UPPSALA
UNIVERSITET

Palindromes

Never odd or even

Per Starbäck

Uppsala University
Department of Linguistics and Philology
Master Programme in Language Technology
Master's Thesis in Language Technology, 30 ECTS credits

August 16, 2019

Supervisors:
Prof. Balthazar, Uppsala University
Some Other Supervisor, NLP Enterprises, Inc.

Abstract

The concept of *palindromes* is introduced, and some method for finding palindromes is developed.

Contents

Preface 4

1. Introduction 5

2. Previous work 6

3. Results 7

A. Code 8

Preface

I want to thank Donald Knuth for making TEX, without which I wouldn't have written this.

1. Introduction

Palindromes are fun. I've tried to find some. In Chapter 2 previous work is reviewed, and Chapter 3 is about my results.

2. Previous work

The longest palindromic word in the O*xford English Dictionary* is the onomatopoeic *tattarrattat*, coined by Joyce (1922) for a knock on the door. There is a growing literature where lots of palindromes are collected (Bergerson, 1973; Chism, 1992), and Joki (2015) lists some surprisingly funny ones.

In computation theory the *palindromic density* of an infinite word w over an alphabet A is defined to be zero if only finitely many prefixes are palindromes; otherwise, letting the palindromic prefixes be of lengths n_k for $k = 1, 2, \ldots$ we define the density to be

$$d_P(w) = \left(\limsup_{k \to \infty} \frac{n_{k+1}}{n_k} \right)^{-1} \tag{2.1}$$

Among aperiodic words, the largest possible palindromic density is achieved by the Fibonacci word, which has density $1/\varphi$, where φ is the Golden ratio (Adamczewski and Bugeaud, 2010, p. 443).

3. Results

I examined a list of first names, and found a few there: *Anna*, *Hannah* and *Otto*.

I have also made a program that searches for anagrams. The full program is listed in appendix A on page 8.

A. Code

```
#! /usr/bin/python3

def palindrome(word):
    reverse = word[::-1]
    return word.lower() == reverse.lower()

def findem(minlength=4):
    for word in open("/local/dict/scowl.txt"):
        word = word.rstrip()    # remove newline at end
        if len(word) >= minlength and palindrome(word):
            print(word)

findem()
```

Bibliography

Adamczewski, Boris and Yann Bugeaud (2010). "Transcendence and diophantine approximation". In: *Combinatorics, automata, and number theory, Encyclopedia of Mathematics and its Applications*. Ed. by Valérie Berthé and Michael Rigo. Chap. 8.

Bergerson, Howard W. (1973). *Palindromes and Anagrams*. Dover Publications.

Chism, Stephen J. (1992). *From A to Zotamorf: The Dictionary of Palindromes*. Word Ways Press. ISBN: 978-1565121096.

Joki, Kimberly (2015). "16 Surprisingly Funny Palindromes". Aug. 10, 2015. URL: https://www.grammarly.com/blog/16-surprisingly-funny-palindromes/ (visited on 2019-08-16).

Joyce, James (1922). *Ulysses*. Paris: Sylvia Beach.