

LZW-Kompression

Wörterbuch-Kompressionen

- Grundidee:
 - Suche nach dem "Vokabular" des Dokuments, d.h. nach sich wiederholenden Teilsequenzen
 - Erstelle Tabelle: Index --> Teilsequenz ("Wort")
 - Tabelle wird dynamisch während der Kodierung aufgebaut
 - Codiere Original als Folge von Indizes
- Praktische Algorithmen:
 - Abraham Lempel, Jacob Ziv (Israel), Ende 70er-Jahre
 - » LZ77- und LZ78-Algorithmen
 - Verbessert 1984 von A. Welch = "LZW"-Algorithmus (Lempel/Ziv/Welch)
 - Basis vieler semantikunabhängiger Kompressionsverfahren (z.B. UNIX "compress", Zip, gzip, V42.bis)
 - Verwendet in vielen Multimedia-Datenformaten (z.B. GIF)

Aufgabe 1: Codierung nach Lempel-Ziv-Welch

Gegeben sei die Codetabelle des ISO8859-Zeichensatzes (siehe S. 3).

Ab Index 256 können neue Einträge erfolgen.

Weiterhin gegeben sei folgende Nachricht:

tohouwabohou

- 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 oben angegebenen Codetabelle initialisiert.
- b) 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“.

Anhang: ISO 8859-1 Codetabelle

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80			,	f	„	…	†	‡	^	%	Š	<	€			
90		‘	’	“	”	•	—	—	~	™	š	>	œ			ÿ
A0		ı	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯
B0	°	±	²	³	´	µ	¶	·	,	ı	°	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

b a n a n e n a n b a u

ba	an	na	ane	en	nan	nb	bau
----	----	----	-----	----	-----	----	-----

SeqChar $p = \langle \text{NächstesEingabezeichen} \rangle$;

Char $k = \text{NächstesEingabezeichen}$;

Wiederhole:

Falls $p \ \& \ \langle k \rangle$ in Tabelle enthalten

dann $p = p \ \& \ \langle k \rangle$

sonst trage $p \ \& \ \langle k \rangle$ neu in Tabelle ein

(und erzeuge neuen Index dafür);

Schreibe Tabellenindex von p auf Ausgabe;

$p = \langle k \rangle$;

Ende Fallunterscheidung;

$k = \text{NächstesEingabezeichen}$;

solange bis Eingabeende

Schreibe Tabellenindex von p auf Ausgabe:

LZW-Codierung (3)

- Beispieltext: "bananenbau"
- Ablauf:

Lesen (k)	Codetabelle schreiben (p & <k>)	Ausgabe	Puffer füllen (p)
b			
a	(<ba>, 256)	98	<a>
n	(<an>, 257)	97	<n>
a	(<na>, 258)	110	<a>
n			<an>
e	(<ane>, 259)	257	<e>
n	(<en>, 260)	101	<n>
a			<na>
n	(<nan>, 261)	258	<n>
b	(<nb>, 262)	110	
a			<ba>
u	(<bau>, 263)	256	<u>
EOF		117	

Aufgabe 1: Codierung nach Lempel-Ziv-Welch

Gegeben sei die Codetabelle des ISO8859-Zeichensatzes (siehe S. 3).

Ab Index 256 können neue Einträge erfolgen.

Weiterhin gegeben sei folgende Nachricht:

tohouwabohou

- 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 oben angegebenen Codetabelle initialisiert.
- b) 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“.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80			,	f	„	…	†	‡	^	%	Š	<	Œ			
90		‘	’	“	”	•	—	~	™	š	>	œ				ÿ
A0		ı	€	£	×	¥		§	”	©	ª	«	–	-	®	–
B0	°	±	²	³	´	µ	¶	·	,	¹	º	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

tohouwa bohou

Lesen (k)	Code tabelle schreiben (p & <k>)	Ausgabe	Puffer füllen (p)
t			<t>
o	<to>, 256	116	<o>
h	<oh>, 257	111	<h>
o	<ho>, 258	104	<o>
u	<ou>, 259	117	<u>
w	<ow>, 260	117	<w>
a	<wa>, 261	119	<a>
b	<ab>, 262	97	
o	<bo>, 263	98	<o>
h			<oh>
o	<oho>, 264	257	<o>
u			<ou>
		259	

SeqChar p = < NächstesEingabezeichen >;

Char k = NächstesEingabezeichen;

Wiederhole:

Falls p & <k> in Tabelle enthalten

dann p = p & <k>

sonst trage p & <k> neu in Tabelle ein

(und erzeuge neuen Index dafür);

Schreibe Tabellenindex von p auf Ausgabe;

p = <k>;

Ende Fallunterscheidung;

k = NächstesEingabezeichen;

solange bis Eingabeende

Schreibe Tabellenindex von p auf Ausgabe:

116 111 104 117 117 119 97 98 257 259

Aufgabe 1: Codierung nach Lempel-Ziv-Welch

Gegeben sei die Codetabelle des ISO8859-Zeichensatzes (siehe S. 3).

Ab Index 256 können neue Einträge erfolgen.

Weiterhin gegeben sei folgende Nachricht:

tohouwabohou

- 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 oben angegebenen Codetabelle initialisiert.
- b) 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“.

LZW-Decodierung (2)

- Prinzipieller Algorithmus:

SeqChar $p := \langle \rangle$;

int $k = \text{NächsteEingabezahl}$;

Schreibe Zeichenreihe mit Tabellenindex k auf Ausgabe;

int $old = k$;

Wiederhole solange Eingabe nicht leer:

$k = \text{NächsteEingabezahl}$;

SeqChar $akt = \text{Zeichenreihe mit Tabellenindex } k$;

Schreibe Zeichenreihe akt auf Ausgabe;

$p = \text{Zeichenreihe mit Tabellenindex } old \text{ (letztes Teilwort)}$;

Char $q = \text{erstes Zeichen von } akt$;

Trage $p \ \& \ \langle q \rangle$ in Tabelle ein
(und erzeuge neuen Index dafür);

$old = k$;

Ende Wiederholung;

Anhang: ISO 8859-1 Codetabelle

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80			,	f	„	…	†	‡	^	%	Š	<	Œ			
90		‘	’	“	”	•	—	~	™	š	>	œ				ÿ
A0		ı	€	£	¤	¥	¦	§	¨	©	ª	«	¬	®	¯	
B0	°	±	²	³	´	µ	¶	·	,	¹	º	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

116 111 104 117
117 119 97 98 257 259

Lesen (li)	Ausgabe (y unterstr.)	Puffer füllen (pl)	Codetabelle p. 6, 297	Merken (old)
116	+			116
111	<u>o</u>	+	<fo?, 256	111
104	<u>h</u>	o	<oh?, 257	104
111	<u>h</u>	h	<ho?, 258	111
117	<u>o</u>	o	<oo?, 259	117
119	<u>u</u>	u	<ou?, 260	119
97	<u>w</u>	w	<wa?, 261	97
98	<u>a</u>	a	<ab?, 262	98
257	<u>b</u>	b	<bo?, 263	257
259	<u>o</u>	oh	<oho?, 264	259
EOS				

to howwa bo how



Aufgabe 2: LZW-Codierung mit Ausnahmebehandlung

Gegeben sei folgende Nachricht:

rokokokostuem

Die Vorbelegung des Wörterbuchs sei entsprechend der ISO8859-Codetabelle (siehe S. 3)

Ab Index 256 können neue Einträge erfolgen.

- a) Codieren Sie die Nachricht mittels LZW analog zu Aufgabe 1.
- b) Decodieren Sie die Nachricht.

Wichtiger Hinweis: Der auf S.39 im Taschenbuch Multimedia¹ vorgestellte Algorithmus zur Dekompression weist an einer Stelle einen Fehler auf. Die Vorlesung enthält die richtige Fassung.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80			,	f	„	…	†	‡	^	%	Š	<	Œ			
90		‘	’	“	”	•	—	~	™	š	>	œ			ÿ	
A0		ı	€	£	×	¥		§	”	©	ª	«	–	-	®	–
B0	°	±	²	³	´	µ	¶	·	,	¹	º	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	à	á	â	ã	ä	å										
F0	ö	ñ	ò	ó	ô	õ										

SeqChar p = < NächstesEingabezeichen >;

Char k = NächstesEingabezeichen;

Wiederhole:

Falls p & <k> in Tabelle enthalten

dann p = p & <k>

sonst trage p & <k> neu in Tabelle ein

(und erzeuge neuen Index dafür);

Schreibe Tabellenindex von p auf Ausgabe;

p = <k>;

Ende Fallunterscheidung;

k = NächstesEingabezeichen;

solange bis Eingabeende

Schreibe Tabellenindex von p auf Ausgabe;

	Lesen(k)	Codetabelle schreiben (p & k)	Ausgabe	Puffer füllen (p)
r				<r>
o		<ro>, 256	114 114	<o>
k		<ok>, 257	111	<k>
o		<ko>, 258	107	<o>
k				<ok>
o		<oko>, 259	257	<o>
k				<ok>
o				<oko>
s		<okos>, 260	259	<s>
t		<st>, 261	115	<t>
u		<tu>, 262	116	<u>
e		<ue>, 263	117	<e>
m		, 264	101	<m>
			109	

114 111 107 257 259 115 116 117 101 109

Aufgabe 2: LZW-Codierung mit Ausnahmebehandlung

Gegeben sei folgende Nachricht:

rokokokostuem

Die Vorbelegung des Wörterbuchs sei entsprechend der ISO8859-Codetabelle (siehe S. 3)

Ab Index 256 können neue Einträge erfolgen.

- a) Codieren Sie die Nachricht mittels LZW analog zu Aufgabe 1.
- b) Decodieren Sie die Nachricht.

Wichtiger Hinweis: Der auf S.39 im Taschenbuch Multimedia¹ vorgestellte Algorithmus zur Dekompression weist an einer Stelle einen Fehler auf. Die Vorlesung enthält die richtige Fassung.

Anhang: ISO 8859-1 Codetabelle

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80			,	f	„	…	†	‡	^	%	Š	<	Œ			
90		‘	’	“	”	•	—	~	™	š	>	œ				ÿ
A0		ı	€	£	¤	¥	¦	§	¨	©	ª	«	¬	®	¯	
B0	°	±	²	³	´	µ	¶	·	,	¹	º	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

114 111 107 257 259
 115 116 117 101 109

Leser (u)	Ausgabe (q) <small>Öhrens.</small>	Puffer füllen (p)	Codefabelle p. b. sq.	Merken (old)
114	<u>r</u>			114
111	<u>o</u>	r	<ro>, 256	111
107	<u>u</u>	o	<ou>, 257	107
257	<u>ok</u>	k	<ko>, 258	257
259				
115				
116				
117				
101				
109				



LZW-Decodierung, vollständige Fassung

SeqChar $p := \langle \rangle$;

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 $p \ \& \ \langle q \rangle$ auf Ausgabe;

Ende Fallunterscheidung;

Trage $p \ \& \ \langle q \rangle$ in Tabelle ein
(und erzeuge neuen Index dafür);

$old = k$;

Ende Wiederholung;

Anhang: ISO 8859-1 Codetabelle

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80			,	f	„	…	†	‡	^	%	Š	<	Œ			
90		‘	’	“	”	•	—	~	™	š	>	œ				ÿ
A0		ı	€	£	¤	¥	¦	§	¨	©	ª	«	¬	®	¯	
B0	°	±	²	³	´	µ	¶	·	,	¹	º	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

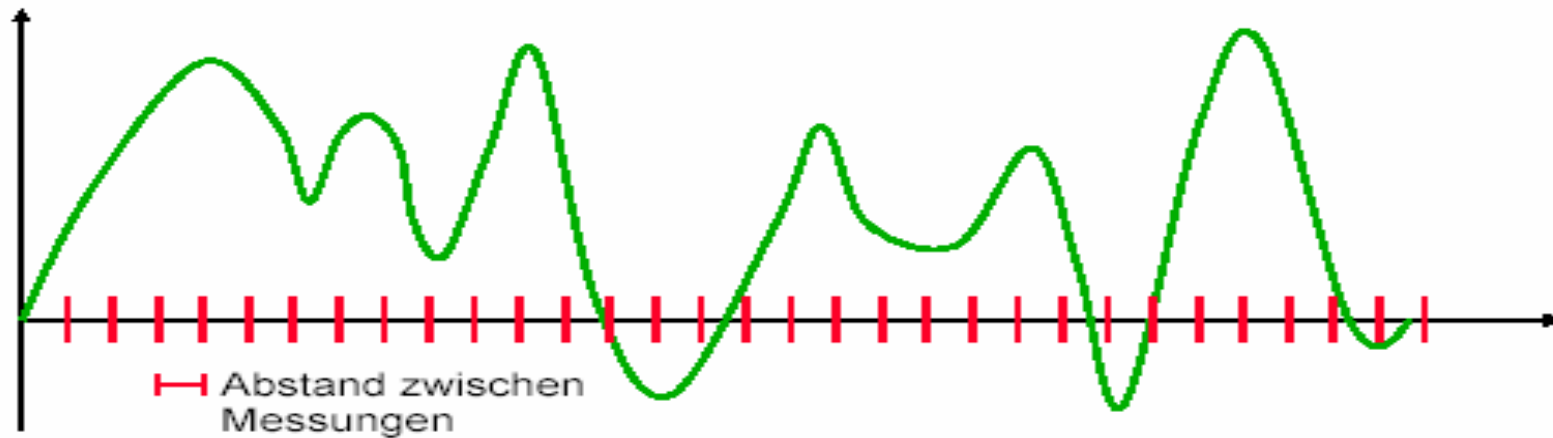
114 111 107 257 259
115 116 117 101 109

Lesen (u)	Ausgabe (q) ^{Orthogr.}	Puffer füllen (p)	Codefabelle p.b. eq?	Merken (old)
114	r			114
111	o	r	<ro>, 256	111
107	u	o	<ou>, 257	107
257	ok	k	<ko>, 258	257
259	oko	ok	<oko>, 259	259
115	s	oko	<okos>, 260	115
116	t	s	<st>, 261	116
117	u	t	<tu>, 262	117
101	e	u	<ue>, 263	101
109	m	e	, 264	109

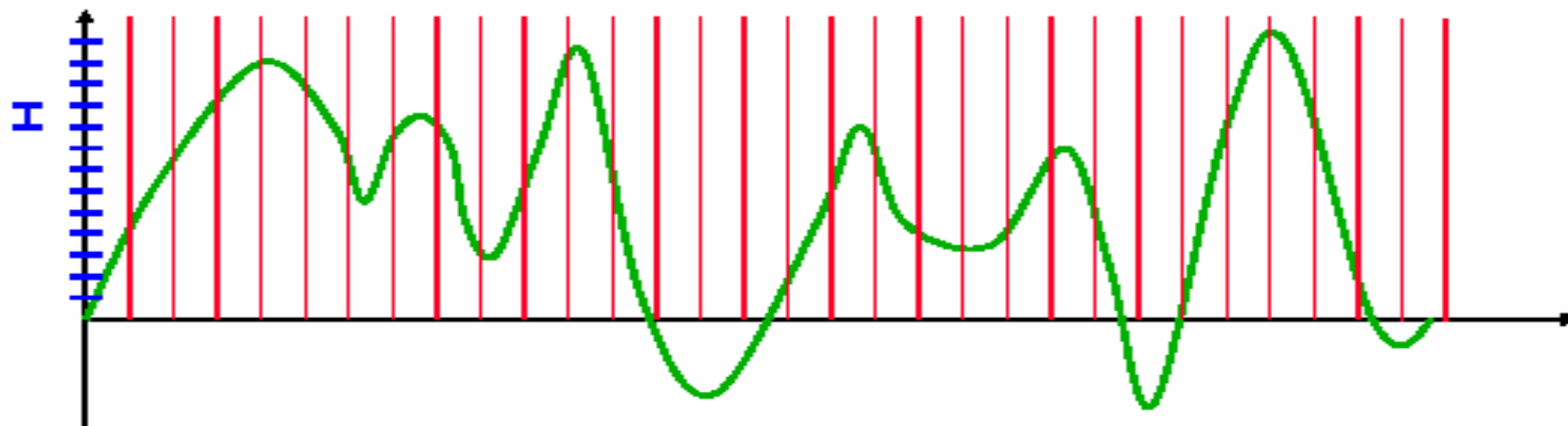
Digitalisierung

Digitalisierung = Diskretisierung+Quantisierung

Diskretisierung

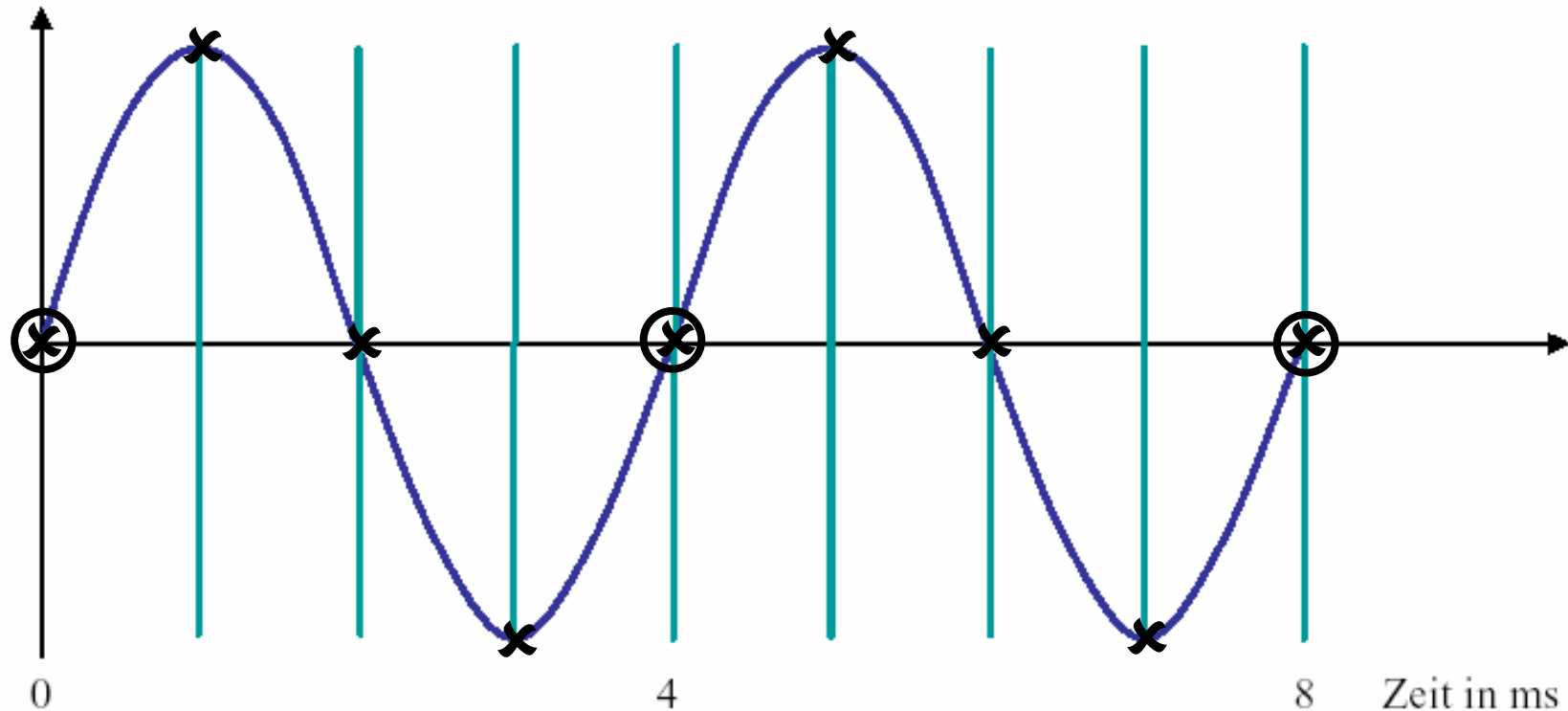


Quantisierung



Aufgabe 3: Digitalisierung

In folgender Abbildung ist ein Signal dargestellt:



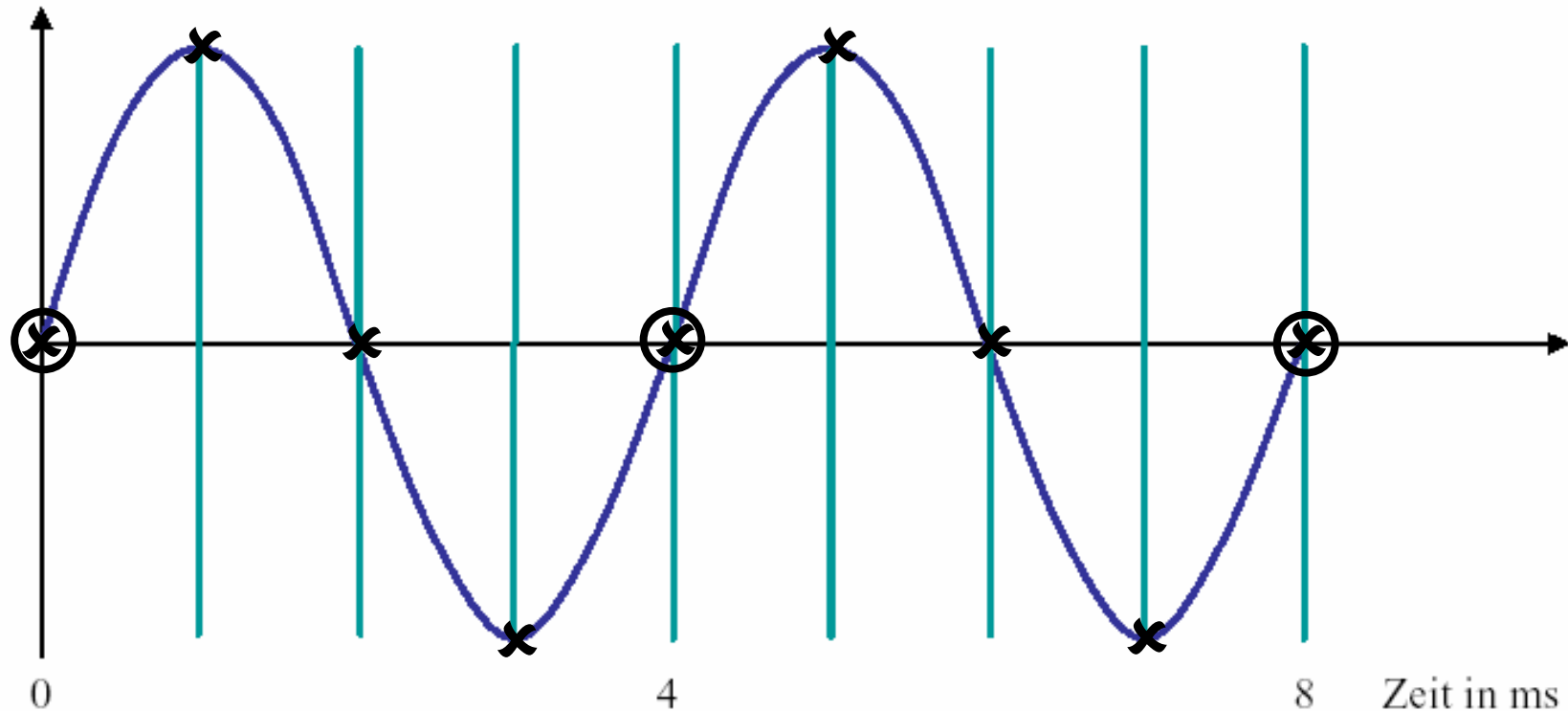
- a) Welche Frequenz hat das Signal? **Lösung: 250 Hz**
- b) Zeichnen Sie in die Abbildung die Abtastung mit 0,25 kHz mittels Kreismarkierungen und die Abtastung mit 1 kHz mittels Kreuzmarkierungen ein.
- c) Welche Abtastrate ist mindestens notwendig, damit das Signal wieder eindeutig rekonstruiert werden kann?

Abtasttheorem

- Nach Harry Nyquist (1928) oft auch Nyquist-Theorem genannt. (Beweis von Claude Shannon)
- Wenn eine Funktion
 - mit höchster vorkommender Frequenz f_g (Bandbegrenzung)
 - mit einer Abtastrate f_s abgetastet wird, so dass
$$f_s > 2 \cdot f_g ,$$
 - dann kann die Funktion eindeutig aus den Abtastwerten rekonstruiert werden.
- Praktisches Beispiel:
 - CD-Abtastrate 44,1 kHz

Aufgabe 3: Digitalisierung

In folgender Abbildung ist ein Signal dargestellt:



- Welche Frequenz hat das Signal? **Lösung: 250 Hz**
- Zeichnen Sie in die Abbildung die Abtastung mit 0,25 kHz mittels Kreismarkierungen und die Abtastung mit 1 kHz mittels Kreuzmarkierungen ein.
- Welche Abtastrate ist mindestens notwendig, damit das Signal wieder eindeutig rekonstruiert werden kann? **Abtastfrequenz $> 2 \cdot f$ (Nyquist-Theorem)**

Aufgabe 4 (optional): LZW-Codierung

Gegeben sei folgende Nachricht:

abrakadabra

Die Vorbelegung des Wörterbuchs sei entsprechend der folgenden Codetabelle:

Zeichen	a	b	d	k	r
Index	0	1	2	3	4

Ab Index 5 können daher neue Einträge erfolgen.

- Beschreiben Sie kurz das Prinzip der Codierung beim LZW-Verfahren in eigenen (vollständigen) Sätzen.
- Codieren Sie die Nachricht mittels LZW analog zu Aufgabe 1.
- Beschreiben Sie ebenso kurz das Prinzip der Dekodierung bei LZW und erläutern Sie, warum das Wörterbuch nicht mit übertragen werden muss.
- Decodieren Sie die Nachricht.

Gegeben sei folgende Nachricht:

abrakadabra

Die Vorbelegung des Wörterbuchs sei entsprechend der folgenden Codetabelle:

Zeichen	a	b	d	k	r
Index	0	1	2	3	4

Lesen(k)	Codetabelle schreiben $n \leq k$	Ausgabe	Puffer füllen (n)
a			<a>
b	<ab>, 5	0	
r	< b r>, 6	1	<r>
a	<ra>, 7	4	<a>
k	<ak>, 8	0	<k>
a	<ka>, 9	3	<a>
d	<ad>, 10	0	<d>
a	<da>, 11	2	<a>
b			<ab>
r	<abr>, 12	5	<r>
a			<ra>
		7	

Aufgabe 4 (optional): LZW-Codierung

Gegeben sei folgende Nachricht:

abrakadabra

Die Vorbelegung des Wörterbuchs sei entsprechend der folgenden Codetabelle:

Zeichen	a	b	d	k	r
Index	0	1	2	3	4

Ab Index 5 können daher neue Einträge erfolgen.

- Beschreiben Sie kurz das Prinzip der Codierung beim LZW-Verfahren in eigenen (vollständigen) Sätzen.
- Codieren Sie die Nachricht mittels LZW analog zu Aufgabe 1.
- Beschreiben Sie ebenso kurz das Prinzip der Dekodierung bei LZW und erläutern Sie, warum das Wörterbuch nicht mit übertragen werden muss.
- Decodieren Sie die Nachricht.

Gegeben sei folgende Nachricht:

abrakadabra

Die Vorbelegung des Wörterbuchs sei entsprechend der folgenden Codetabelle:

Zeichen	a	b	d	k	r
Index	0	1	2	3	4

Lesen (h)	Ausgabe <qr> umkehrsh.	Puffer füllen (p)	Codetabelle p & <qr>	Merken
0	<u>a</u>			0
1	<u>b</u>	a	<ab>, 5	1
4	<u>r</u>	b	 , 6	4
0	<u>a</u>	r	<ra>, 7	0
3	<u>k</u>	a	<ak>, 8	3
0	<u>a</u>	k	<ka>, 9	0
2	a <u>d</u>	a	<ad>, 10	2
5	<u>ab</u>	d	<da>, 11	5
7	<u>ra</u>	ab	 , 12	7