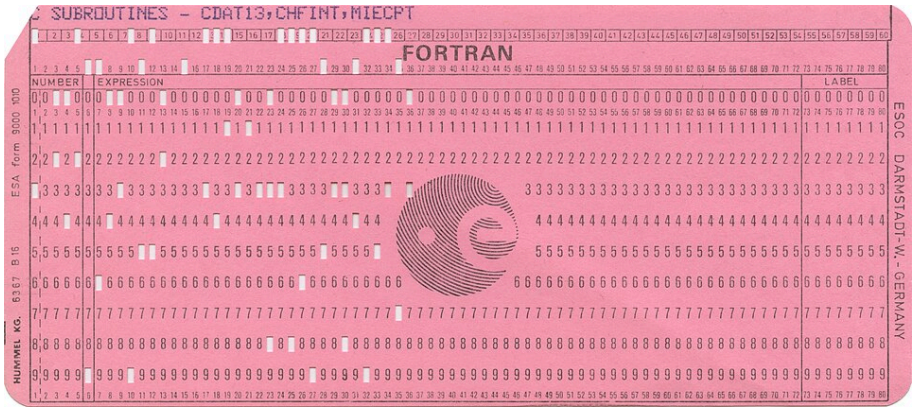

Einführung in Text-Codierung

Benjamin Schmid

Um Daten übermitteln und speichern zu können, müssen diese in ein maschinenlesbares Format gebracht werden. Hierfür gibt es verschiedenste historische und aktuelle Verfahren, die genutzt werden.

Inhaltsverzeichnis

Kapitel 1: Historische Verfahren

Der Wunsch, Informationen schnell über weite Distanzen übertragen zu können, gibt es schon lange. Zu Beginn wurden hierfür primär akustische oder optische Systeme verwendet. Beispiele für akustische Systeme sind z.B. Kirchglocken, die den Bauern auf dem Feld die Uhrzeit mitteilen konnten. Optische Systeme sind z.B. Rauchzeichen, Signalfeuer, oder sogenannte Semaphoren, die mittels Zeigern Informationen übertragen konnten (Abbildung 1). Von diesen gab es ganze Netzwerke, um Informationen über weite Distanzen übertragen zu können. Später kam dann die Möglichkeit hinzu, Signale mittels elektrischer Signale übertragen zu können.

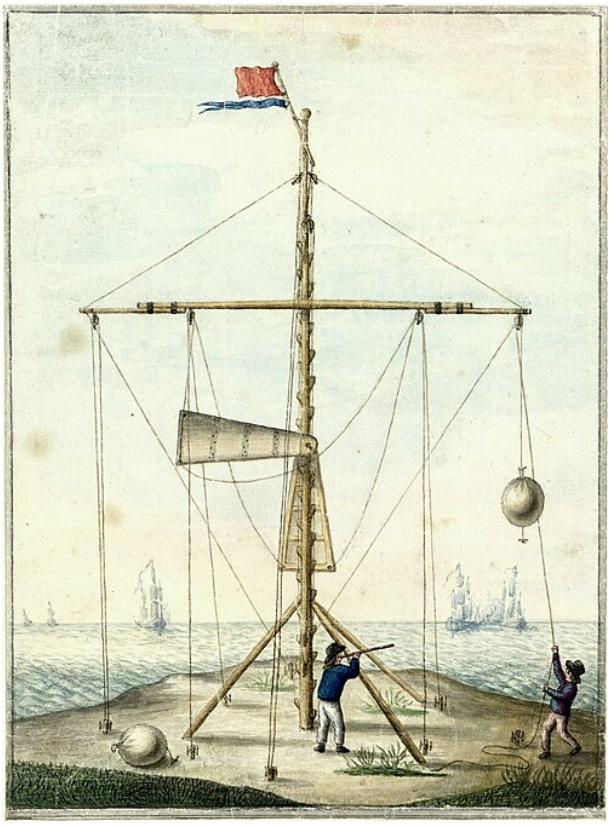

Abbildung 1: Eine Semaphor an der Küste.

Ein bekanntes Beispiel für die Datenübertragung mit elektrischen Signalen ist der Morsetelegraf. Dabei verwendet der Sender eine Morsetaste (Abbildung 2), um kurze und lange Signale zu übertragen. Beim Empfänger werden diese Signale dann mit einem Lautsprecher hörbar ausgegeben und können decodiert werden. Das System, wie Buchstaben mit kurzen und langen Signalen dargestellt werden, nennt man Morse-Code.

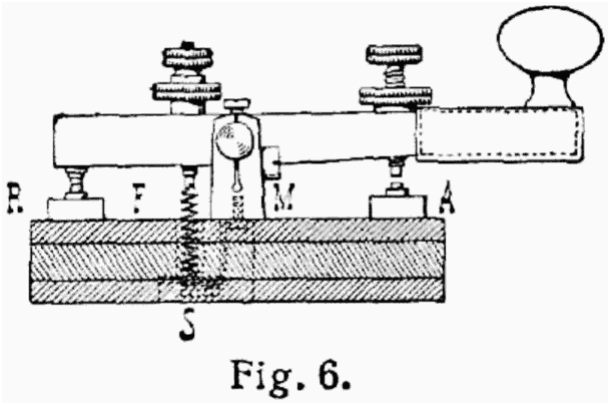

Abbildung 2: Eine Morsetaste.

Bei all diesen Systemen müssen sich Sender und Empfänger einig sein, welche Signale welche Bedeutung haben. Dies nennt man dann eine *Codierung*. Wird ein Text in die Signale für die Übermittlung umgewandelt, nennt man dies *codieren*. Das Gegenteil, die Umwandlung der Signale in Text, nennt man *decodieren*. In den nächsten Kapiteln

werden wir uns verschiedene Text-Codierungen für die Übertragung mittels elektrischer Signale anschauen.

Übung 1

Welche Vor- und Nachteile haben die vorgestellten Systeme gegenüber der Nachrichtenübermittlung mittels Brief?

Übung 2

Welche Vorteile hat die Übertragung mittels elektrischer Signale gegenüber historischer akustischer oder optischer Signale?

Kapitel 2: Fernschreiber

Anfang des 20. Jahrhunderts verbreitete sich eine Weiterentwicklung des Morsetelegrafen: der Fernschreiber. Abbildung 3 zeigt ein Beispiel davon aus dem Jahre 1958. Im Grundsatz handelt es sich um zwei Schreibmaschinen, die miteinander verbunden sind. Wenn auf der einen Schreibmaschine etwas geschrieben wird, wird dasselbe auf der Empfänger-Maschine getippt. So können einfach und schnell Nachrichten übertragen werden. Bei manchen Modellen konnte die Eingabe auch von einer Lochkarte (s. Titelbild) eingelesen werden.

Abbildung 3: Der Fernschreiber T100 Siemens.

Die Daten bei einem Fernschreiber werden binär übertragen, d.h. als eine Abfolge von 0 und 1. Damit diese beim Empfänger korrekt interpretiert werden können, müssen sich Sender und Empfänger auf eine Codierung einigen.

Neben Buchstaben konnten dabei auch Steuerzeichen übertragen werden. Diese erlaubten es, die Empfängermaschine zu steuern, wie z.B. die Schreibposition wieder

an den Anfang der Zeile verschieben, das Papier um eine Zeile hochschieben, oder eine Glocke läuten lassen. Diese Steuerzeichen sind noch heute bei modernen Computern vorhanden, wenn auch die meisten davon nicht mehr verwendet werden.

Übung 3

Wir möchten eine Codierung für Fernschreiber entwerfen. Die Codierung sollte sowohl für die Übertragung als auch für die Speicherung verwendet werden können.

Bearbeite hierfür die folgenden Aufgaben:

- Wähle die Symbole (z.B. Buchstaben, Steuerzeichen) aus, die du übertragen möchtest.
- Beschreibe, welchem Symbol du welche binäre Darstellung zuordnest.
- Welche Überlegungen hast du getroffen bei der Zuweisung der binären Darstellung zu den Symbolen?

Kapitel 3: Ascii

Den Standard, auf den man sich für die Fernschreiber geeinigt hat, nennt man Ascii (American Standard Code for Information Interchange). Die heute verwendete Form wurde 1965 standardisiert.

Bei Ascii werden alle Symbole mit 7 Bit codiert, womit 128 verschiedene Symbole möglich sind. Grob kann man die Symbole in fünf Kategorien aufteilen: Steuerzeichen, Ziffern, Grossbuchstaben, Kleinbuchstaben, und Sonderzeichen.

In Tabelle 1 sind in Ascii verfügbare Symbole und ihre Codierung aufgeführt. Um die Codierung zu erhalten, wird zunächst die Zeilennummer in 3 Bits codiert und anschliessend die Spaltennummer in 4 Bits codiert angefügt. So wird z.B. der Grossbuchstabe E als 1000101_2 codiert (Zeile $4_{10} = 100_2$, Spalte $5_{10} = 0101_2$).

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SP	!	"	#	$	%	&	,	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

Tabelle 1: Die Ascii-Tabelle.

Die Steuerzeichen besetzen die ersten 32 Positionen (die ersten zwei Bits sind 00_2) sowie die letzte Position. Die Ziffern beginnen alle mit 011_2, gefolgt von der Ziffer in Binärdarstellung. Die Grossbuchstaben wurden von 1_{10} bis 26_{10} durchnummeriert und beginnen jeweils mit 10_2. Die Kleinbuchstaben unterscheiden sich genau im zweiten Bit: sie beginnen alle mit 11_2, sind sonst aber identisch zu den Grossbuchstaben. Die Sonderzeichen schlussendlich sind auf die verbleibenden Positionen verteilt.

Die meisten Steuerzeichen sind heutzutage im Kontext vom Computer nicht mehr relevant. Einige werden noch in spezifischen Kontexten genutzt. Um Text zu codieren wird oft noch HT als Tabulator und LF (Line Feed = Papier hochschieben) und CR (Carriage Return = Schreibposition an Anfang verschieben) für Zeilenumbrüche verwendet. Manche Systeme nutzen CR LF als Zeilenumbruch, während andere nur LF verwenden. Dies ist noch nützlich zu wissen, da dies beim Programmieren teilweise relevant sein kann.

Übung 4

Codiere die folgenden Zeichen mit Ascii:

- 5
- *
- H
- x
- BEL

Übung 5

Wenn du einen Kleinbuchstaben als Ascii codiert hast, wie kannst du einfach den zugehörigen Grossbuchstaben bestimmen?

Übung 6

Denk dir ein Wort oder einen Satz mit maximal 20 Zeichen aus. Codiere diesen mittels Ascii und gib nur den codierten Text an deinen Tischnachbar.

Verwende die Tabelle in Tabelle 1, um den Text, den du von deinem Tischnachbar erhalten hast, zu decodieren.

Vergleicht anschliessend, ob der Text korrekt übertragen wurde.

Kapitel 4: Verschiedene Sprachen

Viele Sprachen haben eine Vielzahl von Sonderzeichen oder gar komplett andere Schreibsysteme. Z.B. haben wir in Deutsch die Umlaute Ä, Ö, und Ü. Diese können mit normalem Ascii nicht codiert werden. Solange die Codierung nur für die Informa-

tionsübertragung genutzt wurde, konnte man gut damit umgehen. Z.B. hat man statt Ö einfach OE geschrieben. Doch wenn es nicht nur um die Information geht, sondern auch für die allgemeinere Datenverarbeitung z.B. auf einem Computer, möchte man auch die Sonderzeichen korrekt codieren können.

Übung 7

Wie könnte Ascii erweitert werden, um die deutschen Sonderzeichen codieren zu können?

Wenn wir pro Zeichen 8 Bit verwenden, können wir insgesamt 256 Symbole codieren. Neben den 128 Symbolen, die bereits in Ascii definiert sind, können wir also weitere 128 Symbole für Sonderzeichen nutzen. Damit können wir für viele in Europa und Amerika verbreitete Sprachen die Sonderzeichen codieren. Eine weit verbreitete Codierung hierfür ist z.B. Windows-1252.

Doch Sprachen mit einem anderen Schriftsystem, wie z.B. Griechisch, Russisch, oder Arabisch kann man so nicht codieren. Für diese Sprachen wurden dann eigene Erweiterungen entwickelt, die die zusätzlichen 128 Symbole für die entsprechende Sprache belegten.

Übung 8

Was sind mögliche Probleme, wenn wir verschiedene Codierungen für verschiedene Sprachen nutzen?

Kapitel 5: Unicode und UTF-8, UTF-16, UTF-32

Um die Probleme, die bei Texten in verschiedenen Sprachen auftreten, zu beheben, wurde 1991 Unicode veröffentlicht. Im Gegensatz zu Ascii und Windows-1252 geschieht die Codierung von Zeichen in zwei Schritten: Jedem Symbol ist eine Zahl zugeordnet und diese Zahl wird anschliessend in einer von mehreren Codierungen gespeichert.

Die ersten 128 Zeichen entsprechen genau Ascii. Anschliessend kommen die Sonderzeichen der lateinischen Schriften, gefolgt von weiteren europäischen Schriften. Unicode definiert einen Zahlenwert für alle heutzutage verbreiteten Schriftsysteme und Sprachen, aber auch weitere Symbole und Emojis.

Um den Zahlenwert als Binärdaten zu codieren, stehen verschiedene Methoden zur Verfügung. Am meisten verbreitet ist dabei UTF-8, teilweise wird auch UTF-16 genutzt. Es gäbe auch noch UTF-32, was aber eher selten verwendet wird.

Bei UTF-32 wird jedes Symbol mit 32 Bits dargestellt und der Zahlenwert wird einfach binär dargestellt. UTF-8 und UTF-16 haben eine variable Bitlänge. Bei UTF-8 hat jedes Symbol mindestens 8 Bits, bei UTF-16 sind es mindestens 16 Bits. Abhängig von den

ersten 8 bzw. 16 Bits kann man dann ablesen, wie viele Bits insgesamt für das aktuelle Symbol genutzt werden.

Übung 9

Wie würdest du eine variable Bitlänge wie in UTF-8 umsetzen? Hat deine Codierung hilfreiche Eigenschaften?

Übung 10

Was sind die Vorteile von fixer Bitlänge wie z.B. bei Ascii oder UTF-32 gegenüber variabler Bitlänge wie z.B. bei UTF-8 oder UTF-16?

Was sind die Vorteile von variabler Bitlänge gegenüber fixer Bitlänge?

Für die allermeisten Anwendungszwecke sollte heutzutage UTF-8 verwendet werden. Andere Codierungen werden primär noch aufgrund von Altlasten verwendet.

Kapitel 6: Was haben wir gelernt

Um Informationen zu übertragen oder zu speichern, müssen diese mit den zur Verfügung stehenden Signalen codiert werden. Dies können z.B. Zeigerpositionen, Morse-Code oder Binärdaten sein. Der Sender und Empfänger müssen sich einig darüber sein, wie die Signale zu interpretieren sind, indem sie sich auf eine Codierung verständigen.

Ursprünglich für Fernschreiber entwickelt, ist Ascii eine der grundlegenden Codierungen, die ein Computer verwendet. Um auch internationale Symbole zu unterstützen, gab es Versuche, dies mittels verschiedener Codierungen umzusetzen. Die bessere Lösung war dann Unicode und UTF-8, die heutzutage wo möglich die Standard-Codierung bei Computern ist.

Kapitel 7: Bild-Quellen

- Titelbild: CC by Till Niermann, https://de.wikipedia.org/wiki/Datei:Fortran_punch_card_ESA.jpg
- Abbildung 1: CC0, https://de.wikipedia.org/wiki/Datei:Coast_telegraph,_semaphore_or_signaling_post_at_Scheveningen,_1799.jpg
- Abbildung 2: https://de.wikipedia.org/wiki/Datei:L-Telegraph1.png
- Abbildung 3: CC by Nightflyer, https://de.wikipedia.org/wiki/Datei:Fernschreiber_T100_Siemens.jpg

Kapitel 8: Lösungen

8.1. Kapitel 1: Historische Verfahren

Übung 1: **Vorteile**: Sehr schnell

Nachteile: Sender und Empfänger müssen sich auf eine Codierung einigen, Sender und Empfänger müssen zur selben Zeit „anwesend“ sein.

Übung 2: Die elektrischen Signale können über viel weitere Distanzen übertragen werden. Falls das Signal verstärkt werden muss, kann dies ohne das Zutun von Menschen erfolgen, was z.B. bei Semaphor-Netzwerken nicht der Fall ist.

Bei moderneren Technologien gibt es auch optische Verfahren, die genauso gut, wenn nicht sogar besser, funktionieren wie elektrische Signale: Glasfaserkabel verwenden optische Signale für die Datenübertragung und bilden einen wichtigen Bestandteil der Internetinfrastruktur.

8.2. Kapitel 2: Fernschreiber

Übung 3: Die möglichen Lösungen sind vielfältig. Mögliche Probleme:

- Codierung ist nicht Präfix-frei
- Buchstaben sind nicht alphabetisch geordnet
- Sonderzeichen, Zahlen und Buchstaben sind nicht in einer für die Sortierung sinnvoller Ordnung
- Gross- und Kleinbuchstaben können nicht leicht ineinander umgewandelt werden
- Variable Bit-Breite ohne einfache Möglichkeit, in die Mitte zu springen
- Ineffiziente Verwendung der Bit-Breite (z.B. unäre Codierung)

8.3. Kapitel 3: Ascii

Übung 4:

- 5: 0110101_2
- *: 0101010_2
- H: 1001000_2
- x: 1111000_2
- BEL: 0000111_2

Übung 5: Das zweite Bit kann auf 0 gesetzt werden. Dies ist äquivalent dazu, 32_{10} abzuziehen

8.4. Kapitel 4: Verschiedene Sprachen

Übung 7: Wir könnten eine erweiterte Codierung mit 8 Bits verwenden. Wenn das erste Bit 0 ist, verwenden wir normales Ascii für die restlichen 7 Bits. Falls das erste Bit 1 ist, definieren wir unsere eigene Codierung mit den restlichen 7 Bits für alle benötigten Sonderzeichen.

Übung 8:

- Es muss immer klar sein, mit welcher Codierung ein Text gespeichert wurde, um ihn richtig darstellen zu können
- Texte, die Symbole aus verschiedenen Schreibsystemen nutzen, können nicht gespeichert werden.

8.5. Kapitel 5: Unicode und UTF-8, UTF-16, UTF-32

Übung 9: Tabelle 2 zeigt, wie die variable Bitlänge im Falle von UTF-8 umgesetzt wurde. _ steht dabei für ein frei setzbares Bit.

Eine interessante Eigenschaft dieser Codierung ist, dass man mitten in einen Text springen kann und dann maximal 3 Bytes nach links gehen muss, um den Anfang eines Symbols zu finden: Falls das aktuelle Byte mit 10_2 beginnt, ist es nicht am Anfang.

Wenn nur ein Byte verwendet wird, entspricht die Codierung genau Ascii.

Byte 1	Byte 2	Byte 3	Byte 4
0_______			
110_____	10_____		
1110____	10_____	10_____	
11110___	10_____	10_____	10_____

Tabelle 2: Variable Bitlänge in UTF-8.

Übung 10: Bei der fixen Bitlänge kann man sofort zu einer bestimmten Position im Text springen. Die variable Bitlänge benötigt weniger Speicherplatz.