $1 - \delta$ i.e. $P(\mathcal{B}) > 1 - \delta$.

We let then $I_{\mathcal{B}}$ be the indicator function of $\mathcal{B}$. Now with the measure-preserving transformation T in (Ω, Σ, P) we know by the Birkhoff's ergodic theorem that

$$\frac{1}{n}\sum_{i=1}^{n} I_{\mathcal{B}}(T^{i-1}\boldsymbol{a}) = \int I_{\mathcal{B}}\,\mathrm{d}P = P(\mathcal{B}) > 1-\delta, \text{ almost surely.} \tag{5.14}$$

Consider the interval $C_i = [i, m(i)],$ where

$$\begin{aligned} m(i) &= \min\{s \,:\, s-i+1 > m \text{ and } T^{i-1}\boldsymbol{a} \in B_{s-i+1}\} \\ &= \min\{s \,:\, s-i+1 > m \text{ and } R_{s-i+1}(T^{i-1}\boldsymbol{a}) < 2^{(s-i+1)(H-\epsilon)}\} \end{aligned}$$

and the collection of intervals

$$\mathcal{C} = \{C_i \,:\, i \in \mathbb{Z}_+\},$$

which is a strong cover of $\mathbb{Z}_+$.

Let then $n > \frac{M}{\delta}$. The interval $[1, n]$ is (M, δ)-strongly covered by $\mathcal{C}$, since by the inequality (5.14) there exists at least $(1-\delta)n$ integers $k \in [1, n]$ such that $T^{k-1}\boldsymbol{a} \in \mathcal{B}$ that is $m \le m(k) - k + 1 \le M$, and thus

$$\frac{|\{k \in [1, n] \,:\, m(k) - k + 1 > M|}{n} \le \delta.$$

Now due to the Packing lemma there exists a subcollection

$$\widetilde{\mathcal{C}} = \{[n_i, m(n_i)] \,:\, 1 \le i \le J\}$$

of intervals of $\mathcal{C}$ such that $\widetilde{\mathcal{C}}$ is a $(1-2\delta)$-packing of $[1, n]$. The length of each interval in $\widetilde{\mathcal{C}}$ is at least m and at most M, and since $\widetilde{\mathcal{C}}$ is a $(1-2\delta)$-packing of $[1, n]$, it follows that

$$\sum_{i=1}^{I}(m(n_i) - n_i + 1) \ge (1-2\delta)n \tag{5.15}$$

Now also, since the set B_n is the set of those $\boldsymbol{a} \in \mathcal{A}^{\infty}$ recurrence time of which is less than $2^{n(H-\epsilon)}$ we know by the definition of $m(i)$ that for each $1 \le i \le I,$ there exists a $j \in [1, 2^{(m(n_i)-n_i+1)(H-\epsilon)}[$ such that $a_{n_i}^{m(n_i)} = a_{n_i+j}^{m(n_i)+j}$.

We let now $V(j) = a_{n_i}^{m(n_i)}$ for all $1 \leq j \leq J$. Each block recurs too soon in a_1^n and a_1^n can be written as a concatenation

$$a_1^n = u_1 V(1) u_2 \dots u_J V(J) u_{J+1},$$

where $\sum_{i=1}^{J+1} |u_i| \leq 2\delta n$ by the inequality (5.15). As a result $a_1^n \in G_n$ eventually, almost surely and thus it has $(\delta, m)-$ too-soon-recurrent representation.

We still have to show that since $a_1^n \in G_n$ eventually, almost surely, there exists a too good code which contradicts with Theorem 3.4.

We construct a prefix code $C_n : \mathcal{A}^n \to \mathcal{B}^*$. Let $a_1^n \in G(n)$ and let

$$a_1^n = u_1 V(1) u_2 \dots u_J V(J) u_{J+1}$$

be its too-soon recurrent representation. The codeword $C(a_1^n)$ consists of two different codings. Each filler word u_j is coded one letter at a time with a non-singular code $F : \mathcal{A} \to \{0,1\}^d$, where $d \leq 2 + \log |\mathcal{A}|$ and each codeword starts with a 0. Every $V(j)$ is coded by means of an Elias code $\mathcal{E}$ (see Definition (3.5)). Each codeword starts with 1 followed by the codeword $\mathcal{E}(|(V(j)|)$ which is finally followed by $\mathcal{E}(k_j)$, where k_j is the distance from $V(j)$ to its next occurrence in a_1^n. If $a_1^n \notin G(n)$, then it is coded just by using the code F for each letter. The first bits 0 and 1 before codewords of u_j and $V(j)$ ensure that C_n is a prefix code and they also determine which one of the codes is used.

We now show that if $a_1^n \in G_n$, then for $n \geq m$

$$\mathcal{L}(a_1^n) \leq n(H - \epsilon) + n(2d\delta + \alpha_m),$$

where $\lim_{m\to\infty} \alpha_m = 0$. This leads to the existence of a too good code.

The codeword of a filler word u_j needs $d|u_j|$ bits and since $\sum_{j=1}^{J+1} |u_j| \leq 2n\delta$ the codewords of filler words need together at most $2nd\delta$ bits.

The sequence a_1^n can have at most $\frac{n}{m}$ $V(j)$s since each $|V(j)| \geq m$ and thus at most $\frac{n}{m}$ bits are needed for the 1s in the beginning of each codeword. Further, the codeword $\mathcal{E}(k_j)$ needs $\log(k_j) + o(k_j)$ bits. On the other hand, $V(j)$ recurs too soon and thus $k_j \leq 2^{(H-\epsilon)|V(j)|}$ and at most

$$((H-\epsilon)|V(j)|) + o((H-\epsilon)|V(j)|)) = ((H-\epsilon)|V(j)|) + o(|V(j)|)$$

bits are needed to code k_j. Again $\sum_{j=1}^{J} V(j) \leq n$ and if we define β_m to be an upper bound of $\frac{o(|V(j)|}{n}$, for which $\beta_m \to 0$ as $m \to \infty$, then the sum of lengths of codes of k_j takes at most

$$n(H-\epsilon) + n\beta_m$$

bits. The codeword $\mathcal{E}(|(V(j)|)$ needs

$$\log(|V(j)|) + o(\log(|V(j)|)$$

bits. Let t_i $(1 \leq i \leq M-m)$ be the number of $V(j)$s having length $m+i$. Taking sum of the first term over all j we get

$$\begin{aligned}
&\sum_{j=1}^{J} \log(|V(j)|) \\
&\leq \tfrac{n-t_1(m+1)-t_2(m+2)-\cdots-t_{M-m}M}{m} \log m \\
&+t_1 \log(m+1) + t_2 \log(m+2) + \cdots + t_{M-m} \log(M) \\
&= \tfrac{n}{m} \log(m) + t_1 \left(\log(m+1) - \log(m) - \tfrac{\log(m)}{m} \right) + \\
&t_2 \left(\log(m+2) - \log(m) - 2\tfrac{\log(m)}{m} \right) + \cdots + \\
&t_{M-m} \left(\log(m+s) - \log(m) - (M-m)\tfrac{\log(m)}{m} \right) \\
&\leq \tfrac{n}{m} \log(m).
\end{aligned}$$

The last step follows since if $m \geq 3$ (which holds since $m \to \infty$) for all $l \geq 1$, $m+l < m^{1+\frac{l}{m}}$ and because of that $\log\left(\frac{m+l}{m^{1+\frac{1}{m}}}\right) < 0$ i.e. $\log(m+l) - \log(m) - l\frac{\log m}{m} < 0$.

Since also $o(\log(|V(j)|) \leq o(|V(j)|)$, in total at most

$$\frac{n}{m} + n(H - \epsilon) + n\beta_m + n\frac{\log(m)}{m} + n\beta_m$$

bits are needed to the codes of blocks $V(j)$, and thus

$$\mathcal{L}(a_1^n) \leq n(H - \epsilon) + n(2d\delta + \alpha_m),$$

where

$$\alpha_m = 2\beta_m + \left(\frac{1 + \log m}{m}\right),$$

and $\alpha_m \to 0$, as $m \to \infty$.

Now, if m is large enough and $\delta < \frac{\epsilon}{4d}$, we get that on $G(n)$

$$\mathcal{L}(a_1^n) \leq n(H - \epsilon) + n(2d\delta + \alpha_m) \leq n(H - \epsilon/2).$$

But by the Theorem 3.4 there are no too good codes i.e. $\mathcal{L}(a_1^n) \geq nH$ always. This again means that the measure of $G(n)$ must go to 0. Further this is a contradiction, since we have proven that $a_1^n \in G(n)$ eventually, almost surely, which means that our assumption $\underline{r} < H - \epsilon$ is false and thus $\underline{r} \geq H$. This completes the proof of recurrence time theorem. □

Remark 20. Recurrence time theorem was proved first only in context of probability. Some parts of it were proven in almost sure form by Aaron D. Wyner and Ziv in [21] in 1989 but the whole proof in almost sure form was introduced first by Donald Samuel Ornstein and Benjamin Weiss in [14] in 1993.

5.2 Other results related to recurrence time

In this subchapter we give more theorems in which the recurrence time plays an important role. We start with Kac's lemma which M. Kac proved in 1947.

Remark 21. We use the abbreviation R_n for recurrence time, when there is no danger of misunderstanding.

Lemma 5.1 (Kac's lemma)**.** *Let $S = \{X_n\}$ be a stationary, ergodic source. If the length of a window is N_0, then the expected recurrence time can be bounded by*

$$E[R_n] \leq \frac{1}{P(a_1^n)}.$$

The equality is achieved as $N_0 \to \infty$.

Proof. Cf. [20], [21].
Let $a_1^n \in \mathcal{A}^n$ and $k \in \mathbb{Z}_+$. Define $Q_k(a_1^n)$ as the probability that recurrence time of a_1^n is k, i.e.

$$Q_k(a_1^n) = P(X_{k+1}^{k+n} = a_1^n,\, X_{j+1}^{j+n} \neq a_1^n, 1 \leq j \leq k-1 \,:\, X_1^n = a_1^n). \quad (5.16)$$

Define also the average recurrence time $\nu(R(a_1^n))$ by setting

$$\nu(R(a_1^n)) = \sum_{i=1}^{\infty} k\, Q_k(a_1^n).$$

We define then the event

$$D = \{X_{l+1}^{l+n} = a_1^n \,:\, -\infty \leq l \leq \infty\},$$

and events

$$\begin{aligned} B_+ &= \{X_{l+1}^{l+n} = a_1^n \,:\, 0 \leq l \leq \infty\} \quad \text{and} \\ B_- &= \{X_{l+1}^{l+n} = a_1^n \,:\, -\infty \leq l \leq -1\}. \end{aligned}$$

The event D can be expressed by means of the events B_+ and B_- as

$$D = (B_+ \cap B_-) \cup (B_+ \cap \bar{B}_-) \cup (\bar{B}_+ \cap B_-),$$

where $B_+ \cap B_-$, $B_+ \cap \bar{B}_-$ and $\bar{B}_+ \cap B_-$ are clearly disjoint events. We show next that $P(B_+ \cap \bar{B}_-) = P(\bar{B}_+ \cap B_-) = 0$, and thus $P(D) = P(B_+ \cap B_-)$.

We assume that the event $B_+ \cap \bar{B}_-$ occurs which means that

$$P(B_+ \cap \bar{B}_-) = \sum_{i=0}^{\infty} P(X_{l+1}^{l+n} \neq a_1^n, -\infty < l < i, X_{i+1}^{i+n} = a_1^n) > 0.$$

This means that there exists the smallest $j \geq 0$ such that
$P(X_{l+1}^{l+n} \neq a_1^n, -\infty < l < j, X_{j+1}^{j+n} = a_1^n) > 0$. Now

$$\begin{aligned}
&P(X_{l+1}^{l+n} \neq b, -\infty < l < j, X_{j+1}^{j+n} = a_1^n) \\
&= P(X_{l+1}^{l+n} \neq a_1^n, -\infty < l < j-1, X_j^{j-1+n} \neq a_1^n) \\
&\quad -P(X_{l+1}^{l+n} \neq a_1^n, -\infty < l < j, X_{j+1}^{j+n} \neq a_1^n). \qquad (5.17)
\end{aligned}$$

On the other hand, we know that $\{X_n\}$ is stationary and thus

$$\begin{aligned}
&P(X_{l+1}^{l+n} \neq a_1^n, -\infty < l < j-1, X_j^{j-1+n} \neq a_1^n) \\
&= \lim_{l \to -\infty} P(X_{l+1}^{l+n} \neq a_1^n, X_{l+2}^{l+1+n} \neq a_1^n, \ldots, X_j^{j-1+n} \neq a_1^n) \\
&= \lim_{l \to -\infty} P(X_{l+2}^{l+1+n} \neq a_1^n, X_{l+3}^{l+2+n} \neq a_1^n, \ldots, X_{j+1}^{j+n} \neq a_1^n) \\
&= P(X_{l+1}^{l+n} \neq a_1^n, -\infty < l < j, X_{j+1}^{j+n} \neq a_1^n).
\end{aligned}$$

Because of this $P(B_+ \cap \bar{B}_-) = 0$ and it is impossible that the event $B_+ \cap \bar{B}$ to occurs. The impossibility of the event $\bar{B}_+ \cap B_-$ can be established similarly and hence we have proven that $P(D) = P(B_+ \cap B_-)$.

Now we get the probability of D as follows

$$P(D) = \sum_{\substack{i=0 \\ j=1}}^{\infty} P(X_{i+1}^{i+n} = a_1^n, X_{-j+1}^{-j+n} = a_1^n, X_{l+1}^{l+n} \neq a_1^n, -j+1 \leq l \leq i-1) =$$

$$\sum_{\substack{i=0 \\ j=1}}^{\infty} P(X_{-j+1}^{-j+n} = a_1^n) P(X_{i+1}^{i+n} = a_1^n, X_{l+1}^{l+n} \neq a_1^n, -j+1 \leq l \leq i-1 : X_{-j+1}^{-j+n} = a_1^n). \qquad (5.18)$$

As a result of the definition (5.16) and stationarity, the expression (5.18) is equal to

$$\sum_{\substack{i=0\\j=1}}^{\infty} P(X_1^n = a_1^n) P(X_{i+j+1}^{i+j+n} = a_1^n, X_{l+1}^{l+n} \neq a_1^n, 0 \leq l \leq i+j-1 : X_1^n = a_1^n) =$$

$$= \sum_{\substack{i=0\\j=1}}^{\infty} P(X_1^n = b) Q_{i+j}(a_1^n). \tag{5.19}$$

Now for each $k = i + j \geq 0$, Q_k occurs in sum (5.19) k times (see the table below)

i	0	1	$\cdots$	$k-1$
j	k	$k-1$	$\cdots$	1

.

Hence $P(D)$ can be written as

$$P(D) = P(X_1^n = a_1^n) \sum_{k=1}^{\infty} k\, Q_k(a_1^n) = P(X_1^n = a_1^n) \nu(a_1^n).$$

By the ergodicity of the source, we get that $P(D) = 1$, and it follows that

$$\nu(a_1^n) = \frac{1}{P(X_1^n = a_1^n)}. \tag{5.20}$$

The expected recurrence time of each a_1^n is

$$\begin{aligned} E(R_n(a_1^n)) &= \sum_{k=1}^{N_0} k\, Q_k(a_1^n) + \sum_{i=N_0+1}^{\infty} N_0 Q_k(a_1^n) \\ &\leq \sum_{k=1}^{\infty} k\, Q_k(a_1^n) = \nu(a_1^n). \end{aligned} \tag{5.21}$$

Therefore the equality (5.20) and the inequality (5.21) lead to the result

$$E[R_n] \leq \frac{1}{P(X_1^n = a_1^n)}.$$

If $N_0 \to \infty$, then the equality in (5.21) (and in the result) is achieved.

Thus we have proven Lemma 5.1. □

The next theorem shows that if the source is "sufficiently good", then there is a code the expected code length of which is near the entropy of the source.

Theorem 5.2 (Recurrence time coding theorem)**.** *Let $\delta > 0$ be arbitrarily small and S a stationary, ergodic source with an alphabet $|\mathcal{A}| = 2$. For any $S > 0$, let T_S be the set defined by*

$$T_S = \{x_1^n \,:\, P(x_1^n) < 2^{-Sn}\}.$$

Define also

$$B_n = \min[S \,:\, P(T_S) \leq \delta].$$

Now for N_0 sufficiently large and for any n such that $B_n \leq \frac{\log N_0}{n} - \delta$ there is a coding C with a window $X_{-N_0+1}^0$ for which

$$\frac{1}{l} E[\mathcal{L}(C(X_1^n \,|\, X_{-N_0+1}^0))] \leq H_n(X_1^n) + O\left(\frac{\log\log N_0}{n}\right) + 2^{-n\delta} + \delta.$$

Proof. Cf. [23].

Take the code C presented in the proof of Lemma 3.2. Then let N_0 be large enough so that for all $4 \leq N_0$, it holds that $\mathcal{L}(C(N_0)) \leq \log N_0 + O(\log\log N_0)$ (see Lemma 3.2). Now let a code $C^* : \mathcal{A}^n \to \mathcal{B}^*$ code a sequence of length n with a window $X_{-N_0+1}^0$ such that there is first a "yes-no" flag which tells whether the block X_1^n occurs in the window. If it occurs, then the code codes the recurrence time R_n and if the block does not occur, then X_1^n is coded just by its binary representation. Now by Lemma 3.2

$$\mathcal{L}(C^*(X_1^n)) \leq \begin{cases} \log R_n + O(\log\log N_0), & \text{if } R_n \leq N_0 \\ n, & \text{otherwise.} \end{cases}$$

We let now n be such that $B_n \leq \frac{\log N_0}{n} - \delta$. Now if $X_1^n \notin T_{B_n}$ and $R_n \leq N_0$,

then the length of the codeword is at most $\log R_n + O(\log\log N_0)$ bits. In other cases it takes at most n bits to code X_1^n. Thus we get

$$\begin{aligned}\frac{1}{n}E[\mathcal{L}(C^*(X_1^n)|X_{-N_0+1}^0] = \quad & E\left[\frac{\log R_n + O(\log\log N_0)}{n}\right] \\ & + P\{X_1^n \notin T_{B_n}, R_n > N_0\} + P\{X_1^n \in T_{B_n}\}.\end{aligned} \tag{5.22}$$

Now by the Kac's lemma, we know that

$$E[R_n] \leq \frac{1}{P(X_1^n)},$$

and thus

$$\frac{E[\log R_n]}{n} = \frac{1}{n}\sum_{a_1^n \in \mathcal{A}^n} P(a_1^n)\log R_n \leq \frac{1}{l}\sum_{a_1^n \in \mathcal{A}^n} P(a_1^n)\log\frac{1}{P(a_n^1)} = H_l(X_1^n).$$

Again by the definition of T_{B_n} and B_n, we get that

$$P\{X_1^n \in T_{B_n}\} \leq \delta.$$

If $X_1^n \notin T_{B_n}$, then it follows that

$$P(X_1^n) \geq 2^{-B_n n} \geq 2^{-\left(\frac{\log N_0}{l} - \delta\right)n} = \frac{2^{\delta n}}{N_0} \tag{5.23}$$

and from the Markov inequality it follows that

$$P\{R_n > N_0\} \leq \frac{E[R_n]}{N_0},$$

and hence using first the Kac's lemma and then the inequality (5.23), we get

$$P\{X_1^n \notin T_{B_n}, R_n > N_0\} \leq \max_{X_1^n \notin T_{B_n}} \frac{E[R_n]}{N_0} \leq \max_{X_1^n \notin T_{B_n}} \frac{1}{P(X_1^n)N_0} \leq \frac{N_0}{N_0 2^{\delta n}}.$$

Thus we get from the expression (5.22) that

$$\frac{1}{n}E[\mathcal{L}(C^*(X_1^n))|X_{-N_o+1}0] \leq H_n(X_1^n) + O\left(\frac{\log\log N_0}{n}\right) + 2^{-\delta n} + \delta.$$

□

The last theorem in the thesis is about properties of recurrence time.

Theorem 5.3. *Let $\{X_n\}$ be a stationary, ergodic, finite-valued process. Let also $\{c_n\}$ be a sequence, such that $c_n \geq 0$ and $\sum_{n=1}^{\infty} n2^{-c_n} < \infty$. Now*

i) $\log[R_n P(X_1^n)] \leq c_n$ *eventually, almost surely, and*

ii) $\log[R_n P(X_1^n \,|\, X_{-\infty}^0)] \geq -c_n$ *eventually, almost surely.*

Proof. Cf. [13].

i) In the proof of Theorem 5.2 we have already seen that from the Kac's lemma and the Markov inequality it follows that

$$P(R_n > K \,|\, X_1^n = a_1^n) \leq \frac{1}{KP(a_1^n)}. \tag{5.24}$$

Now $P(a_1^n)$ is a constant relative to $P(\cdot \,|\, X_1^n = a_1^n)$ and thus if we let $K = \frac{2^{c(n)}}{P(a_1^n)}$, we get from the inequality (5.24)

$$\begin{aligned} &P\left(R_n > \frac{2^{c(n)}}{P(a_1^n)} \,|\, X_1^n = a_1^n\right) \\ &= P\left(\log[R_n P(X_1^n)] > c(n) \,|\, X_1^n = a_1^n\right) \leq \frac{1}{2^{c(n)}}. \end{aligned} \tag{5.25}$$

Now since $\sum_{n=1}^{\infty} P(C_n) \leq \sum_{n=1}^{\infty} \frac{1}{2^{c(n)}} < \infty$ by the inequality (5.25) and as we define the set C_n by $C_n = \{a_1^n : \log[R_n P(a_1^n)] > c(n)\}$, the Borel Cantelli lemma gives that $x_1^n \notin C_n$ eventually, almost surely and hence $\log[R_n P(X_1^n)] \leq c_n$ eventually, almost surely.

ii) We fix $a_{-\infty}^0$, and set

$$G_n = G_n(a_{-\infty}^0) = \left\{ b_1^n \in \mathcal{A}^n \,:\, P(b_1^n \,|\, a_{-\infty}^0) < \frac{2^{-c(n)}}{R_n(a_{-\infty}^0 * b_1^n)} \right\},$$

where $a^0_{-\infty} * b^n_1 = \ldots a_{-1}a_0b_1b_2\ldots b_n$. Now

$$\begin{aligned}
&P\{\log[R_n(X)P(X_1^n \,|\, X^0_{-\infty} = a^0_{-\infty})] < -c(n) \,|\, X^0_{-\infty} = a^0_{-\infty}\} \\
&= P\left\{b_1^n \in \mathcal{A}^n \,:\, P(X_1^n = b_1^n \,|\, X^0_{-\infty}) < \frac{2^{-c(n)}}{R_n(a^0_{-\infty} * b_1^n)} \,|\, X^0_{-\infty} = a^0_{-\infty}\right\} \\
&= \sum_{b_1^n \in G_n} P(b_1^n \,|\, a^0_{-\infty}) \\
&\leq \sum_{b_1^n \in G_n} \frac{2^{-c(n)}}{R_n(a^0_{-\infty} * b_1^n)} \\
&\leq 2^{-c(n)} \sum_{b_1^n \in \mathcal{A}^n} \frac{1}{R_n(a^0_{-\infty} * b_1^n)}. \qquad (5.26)
\end{aligned}$$

Now for fixed $a^0_{-\infty}$ there exists exactly one $b_1^n \in \mathcal{A}^n$ such that $R_n(a^0_{-\infty} * b_1^n) = j$, for each $1 \leq j \leq |\mathcal{A}|^n$. As a result we get from the inequality (5.26)

$$\begin{aligned}
&2^{-c(n)} \sum_{b_1^n \in \mathcal{A}^n} \frac{1}{R_n(a^0_{-\infty} * b_1^n)} \\
&\leq 2^{-c(n)} \sum_{j=1}^{|\mathcal{A}|^n} \frac{1}{j} \leq 2^{-c(n)} E_n n, \qquad (5.27)
\end{aligned}$$

where $E_n > 0$ is a constant.

Let $D_n = \{a_1^n \,:\, \log[R_n P(X_1^n \,|\, X^0_{-\infty} = a^0_{-\infty})] < -c(n) \,|\, X^0_{-\infty} = a^0_{-\infty}\}$, since $\sum_{n=1}^{\infty} P(D_n) \leq \sum_{n=1}^{\infty} E_n n \frac{1}{2^{c(n)}} < \infty$ by (5.27), the Borell Cantelli lemma gives that eventually, almost surely $a_1^n \notin D_n$ and hence eventually, almost surely $\log[R_n P(X_1^n \,|\, X^0_{-\infty})] \geq -c_n$.

This completes the proof of Theorem 5.3. □

6 Using recurrence time for analysing DNA properties

In this chapter we consider analysing DNA (deoxyribonucleic acid) sequences using the theory developed in previous chapters. DNA and other biological sequences have a big importance in nowadays biology and the amount and their lengths are increasing rapidly. Thus it is important to be able to compress efficiently such sequences. Compression of DNA is essentially compression of text because we can think DNA as a specific kind of text built up on an alphabet $\mathcal{A} = \{A, C, G, T\}$. These letters signify the bases adenine, cytosine, guanine and thymine. Many algorithms for compressing DNA have been proposed but many of them fail more or less since statistical properties of DNA are hard to find and DNA sequences seem to be almost random. However, althugh the probabilities of individual bases are quite similar, if longer sequences are investigated, then the situation changes. [10]

For those who are interested in biological properties of DNA we recommend the book Bruce Alberts & al.: "Essential cell biology: an introduction to the molecular biology of the cell" (2002, Garland) and if mathematical properties and methods of DNA are of interest, there is Michael S. Waterman's book "Mathematical Methods for DNA sequences" (1989, CRC Press) which gives a quite good summary of different methods.

LZ algorithms have also been applied to DNA but these efforts have not been very successful. Statistical methods have fared better but still the compression is not that good. In [10] a combination of statistical and LZ method is introduced, Biocompress-2 (Biocompress-1 has been published earlier). This method seem to compress biological sequences quite well. [10]

Our goal in the following is to test the Kac's lemma (Lemma 5.1) with real DNA sequences. As the sequence, we use the human chromosome 22 which

is the first sequenced human chromosome. The chromosome has in total $48 \cdot 10^6$ bases and the sequenced parts contain $33,4 \cdot 10^6$ bases. The rest are not stable. The chromosome 22 covers only 1,5% of the total human genome (the genomic information of human which DNA contains) which is in total $3,2 \cdot 10^9$ bases long, the other 23 chromosomes containing the rest of DNA. The human chromosome 22 is quite repetitive i.e. the same or almost same sequence is repeated one or several times in the chromosome. The lengths of the repeats vary from couple to thousands bases. In total, repeats cover about 41,91 % of the chromosome. This encourages us to believe that the recurrence time of short sequences cannot be very long. [2, pages 169-179, 311-313], [7]

We loaded the sequence of chromosome 22 from Gen bank [9]. On total, the chromosome is about 33 millions bases long and it is organized in 11 parts. Since the fourth part is about 22 millions bases long, the memory of our computer does not suffice for calculations with this long sequence, and we split this part to three parts, so in total we have 13 parts of length 200 000 to 7 500 000 bases.

We have assumed that the sequence of chromosome is stationary and ergodic. We computed the recurrence time for the blocks of length $k \in \{4, 6, 8, 10\}$ with a window of length $N_0(k)$ with $N_0(4) = 100000, N_0(6) = 300000$, $N_0(8) = 700000, N_0(10) = 1500000$. We computed the recurrence time for all overlapping blocks i.e. with blocks of length 4 if the sequence was $a_1 a_2 \dots a_n$ then in the first time the window was $a_1 a_2 \dots a_{100000}$ and the block recurrence time of which was investigated was $a_{100001} \dots a_{100004}$ and in the next checking the window was $a_2 a_3 \dots a_{100001}$ and the block was $a_{100002} \dots a_{100005}$. With block lengths 4,6 and 8 almost all blocks were found in the window and thus these cases are (almost) similar to the case with an infinite length of a window. With length 10, only 83,8 % of blocks were found (and the recurrence time of the remaining block was then marked as 1 500 000).

After we had gotten all recurrence times collected we estimated the expected recurrence time of each block $a_1^k \in \mathcal{A}^k$ as the average of the recurrence times (we denote these with $\dot{R}(a_1^k)$). In Figure 2 we can see the histograms of recurrence times $\dot{R}(a_1^k)$. We also collected the frequencies of the blocks and

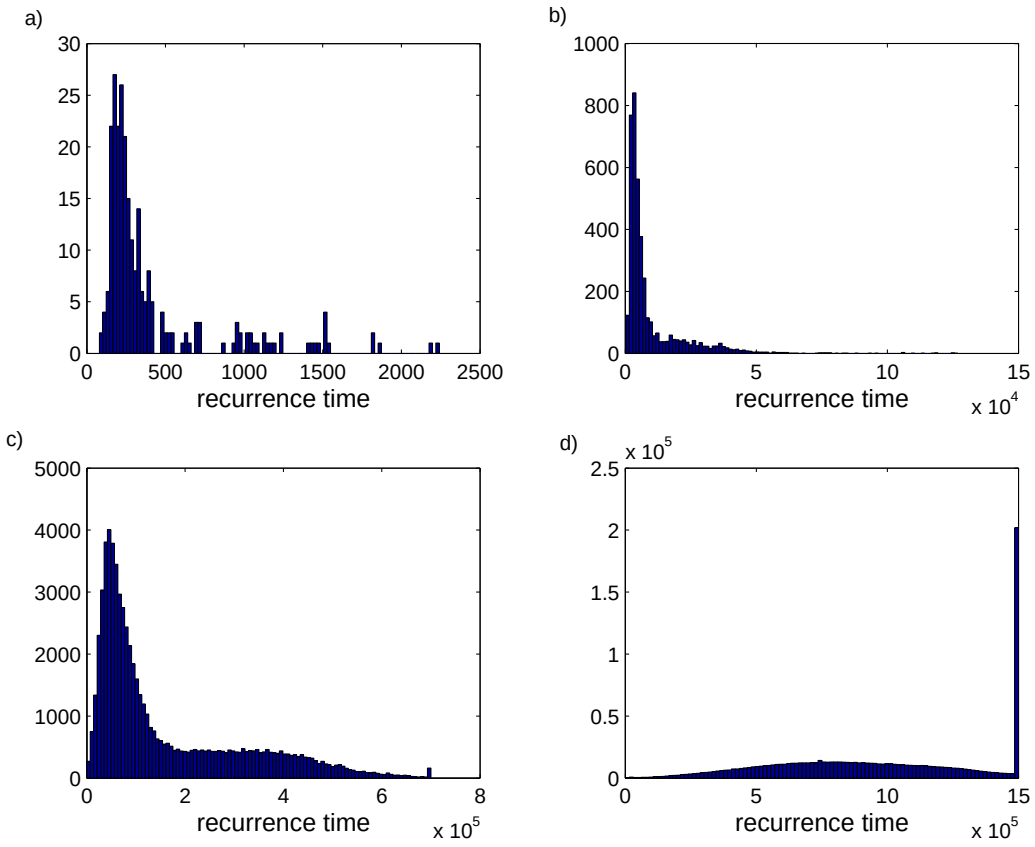

Figure 2: Histograms of recurrence times with the block length a) k=4, b) k=6, c) k=8 and d) k=10

then computed the empirical probabilities of blocks $\widehat{P}(a_1^k)$. After this we

used the Kac's lemma for computing the expected recurrence time $\widehat{R}$ of each a_1^k with the formula

$$\widehat{R}(a_1^k) = \min\left\{N_0, \frac{1}{\widehat{P}(a_1^k)}\right\}.$$

After this we collected the empirical Markov probabilities of order 3 of the chromosome 22 and the initial empirical probabilities (i.e. probabilities $\widetilde{P}_M(A|AAA), \widetilde{P}_M(C|AAA), \ldots, \widetilde{P}_M(T|TTT)$ and $\widetilde{P}_M(AAA), \widetilde{P}_M(AAC)(\ldots$. Then we computed the probability of each n-block as

$$\widetilde{P}_M(a_1a_2\ldots a_n) = \widetilde{P}_M(a_1a_2a_3)\prod_{i=4}^{n}\widetilde{P}_M(a_i|a_{i-3}a_{i-2}a_{i-1}).$$

Then we computed the expected recurrence time $\widetilde{R}_M$ of each block again by using the Kac's lemma. In figure 3 there are the histograms of recurrence times $\widehat{R}$ and $\widetilde{R}_M$ with the block length 8. As we can see, by computing there are much more blocks with recurrence time of 700 000 than with observed sequence (i.e. those blocks could not be found in the past).

In the end we computed the expected recurrence time of random variable X using the three different models:

$$\begin{aligned}
E[\dot{R}(X_k)] &= \sum_{a_1^k\in\mathcal{A}^k}\dot{R}(a_1^k)\widehat{P}(a_1^k)\\
E[\widehat{R}(X_k)] &= \sum_{a_1^k\in\mathcal{A}^k}\widehat{R}(a_1^k)\widehat{P}(a_1^k)\\
E[\widetilde{R}_M(X_k)] &= \sum_{a_1^k\in\mathcal{A}^k}\widetilde{R}_M(a_1^k)\widetilde{P}_M(a_1^k).
\end{aligned}$$

The results are summed together in the Table 1. With the block lengths 4 and 6 it is natural that $E[\widehat{R}(X_4)] = E[\widetilde{R}_M(X_4)] = 256$ and $E[\widehat{R}(X_6)] = E[\widetilde{R}_M(X_6)] = 4096$ which is the number of different blocks since the past was so long that $\widehat{R}(a_1^k)$ and $\widehat{P}(a_1^k)$ were always inverses. The observed recurrence

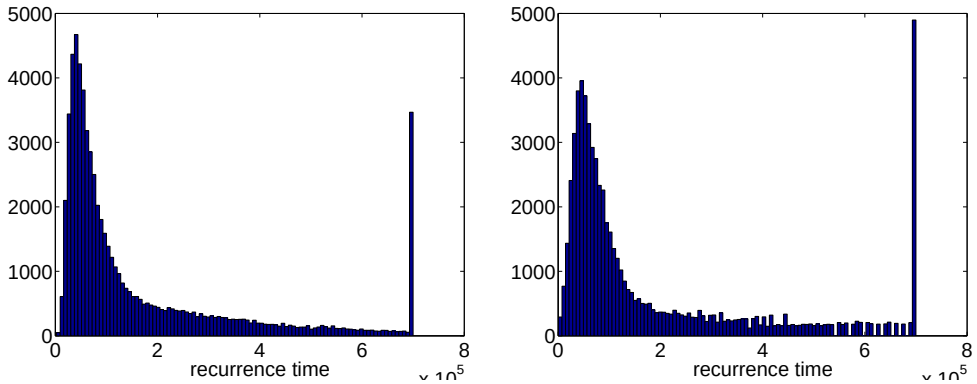

Figure 3: Histograms of recurrence times computed with Kac's lemma with the block length k=8 when we use a) estimated probabilities using a 0 order model b) probabilities of a Markov model of order 3

time is also very close to computed values and we can say that Kac's lemma holds. When we have longer blocks we can see that $E[\dot{R}(X_k)] < E[\widehat{R}(X_k)] < E[\widetilde{R}_M(X_k)]$. The latter inequality is probably because the Markov probabilities do not "take into account" so clearly that some blocks are more general than others and we think that the first one is a result of our assumption of stationarity and ergodicity. The DNA contains both coding and noncoding regions which have different structure. Noncoding regions have far more short repeats and thus our assumptions of stationarity and ergodicity does not hold [2, pages 169-179, 311-313].

An interesting trial would be test the Recurrence time theorem with real data. However, we have seen in previous paragraphs that the longer a block is, the smaller is the possibility of finding it again in a window. We would need to use far longer window as we did this time, and the efficiency of a standard desktop computer would not be enough for our simple algorithms. Since the focus of the thesis is not on efficient implementation of algorithms we could

Block length k	$N_0(k)$	% found in window	$E[\dot{R}(X_k)]$	$E[\widehat{R}(X_k)]$	$E[\widetilde{R}_M(X_k)]$
4	100000	100	256,1	256	256
6	300000	99,997	4112	4096	4096
8	700000	99,07	60510	63900	64330
10	1500000	83,81	551300	672800	721000

Table 1: Expected recurrence times

not test this theorem. Another impediment in testing the Recurrence time theorem is that estimating the entropy of DNA is not a simple problem. In [8] it is assumed that if the stationarity of DNA is assumed (and also that DNA is a random process), then the entropy estimates can be very poor.

7 Conclusions

In the thesis we have studied mathematical properties of recurrence time and also taken a look at data coding. Since the theorems have different kinds of assumptions (stationarity, ergodicity), they do not exactly hold in real cases (as we see in Chapter 6). However, most mathematical results are applicable as good models for the real world data.

As we have mentioned earlier, recurrence time can be an useful tool in data compression. We have confined ourselves to examining the recurrence time with no distortion, which is used in lossless compression, but there are also many research on the recurrence time, when small distortion of the investigated block in a window is allowed. The results of the recurrence time with distortion are used inter alia in lossy data compression. There is also a concept of waiting time which has many similar or almost similar properties as

recurrence time. When recurrence time is the time which it takes for some block to reappear in the sequence, waiting time is the time for a block appears the first time. If reader is interested in lossy compression and waiting time, these are investigated for instance in [3] and [13].

References

[1] Abrahamson Norman, *Information theory and coding*, McGraw-Hill Book Company Inc., New York, 1963

[2] Alberts Bruce, & al.: *Essential cell biology: an introduction to the molecular biology of the cell*, Garland Science, USA, 2002

[3] Andreasen Peter, *Universal Source Coding*, Master of Science thesis, University of Copenhagen, 2001

[4] Bell Timothy C., Cleary John G., Wilter Ian H., *Text Compression*, Prentice- Hall, inc. New Jersey, 1990

[5] Billingsley Patrick, *Ergodic Theory and Information*, John Wiley & sons, inc. New York, 1965

[6] Cover Thomas M., Thomas Joy A., *Elements of Information Theory*, John Wiley & sons, inc. New York, 1991

[7] Dunham I., Shimizu N., Roe B. A, Chissoe S. et al., The DNA Sequence of Human Chromosome 22, *Nature*, 402 (1999), 489-495

[8] Farach Martin, Noordewier Michiel, Savari Serap, Shepp Lary, Wyner Abraham, Ziv Jakob, On the entropy of DNA: Algorithms and measurements based on memory and rapid convergence, In *Proceedings of the 6th Annual Symposium on Discrete Algorithms (SODA95)*, ACM Press, 1994

[9] Gen bank, Human genome resources `http://www.gl.iit.edu /frame/genbank.htm` (accessed November 27, 2003)

[10] Grumbach Stéphane, Tahi Fariza, A New Challenge for Compression Algorithms: Genetic Sequences, *Information Processing & Management*, 30 (1994), 875-886

[11] Huuhtanen Pentti, Kallinen Arto, *Matemaattinen tilastotiede*, Tampereen yliopisto, Matemaattisten tieteiden laitos, Tampere, 1992 (in Finnish)

[12] Kemeny John G., Snell J. Laurie, *Finite Markov Chains*, D. Van Nostrand Company, inc, Princeton, New Jersey 1965

[13] Kontoyiannis Ioannis, *Recurrence and Waiting Times in Stationary Processes, and Their Applications in Data Compression*, Doctor of Philosophy thesis, Stanford University, May 1998

[14] Ornstein Donald Samuel, Weiss Benjamin, *Entropy and Data Compression Schemes*, IEEE Transactions on Information Theory, 39 (1993), 78-83

[15] Ross Sheldon, *Stochastic Processes*, John Wiley & sons, inc. New York, 1996

[16] Roussas George G., *A First Course in Mathematical Statistics*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1993

[17] Shields Paul C., *The Ergodic Theory of Discrete Sample Paths, AMS Graduate Studies in Mathematics*, American Mathematical Society, Providence, 1996

[18] Weisstein Eric W., *CRC Concise Encyclopedia of Mathematics, second edition*, Chapman & Hall/CRC, 2003

[19] Wikipedia contributors, "GIF" Wikipedia: The Free Encyclopedia, `http://en.wikipedia.org/wiki/GIF` (accessed February 1, 2005)

[20] Willems Frans M. J., Universal Data compression and Repetition Times, *IEEE Transactions on Information Theory*, 35 (1989), 54-58

[21] Wyner Aaron D., Ziv Jakob, Some Asymptotic Properties of the Entropy of a Stationary Ergodic Data Source with Applications to Data Compression, *IEEE Transactions on Information Theory*, 35 (1989), 1250-1258

[22] Wyner Aaron D., Ziv Jakob, The Sliding-Window Lempel-Ziv Algorithm is Asymptotically Optimal, *Proceedings of the IEEE*, 82 (1994),872-873

[23] Wyner Aaron D., Ziv Jakob, Wyner Abraham J., On the Role of Pattern Matching in Information Theory, *IEEE Transactions on Information Theory*, 44 (1998), 2045-2056

[24] Ziv Jakob, Lempel Abraham, A Universal algorithm for Sequential Data Compression, *IEEE Transactions on Information Theory*, 23 (1977), 337-343

[25] Ziv Jakob, Lempel Abraham, Compression of Individual Sequences via Variable-rate Coding, *IEEE Transactions on Information Theory*, 24 (1978), 530-536