2.2 Codierung/Von ASCII bis UNICODE

2.2.1 ASCII

Lange Zeit waren für die Speicherung und Übertragung von Texten in Computern verschiedene Codes verbreitet. Eine Vereinheitlichung gelang durch ASCII (*american standard code for information interchange*, Tabelle 1). ASCII besitzt 7 Bit Breite, damit kann man 128 verschiedene

Nr.	Z.	Nr.	Z.	Nr.	Z.	Nr.	Z.	Nr.	Z.	Nr.	Z.	Nr.	Z.	Nr.	Z.
0		16		32	␣	48	0	64	@	80	P	96	`	112	p
1		17		33	!	49	1	65	A	81	Q	97	a	113	q
2		18		34	"	50	2	66	B	82	R	98	b	114	r
3		19		35	#	51	3	67	C	83	S	99	c	115	s
4		20		36	$	52	4	68	D	84	T	100	d	116	t
5		21		37	%	53	5	69	E	85	U	101	e	117	u
6		22		38	&	54	6	70	F	86	V	102	f	118	v
7		23		39	'	55	7	71	G	87	W	103	g	119	w
8		24		40	(	56	8	72	H	88	X	104	h	120	x
9		25		41	)	57	9	73	I	89	Y	105	i	121	y
10		26		42	*	58	:	74	J	90	Z	106	j	122	z
11		27		43	+	59	;	75	K	91	[	107	k	123	{
12		28		44	,	60	<	76	L	92	\	108	l	124	|
13		29		45	-	61	=	77	M	93	]	109	m	125	}
14		30		46	.	62	>	78	N	94	^	110	n	126	~
15		31		47	/	63	?	79	O	95	_	111	o	127	

Tabelle 1: ASCII, auch Fernschreibcode CCITT-5 genannt

Zeichen darstellen. Im ASCII sind Groß- und Kleinbuchstaben vorhanden, sie liegen jeweils 32 Positionen weit auseinander. Sowohl Ziffern (ab Nr. 48) als auch Buchstaben (ab Nr. 65 bzw. 97) sind fortlaufend angeordnet. Zwischen diesen großen Blöcken liegen die Satzzeichen. Sämtliche Steuerzeichen liegen an den Positionen null bis 31, so z.B. LF (Nr. 10), CR (Nr. 13), Tabulator (Nr. 9) und Rückschritt (Nr. 8). Manche dieser Steuerzeichen dienten dem Aufbau und Abbau von Fernschreibverbindungen und sind daher heute nicht mehr gebräuchlich. Ansonsten ist am ASCII nicht viel auszusetzen – außer, dass Sprachen außer Englisch nicht berücksichtigt wurden.

2.2.2 ASCII-D

Wenn man mit ASCII deutsche Umlaute benutzen will, kann man eine Abwandlung benutzen, bei der die Umlaute an den Code-Positionen eingesetzt werden, die man in Texten nicht braucht: So liegen Ä, Ö und Ü an den Positionen von `[`, `\` und `]`. Die Kleinbuchstaben ä, ö und ü ersetzen die Zeichen `{`, `|` und `}` und das ß findet man anstelle der Tilde `~`. Diese Lösung hat den Nachteil, dass die eckigen und geschweiften Klammern einfach nicht mehr verfügbar sind. Außerdem kann man Texte damit nicht zwischen verschiedenen Ländern austauschen, denn für jedes Land gibt es nun einen eigenen leicht abgewandelten ASCII-Zeichensatz,

2.2.3 ASCII-Erweiterungen bis ISO-Latin-1

Mit dem Aufkommen von Mikrocomputern in den achtziger Jahren wurde für diese Kleinstrechner (denn das bedeutete ja das Wort Mikrocomputer) eine Wortbreite von 8 Bit üblich. Bei größeren Rechnern setzten sich damals Wortbreiten von 16, 32 und 64 durch. Da lag es nahe, jeweils ein ASCII-Zeichen in einem 8-Bit-Wort, einem Byte zu speichern und das 8. Bit freizulassen. Man konnte aber auch den ASCII erweitern um 128 neue Zeichen. Und auf diese Idee kamen mehrere Firmen, die ihre IT-Systeme in andere Länder (meist westeuropäische) verkaufen wollten. So gab und gibt es eigene Zeichensätze für IBM-PCs (heute noch im DOS-Fenster und im BIOS zu

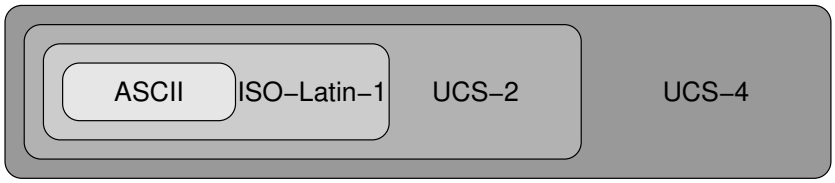

Abbildung 1: Kompatibilität der alphanumerischen Codes von ASCII bis UCS4

besichtigen), für Macs (nicht mehr verwendet), für HP-Rechner und viele weitere. Der IBM-PC-Code hatte dabei die Eigenschaft, viele Codes für so genannte Blockgrafik zu reservieren, also für Zeichen, aus denen man Zeichnungen aus senkrechten und waagerechte Linien konstruieren konnte.

Damit auch dieser Wildwuchs ein Ende hatte, wurde von der ISO eine ASCII-Erweiterung festgelegt, die eine ganze Reihe west- und mitteleuropäischer Sprachen berücksichtigte. Sie heißt ISO-Latin-1 bzw. ISO-8859-1. In Windows wird sie (fälschlich) als ANSI-Code bezeichnet. Mit den weiteren Codes ISO-8859-2, ISO-8859-3 usw. werden weitere Sprachen abgedeckt.

2.2.4 UCS4 und UCS2, UNICODE

Mit der zunehmenden Globalisierung in den neunziger Jahren wuchs der Bedarf, eine einheitliche Codierung für alle Textdokumente weltweit zu ermöglichen. Man setzte eine Breite von 32 Bit für jedes Codewort fest und ging daran, alle 4 Milliarden Möglichkeiten für Codes auszunutzen. Die ersten 256 Codes waren dabei kompatibel zu ISO-Latin-1. Dieser Code heißt UCS4 (*universal character set, 4 bytes*).

Leider dauerte das einigen Firmen zu lange. Sie setzten auf eine Wortbreite von nur 16 Bit, die eben für die meisten Sprachen auf der Welt reichen sollten. Dieser Code sollte UCS2 oder UNICODE (*universal code* heißen. Er hätte 65536 verschiedene Zeichen ermöglicht.

Da die Macher von UCS4 und die von UNICODE weitgehend identisch waren, wurde schließlich ein Kompromiss daraus: Man integrierte die ersten 65536 Zeichen von UCS4 in den UNICODE.

Aus den angepeilten 65536 Zeichen sind allerdings inzwischen über 180000 geworden. Damit reichen die 16 Bit von UNICODE nicht mehr aus. So hat man weiter getrickst und nennt die weiteren 120000 Zeichen UCS4 auch UNICODE.

Wenn man jetzt ein Zeichen mit einer Position größer als 65535 mit zwei Byte speichern oder übertragen will, muss man ein Zeichen in mehreren Schritten übertragen (so dass man für dieses Zeichen dann doch vier oder sogar sechs Byte braucht). Diese weitere Codierung nennt man UTF-16 (*UCS transformation format, 16 bit*).

Der große Vorteil bei UCS4 (oder UCS2 oder UNICODE) ist, dass für (fast) jeden denkbaren Einsatzzweck Zeichen vorhanden sind: Alle lebenden Sprachen sind dabei (auch CJK=chinesisch, japanisch, koreanisch), viele nicht-mehr-gesprochene Sprachen (Keilschrift), Fantasiesprachen (klingonisch, elbisch) und viele Sonderzeichen und Symbole. Wer eine kleine Auswahl sehen will, sollte unter Linux das Programm `gucharmap` (Zeichentabelle) öffnen.

2.2.5 Kompatibilität von ASCII bis UCS4

Abbildung 1 zeigt, welche der genannten Codes zueinander kompatibel sind.