Poročilo za tretjo laboratorijsko vajo predmeta Informacija in Kodi

Gašper Šavle[1]

[1] *Univerza v Ljubljani, Fakulteta za Elektrotehniko*
E-pošta: gaspersavle@yahoo.com

Abstract

This exercise focused on efficient data compression using the LZW (Lempel-Ziv-Welch) algorithm.[1] We had to study the principles of LZW, implement a decoding function, and validate its correctness by encoding and decoding a text file, and comparing MD5 hashes of the original and decoded files. Additionally, we have evaluated the compression efficiency of LZW by comparing the sizes of original and compressed files across different file types and analyzed the performance on partially encoded text files. The exercise provides hands-on experience with data compression techniques and their practical applications.

1 Uvod

Vaja raziskuje gospodarno kodiranje binarnih računalniških datotek z uporabo algoritma LZW (Lempel-Ziv-Welch). Algoritmi za kompresijo so ključnega pomena za učinkovito shranjevanje in prenos podatkov, saj omogočajo zmanjšanje velikosti datotek brez izgube informacij (pri brezizgubni kompresiji).

Namen vaje je seznanitev z osnovami delovanja algoritma LZW, ki temelji na slovarju za kodiranje ponavljajočih se vzorcev v podatkih. V okviru vaje sem implementiral funkcijo za dekodiranje LZW kodiranih datotek, ki je bila preizkušena na vključeni tekstovni datoteki. Preveril sem skladnost izvornih in dekodiranih datotek z izračunom MD5 izvlečkov, kar je omogočilo oceno pravilnosti implementacije.

Poleg tega sem analiziral uspešnost algoritma LZW s primerjavo velikosti izvornih in kompresiranih datotek, pri čemer je bilo vrednoteno gospodarno kodiranje na različnih vrstah datotek. Na ta način smo pridobili vpogled v delovanje kompresijskih algoritmov in njihove

praktične omejitve ter prednosti.

2 Metodologija

2.1 Dekodirnik

Prvi del laboratorijske naloge je bil implementacija dekodirnika za LZW algoritem, pri čemer sem uporabil priloženo predlogo, ki vsebuje osnovne metode za kodiranje. Po preučitvi delovanja algoritma sem ga prilagodil za dekodiranje s pomočjo lekcije na: [2]

2.2 Implementacija MD5 preverjanja

Za preverjanje pravilnosti izvedbe dekodirnika sem z uporabo knjiznice `hashlib` implementiral funkcijo za izračun MD5 izvlečka, s katero sem primerjal izvorno in dekodirano datoteko. Če se izvlečka ujemata, je dekodiranje uspešno in zagotavlja nespremenjenost podatkov, sicer pa dekodiranje ni pravilno.

2.3 Izračun dolžine kompresiranega sporočila

Drugi del vaje vključuje izračun velikosti kompresiranega sporočila na podlagi dolžine kodnega izhoda algoritma. S tem sem ocenil uspešnost algoritma LZW pri kompresiji različnih vrst datotek, kot so nekompresirane (npr. WAV), brezizgubno kompresirane (npr. FLAC) in izgubno kompresirane (npr. MP3). Za vsako datoteko sem izračunal razmerje med izvorno in kompresirano velikostjo, kar omogoča vpogled v učinkovitost algoritma pri različnih tipih podatkov.

2.4 Vrednotenje razmerja med dolžino sporočila in uspešnostjo kompresije

Zadnji del vaje je bil prikazati, kako dolžina podatkov vpliva na kompresijsko razmerje. Na besedilni datoteki sem izvedel kodiranje za različne dolžine izvornih podatkov in za vsak del izračunal kompresijsko razmerje. Rezultate teh analiz sem predstavil tudi vizualno, z grafom, ki prikazuje kompresijsko razmerje glede na velikost izvornih podatkov, z označenimi različnimi dolžinami delov besedilne datoteke.

3 Rezultati

V tem poglavju so predstavljeni rezultati izvedbe laboratorijske vaje, ki vključujejo testiranje algoritma LZW na različnih vrstah datotek, preverjanje integritete podatkov s pomočjo MD5 izvlečkov ter analizo učinkovitosti kompresije. Prikazane so tudi razlike v uspešnosti algoritma glede na dolžino in vrsto podatkov, podprte z numeričnimi izračuni in grafičnimi vizualizacijami. Namen poglavja je ovrednotiti delovanje algoritma LZW in podati vpogled v njegove prednosti ter omejitve.

3.1 Preizkus gospodarnosti kodiranja na različnih vrstah datotek

Rezultati kodiranja LZW algoritma za različne vrste zvočnih datotek odražajo naravo podatkov in njihove začetne stopnje stisnjenosti. Kompresija je bila najuspesnejsa na nekompresirani datoteki tipa WAV, ta tip datoteke je nekompresiran, zato vsebuje veliko ponavljajočih se vzorcev v podatkih. LZW algoritem je zelo učinkovit pri stiskanju takšnih podatkov, saj prepoznava in stiska ponovitve. Zaradi tega je razmerje med izvorno in kompresirano velikostjo visje. [3]

Tabela 1: LZW kodiranje zvočne datoteke tipa WAV

WAV	
Kompresija	Velikost [bajti]
Nekompresirana	145448746
Kompresirana	53372167
Kompresijsko razmerje:	
2.725179698999293	

FLAC je že brezizgubno kompresiran format, ki optimizira podatke in odstrani redundanco, ne da bi izgubili kakovost zvoka. Čeprav LZW še vedno stisne podatke, so možnosti za dodatno kompresijo manjše kot pri nekompresiranih datotekah, saj je večina ponavljajočih se vzorcev že odstranjena. [4]

Tabela 2: LZW kodiranje zvočne datoteke tipa FLAC

FLAC	
Kompresija	Velikost [bajti]
Nekompresirana	97617170
Kompresirana	39985245
Kompresijsko razmerje:	
2.4413297955283255	

MP3 format je izgubno kompresiran in že močno optimiziran za zmanjšanje velikosti datoteke, pri čemer se odstrani del podatkov, ki niso kritični za zaznavanje zvoka. LZW algoritem lahko stisne le preostale strukture v podatkih, vendar je zaradi že zmanjšane redundance razmerje med izvirno in kompresirano velikostjo najmanjše med vsemi tremi primeri. [5]

Tabela 3: LZW kodiranje zvočne datoteke tipa mp3

mp3	
Kompresija	Velikost [bajti]
Nekompresirana	30113070
Kompresirana	13051216
Kompresijsko razmerje:	
2.307299948142763	

3.2 Preizkus gospodarnosti kodiranja pri delnem kodiranju nekompresirane datoteke pri različnih dolžinah vhodne datoteke

Navedeni rezultati se osredotočajo na učinkovitost LZW algoritma pri različnih dolžinah iste besedilne datoteke. Pri večjih deležih izvorne datoteke (npr. 1/1 ali 1/2) algoritem lahko izkoristi več ponavljajočih se vzorcev, kar vodi do večjega razmerja med izvorno in kompresirano velikostjo. Na primer, pri dolžini 1/1 je razmerje približno 5.8:1, medtem ko je pri dolžini 1/10 dolzine izvorne datoteke, razmerje približno 4.2:1. Pri manjših deležih je manj ponovitev, kar pomeni, da slovar LZW vsebuje manj informacij, zaradi česar je dodatna kompresija manj učinkovita. Večji kot je delež vhodnih podatkov, večji slovar lahko algoritem ustvari, kar izboljša kompresijsko razmerje. Pri manjših delih datoteke slovar nima dovolj informacij, da bi dosegel podobno raven stiskanja.

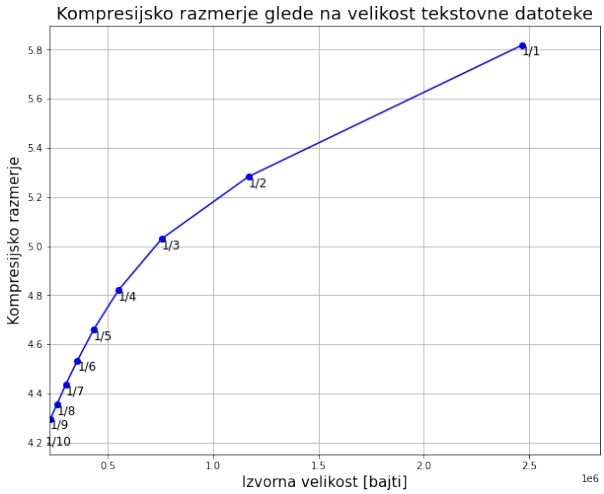
V praksi nam rezultati kazejo, da LZW algoritem bolje deluje pri večjih količinah podatkov, saj lahko prepozna in izkoristi večje število vzorcev. Pri manjših datotekah so njegovi učinki manj izraziti, vendar še vedno učinkoviti.

Tabela 4: LZW kodiranje tekstovne datoteke pri različnih dolžinah

Tekstovna datoteka		
Dolžina	Kompresija	Velikost [bajti]
1/1	Nekompresirana	3011300
	Kompresirana	13051216
1/2	Nekompresirana	1168600
	Kompresirana	221198
1/3	Nekompresirana	757348
	Kompresirana	150571
1/4	Nekompresirana	551991
	Kompresirana	114505
1/5	Nekompresirana	435111
	Kompresirana	93347
1/6	Nekompresirana	357189
	Kompresirana	78785
1/7	Nekompresirana	301530
	Kompresirana	67985
1/8	Nekompresirana	259905
	Kompresirana	59687
1/9	Nekompresirana	229241
	Kompresirana	53356
1/10	Nekompresirana	204712
	Kompresirana	48386

Tabela 5: Kompresijska razmerja pri razlicnih dlzinah tekstovne datoteke

Kompresijska razmerja	
Dolžina	Razmerje
1/1	5.818746668050556
1/2	5.2830495754934494
1/3	5.029839743376879
1/4	4.820671586393607
1/5	4.661221035491232
1/6	4.533718347401155
1/7	4.435243068323895
1/8	4.354465796572118
1/9	4.296442761826224
1/10	4.230810565039474



Slika 1: Grafični prikaz uspešnosti kompresije pri različnih dolžinah vhodne datoteke

Literatura

[1] Wikipedia contributors, "Lempel–ziv–welch — Wikipedia, the free encyclopedia," 2024. [Online; accessed 1-December-2024].

[2] GeeksForGeeks, "Lzw (lempel–ziv–welch) compression technique." [Online; accessed 1-December-2024].

[3] Wikipedia contributors, "Wav — Wikipedia, the free encyclopedia," 2024. [Online; accessed 1-December-2024].

[4] Wikipedia contributors, "Flac — Wikipedia, the free encyclopedia," 2024. [Online; accessed 1-December-2024].

[5] Wikipedia contributors, "Mp3 — Wikipedia, the free encyclopedia," 2024. [Online; accessed 1-December-2024].