

Übung zur Vorlesung

Digitale Medien

Vorlesung: Heinrich Hußmann

Übung: Renate Häuslschmid, Hanna Schneider

Ludwig-Maximilians-Universität München

Wintersemester 2015/16

Wiederholung: LZW-Komprimierung

Idee:

Nicht einzelne Zeichen werden günstig kodiert, sondern ganze Zeichenketten (*Wörterbuch-Kompression*).

Nachteile:

Mehr Symbole nötig (jeder Eintrag ins Wörterbuch ist ein Symbol) und Empfänger muss Wörterbuch kennen, um die Nachricht dekodieren zu können.

Aber: Muss das Wörterbuch wirklich mit übertragen werden?

Bei der LZW-Komprimierung wird das Wörterbuch *während* der Dekodierung aufgebaut!

LZW-Dekomprimierung

Algorithmus zur Dekomprimierung:

SeqChar p := <>;

int k = NächsteEingabezahl;

Schreibe Zeichenreihe mit Tabellenindex k auf Ausgabe;

int old = k;

Wiederhole solange Eingabe nicht leer:

 k = NächsteEingabezahl;

SeqChar akt = Zeichenreihe mit Tabellenindex k;

 p = Zeichenreihe mit Tabellenindex old (letztes Teilwort);

Falls Index k in Tabelle enthalten

dann Char q = erstes Zeichen von akt;

 Schreibe Zeichenreihe akt auf Ausgabe;

sonst Char q = erstes Zeichen von p;

 Schreibe Zeichenreihe p & <q> auf Ausgabe;

Ende Fallunterscheidung;

 Trage p & <q> in Tabelle ein

 (und erzeuge neuen Index dafür);

 old = k;

Ende Wiederholung;

a	97	h	104	o	111	v	118
b	98	i	105	p	112	w	119
c	99	j	106	q	113	x	120
d	100	k	107	r	114	y	121
e	101	l	108	s	115	z	122
f	102	m	109	t	116		
g	103	n	110	u	117		

LZW-Dekomprimierung

Lesen (k)	Ausgabe (q ist unterstrichen)	Puffer füllen (p)	Codetabelle (p & <q>)	Merken (old)
108	l			108
97	<u>a</u>	l	<la>, 256	97
98	<u>b</u>	a	<ab>, 257	98
97	<u>a</u>	b	<ba>, 258	97
256	<u>l</u> a	a	<al>, 259	256

SeqChar p := <>;

int k = NächsteEingabezahl;

Schreibe Zeichenreihe mit Tabellenindex k auf Ausgabe;

int old = k;

Wiederhole solange Eingabe nicht leer:

 k = NächsteEingabezahl;

SeqChar akt = Zeichenreihe mit Tabellenindex k;

 p = Zeichenreihe mit Tabellenindex old (letztes Teilwort);

Falls Index k in Tabelle enthalten

dann Char q = erstes Zeichen von akt;

 Schreibe Zeichenreihe akt auf Ausgabe;

sonst Char q = erstes Zeichen von p;

 Schreibe Zeichenreihe p & <q> auf Ausgabe;

Ende Fallunterscheidung;

 Trage p & <q> in Tabelle ein

 (und erzeuge neuen Index dafür);

WS 2015/16 old = k;

Ende Wiederholung;

108 – 97 – 98 – 97 - 256

LZW-Dekomprimierung

98 – 97 – 108 – 108 – 97 – 256 – 258 – 257 - 97

Lesen (k)	Ausgabe (q ist unterstrichen)	Puffer füllen (p)	Codetabelle (p & <q>)	Merken (old)

SeqChar p := <>;

int k = NächsteEingabezahl;

Schreibe Zeichenreihe mit Tabellenindex k auf Ausgabe;

int old = k;

Wiederhole solange Eingabe nicht leer:

 k = NächsteEingabezahl;

SeqChar akt = Zeichenreihe mit Tabellenindex k;

 p = Zeichenreihe mit Tabellenindex old (letztes Teilwort);

Falls Index k in Tabelle enthalten

dann Char q = erstes Zeichen von akt;

 Schreibe Zeichenreihe akt auf Ausgabe;

sonst Char q = erstes Zeichen von p;

 Schreibe Zeichenreihe p & <q> auf Ausgabe;

Ende Fallunterscheidung;

 Trage p & <q> in Tabelle ein

 (und erzeuge neuen Index dafür);

 old = k;

Ende Wiederholung;

LZW-Dekomprimierung

98 – 97 – 108 – 108 – 97 – 256 – 258 – 257 - 97

Lesen (k)	Ausgabe (q ist unterstrichen)	Puffer füllen (p)	Codetabelle (p & <q>)	Merken (old)
98	b			98
97	<u>a</u>	b	<ba>, 256	97
108	<u>l</u>	a	<al>, 257	108
108	<u>l</u>	l	<ll>, 258	108
97	<u>a</u>	l	<la>, 259	97
256	<u>ba</u>	a	<ab>, 260	256
258	<u>ll</u>	ba	<bal>, 261	258
257	<u>al</u>	ll	<lla>, 262	257
97	<u>a</u>	al	<ala>, 263	97

Ergebnis: ballaballala

LZW-Dekomprimierung

97 – 98 – 256 – 258

Lesen (k)	Ausgabe (q ist unterstrichen)	Puffer füllen (p)	Codetabelle (p & <q>)	Merken (old)
97	a			97
98	<u>b</u>	a	<ab>, 256	98
256	<u>ab</u>	b	<ba>, 257	256
258	aba	<u>ab</u>	<aba>, 258	258

Ergebnis: abababa

Aufgabe 1

Gegeben sei folgende Nachricht: **wawawawobawaw**

- a) Codieren Sie die Nachricht mittels LZW-Codierung. Gehen Sie dabei entsprechend dem in der Vorlesung behandelten Algorithmus vor. Verwenden Sie zur Darstellung der Zwischenschritte des Algorithmus eine Tabelle mit den Spalten: „Lesen“, „Codetabelle schreiben“, „Ausgabe“ und „Puffer füllen“. Das Wörterbuch wird mit der in Blatt 3 angegebenen Codetabelle initialisiert.

- a) Decodieren Sie die Nachricht. Verwenden Sie zur Darstellung der Zwischenschritte des Algorithmus eine Tabelle mit den Spalten: „Lesen“, „Ausgabe“, „Puffer füllen“, „Codetabelle schreiben“ und „Merken“.

Lösung zu Aufgabe 1a)

Beispiel: **wawawobawaw**

Lesen	Codetabelle schreiben	Ausgabe	Puffer füllen
			<w>
a	<wa>, 256	119 (w)	<a>
w	<aw>, 257	97 (a)	<w>
a			<wa>
w	<waw>, 258	256 (wa)	<w>
a			<wa>
w			<waw>
o	<wawo>, 259	258 (waw)	<o>
b	<ob>, 260	111 (o)	
a	<ba>, 261	98 (b)	<a>
w			<aw>
a	<awa>, 262	257 (aw)	<a>
w			<aw>
		257 (aw)	<>

Lösung zu Aufgabe 1a)

119

97

256

258

111

98

257

257

Lesen (k)	Ausgabe (q)	Puffer füllen (p)	Codetabelle schreiben (p & <q>)	Merken (old)
119	w			119
97	<u>a</u>	w	<wa>, 256	97
256	<u>wa</u>	a	<aw>, 257	256
258	waw	<u>wa</u>	<waw>, 258	258
111	<u>o</u>	waw	<wawo>, 259	111
98	<u>b</u>	o	<ob>, 260	98
257	<u>aw</u>	b	<ba>, 261	257
257	<u>aw</u>	aw	<awa>, 262	257
EOF				

SPEICHERPLATZBEDARF

Speicherplatzbedarf

Dateigröße von Bildern:

Breite x Höhe x Farbtiefe

Dateigröße von Videos:

Breite x Höhe x Farbtiefe x Framerate x Zeit

Dateigröße von Audio:

Abtastrate x Auflösung x Zeit

Speicherplatzbedarf eines Bildes

Berechnen Sie den Speicherplatzbedarf eines unkomprimierten 1600x1200 Pixel großen Farbfotos mit einer Farbtiefe von 24 Bit in Bit und Kilobyte.

$$\begin{aligned} &1600 \times 1200 \times 24 \\ &= \mathbf{46\ 080\ 000\ bit} / 8 / 1000 \\ &= \mathbf{5\ 760\ Kilobyte} \end{aligned}$$

Speicherplatzbedarf eines Videos

Berechnen Sie die Datenrate in MBit/s für ein unkomprimiertes Video mit der Auflösung 320x240 Pixel, 25 Bilder/s und 24 Bit Farbtiefe.

$$320 \times 240 \times 25 \times 24$$

$$= 46\,080\,000 \text{ bit/s} / 1000$$

$$= 46\,080 \text{ KBit/s} / 1000$$

$$= \mathbf{46,08 \text{ MBit/s}}$$

Speicherplatzbedarf eines Liedes

Berechnen Sie die Größe der digitalen Rohdaten in Byte für ein 2 Minuten langes Musikstück mit einer Monoaudiospur mit 22kHz Abtastrate und 16Bit Auflösung.

$$\begin{aligned} &22\,000 \times 16 \times 2 \times 60 \\ &= 42240000 \text{ Bit} / 8 \\ &= \mathbf{5\,280\,000 \text{ Byte}} \end{aligned}$$

Aufgabe 2

- a) Berechnen Sie den Speicherplatzbedarf eines unkomprimierten 1024x768 Pixel großen Farbfotos mit einer Farbtiefe von 24 Bit.
- b) Berechnen Sie die Datenrate in MBit/s für ein unkomprimiertes Video mit der Auflösung 320x200 Pixel, 30 Bilder/s und 24 Bit Farbtiefe.
- c) Berechnen Sie die Größe der digitalen Rohdaten in Byte für ein 3 Minuten langes Video mit einer Auflösung von 320x240 Pixel, 15 Bilder pro Sekunde (fps), einer Farbtiefe von 24bit, Monoaudiospur mit 22kHz Abtastrate und 16Bit Auflösung.

Lösung zu Aufgabe 2a)

$$1024 \times 768 \text{ px} = 786432 \text{ px}$$

$$786432 \text{ px} * 24 \text{ bit/px} = 18874368 \text{ bit}$$

$$18874368 \text{ bit} / 8 = 2359296 \text{ Byte}$$

$$2359296 \text{ Byte} \approx 2359 \text{ Kbyte} [/ 1000]$$

$$2359 \text{ KByte} \approx \mathbf{2,36 \text{ MB}} \quad [/ 1000]$$

vgl. JPEG ca. 50 KB

Lösung zu Aufgabe 2b)

$$320 \times 200 \text{ px} = 64000 \text{ px}$$

$$64000 \text{ px} * 24 \text{ bit/px} = 1536000 \text{ bit (1 Bild)}$$

$$1536000 \text{ bit} * 30 \text{ fps} = 46080000 \text{ bps}$$

$$46080000 \text{ bps} = 46080 \text{ Kbps} \quad [/ 1000]$$

$$46080 \text{ Kbps} = 46,08 \text{ Mbps} \quad [/ 1000]$$

$$46,08 \text{ Mbps} = \mathbf{5,76 \text{ MBps}} \quad [/ 8]$$

vgl. normales Fernsehen 768x576 Pixel

Lösung zu Aufgabe 2c)

1 Bild = $320 \times 240 \text{ px} * 24 \text{ bit/px} = 1843200 \text{ bit}$

15 fps * 3 * 60s = 2700 Frames

Bildmaterial = $1843200 \text{ b} * 2700 = 622,08 \text{ MB}$

Audio: $180 \text{ s} * 22000 \text{ 1/s} * 16 \text{ b} = 7,92 \text{ MB}$

$622,08 \text{ MB} + 7,92 \text{ MB} = \text{ca. } \mathbf{630 \text{ MB}}$

630 MB für 3 Minuten! Ohne Kompression nicht möglich...

Übungsblatt 4

- Übungsblatt 4:
<https://www.medien.ifi.lmu.de/lehre/ws1516/dm/>
- Abgabe bis Freitag den 20.11.2015, 09:00 Uhr
morgens in [UniWorX](#)