

Übung zur Vorlesung
Digitale Medien

Hanna Schneider
Ludwig-Maximilians-Universität München
Wintersemester 2014/2015

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)
- 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 (1)

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 (2)

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);

 old = k;

Ende Wiederholung;

108 – 97 – 98 – 97 - 256

LZW-Dekomprimierung (3)

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 (4)

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 (5)

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

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

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

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

Speicherplatzbedarf

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

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

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