Multimedia

§2 Text coding

Prof. Dr. Georg Umlauf

Content

§2.1 Basics of character coding

- **Character:**

  Elementary text entity

  ➡ atomic information entity to represent, organize, or control text

  - Example: Letter, digit, punctuation marks, accents, graphical symbols, ideographic symbols, space, tabulators, line feed, control codes, etc.

- Not to be confused with:
  - **Glyph:** graphical representation of a symbol
    - E.g.: the abstract form of "A", realized by **glyph images** A *A* A *A* *A*
  - Input symbols (key press)
  - Phonetic entities (phoneme, syllable, word)

§2.1 Basics of character coding

- **Plain text:**

  Written text as sequence of elementary text entities (characters), such as letters, digits, punctuation marks, etc. including

  - **escape mechanisms** (example: cursor positioning) and
  - **markup** (formatting instructions).
  - **E.g.:** XML defined as plain text, structured by syntactical rules.

- **Fancy text:**

  Text with font attributes, margin alignment, page arrangement, etc.

§2.1 Basics of character coding

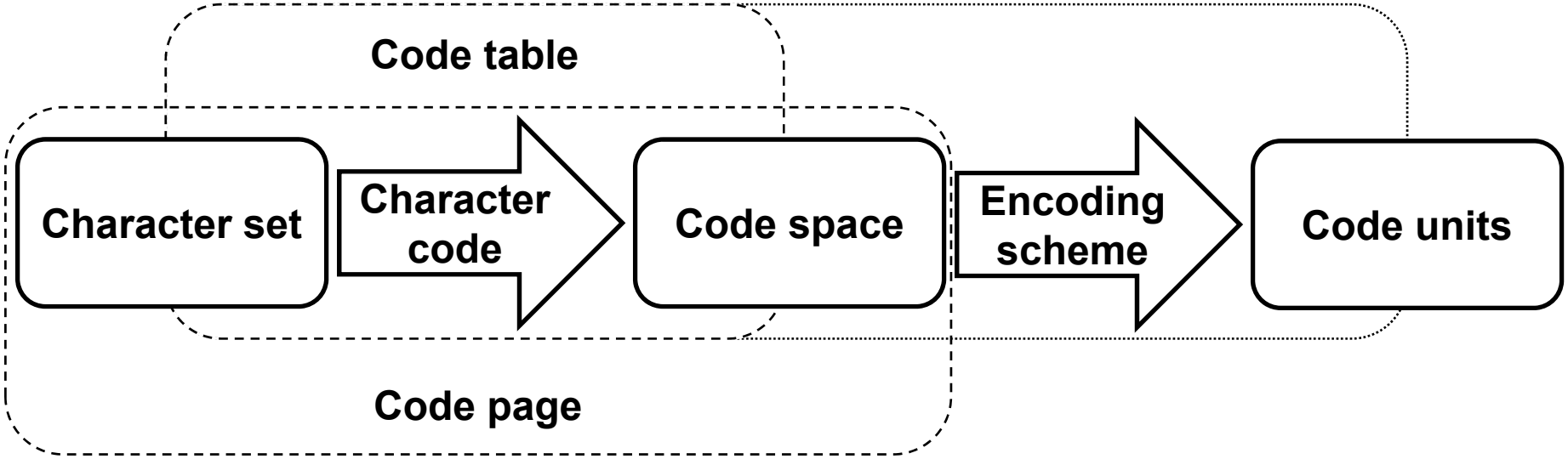
- **Character set:**
  Pool of possible characters
  - E.g.: Letters of an alphabet, digits, symbols, etc.
- **Character code:** Unique mapping of characters of a character set to so-called **code positions** of the **code space.**
- **Code positions** are usually natural numbers.
- **Encoding scheme:** Method to represent code positions in the computer as bit- or byte-sequences (**code units**).

§2.1 Basics of character coding

- The size (cardinality) of the code space determines the number of bits of the code units necessary to represent all code positions.
  - E.g.:
    - 8 Bit = 256 values in the code space,
    - 16 Bit = 65.536 values in the code space.

§2.1 Basics of character coding

- **Code tables:** Mapping of characters to code positions rsp. code-units in table form:
  - Characters and assigned code positions are given in a table.
  - For a trivial encoding scheme characters and assigned code units are given in a table.
- **Code page:** Aggregation of character set, code space and character code.



§2.1 Basics of character coding

- **Text coding:** Transformation of a text in bits and bytes for the computerized representation following the rules of a character coding scheme.

➡ A text becomes the sequence of code positions of its characters.

➡ In the computer, a text is represented as the sequence of code units of its characters.

- **Problems:**
  - Huge number of characters and
  - numerous national coding schemes.

§2.1 Basics of character coding

Synonymous terms (German and English)

- **Character:** Schriftzeichen, Zeichen
- **Character Set:** Zeichenvorrat, Zeichensatz, character repertoire
- **Character coding:** Zeichencodierung, Codierungsschema, coded character set
- **Code Position:** Code-Punkt, Code, code set position, code point, character number
- **Code Space:** Code-Raum, Code-Menge, code set
- **Encoding Scheme:** character encoding form (cef), encoding form, character encoding scheme (ces)
- **Code Table:** Code-Tabelle, coded character set (ccs), character code.

§2.1 Basics of character coding

E.g.: ASCII

Content

§2.2 Standards

ASCII: American **S**tandard **C**ode for **I**nformation **I**nterchange

- 1963-1968 developed: US-ASCII (ANSI X3.4).
- **Character set:** printable characters of the English alphabet (incl. space character) and some control characters (line feed, etc.).
- **Code space:** natural numbers 0 – 127, i.e. 0 – 7F.
- **Code positions:**
  - 32 – 126 printable characters (95 characters),
  - 0 – 31, 127 control characters (33 characters).
- **Coding scheme:** Every characters is mapped to a natural number of the code space in its binary representation stored in one byte.
  - ➡ The most significant bit is zero **(7-Bit-ASCII)**.
  - ➡ Coding scheme and encoding form coincide.
  - ➡ Code tables suffice for the compete characterization.

§2.2 Standards

ASCII code table (hexadecimal)

Code	. 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*

Further Standards

- **ISO 646** corresponds basically to US-ASCII, with some national special characters (e.g. Umlauts).
  - There are 16 national variants of ISO 646.
  - E.g.: ISO 646-DE for German.
- **ISO 8859-1** (ISO Latin 1) extends the code tables of US-ASCII to characters at code positions 128 to 255.
  - Code space: 0 – FF.
  - Encoding scheme is trivial (8-Bit-ASCII), i.e. using code tables.
  - Other ASCII extensions in the ISO 8859 family:
    - **ISO 8859-9** (ISO Latin 5) fur Turkish and
    - **ISO 8859-15** (ISO Latin 9) with the Euro-Symbol.

§2.2 Standards

ISO-8859-15 Cod table

8.	*PAD*	*HOP*	*BPH*	*NBH*	*IND*	*NEL*	*SSA*	*ESA*	*HTS*	*HTJ*	*VTS*	*PLD*	*PLU*	*RI*	*SS2*	*SS3*
9.	*DCS*	*PU1*	*PU2*	*STS*	*CCH*	*MW*	*SPA*	*EPA*	*SOS*	*SGCI*	*SCI*	*CSI*	*ST*	*OSC*	*PM*	*APC*
A.	*NBSP*	¡	¢	£	€	¥	Š	§	š	©	ª	«	¬	*SHY*	®	¯
B.	°	±	²	³	Ž	µ	¶	·	ž	¹	º	»	Œ	œ	Ÿ	¿
C.	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D.	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E.	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F.	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

§2.2 Standards

- Standardized code tables

Language	Code table	Cardinality
English	US-ASCII (ISO 646:1991)	95
German/French	ISO 8856-1:1987	191
Chinese	GB 2312-80	7.455
Chinese	Big 5	13.523
Japanese	JIS X 0208-1990	6.897
Korean	KS C 5601-1992	8.224
All	ISO/IEC 10646-1:2000 (Unicode 3)	> 100.000

- ... and many more: EBCDIC, ISO/IEC 6937:2001, etc.

Content

§2.3 Unicode

- **Goal**
  - Coding of all alphabets of the world.
  - Basis for the implementation of word processing of arbitrary texts.
- **Design principles**
  - Unicode codes characters, not glyphs.
  - Unicode codes plain text, no markup.

➡ **Coding model:** conceptual frame to structure the coding of some billion characters to bit pattern.

It consists of:

- Character sets
- Character codes using code tables
- Non-trivial encoding scheme (**code format**)

§2.3 Unicode

- Character set:

  Universal Character Set (**UCS**) rsp. **ISO/IEC 10646**

  - Defines a universal character set, which is meant to cover all alphabets of the world.
  - Contains today more than 100,000 characters.

- Character code:

  Unicode, UCS rsp. **ISO/IEC 10646** defines a code table for UCS

  - Codes today more than 100,000 characters.
  - Standardized character sets are already covered:
    - BMP (Basic Multilingual Plane): Unicode characters, with code in the range 0-FFFF (65.536 characters).

§2.3 Unicode

- Comparison of code spaces

Coding	Code space	Code positions	Code length [bit]
US-ASCII	0-1F	128	7
ISO 8859	0-FF	256	8
Unicode	0-10FFFF	65.536+1.048.576	„21“
ISO/IEC 10646	0-7FFFFFFF	2.147.483.648	31

Unicode

- Character set is subdivided in 17 planes of 2^{16} = 65.536 characters each.
- Each plane is subdivided in blocks for different alphabets.
  - E.g.: Latin, Greek, Cyrillic, etc.
- Only six planes are used today.

Unicode planes

Name	Description	Plane
BMP	*Basic Multilingual Plane:* currently used alphabets, punctuation marks, symbols, control characters, etc.	0
SMP	*Supplementary Multilingual Plane:* historical alphabets, collection of special characters, etc.	1
SIP	*Supplementary Ideographic Plane:* Rare CJK-characters.	2
TIP	*Tertiary Ideographic Plane:* Empty, but reserved.	3
	Unused	4-13
SSP	*Supplementary Special-purpose Plane:* Control characters for language identification.	14
PUA	*Supplementary Private Use Area-A* und *–B:* For private use based on individual agreement between sender/receiver of a text.	15+16

§2.3 Unicode

- Comparison of code formats

Format	Coding	Code space	Code length [byte]
	US-ASCII	0-1F	fix, 1
	ISO 8859	0-FF	fix, 1
UCS-2	ISO 10646	0-FFFF	fix, 2
UCS-4	ISO 10646	0-7FFFFFFF	fix, 4
UTF-32	Unicode 3.0	0-10FFFF	fix, 4
UTF-8	Unicode 3.0	0-10FFFF	variable, 1-4
UTF-8	ISO 10646	0-7FFFFFFF	variable, 1-6
UTF-16	Unicode 3.0	0-10FFFF	variable, 2/4

  - UTF = Unicode Transformation Format

§2.3 Unicode

UTF-8

- Variable length of code units of 1-6 Bytes.
  - Encoded code positions in the range of 0-FFFF using 1-3 Bytes (BMP).
  - Encoded code positions in the range of 0-7FFFFFFF using 1-6 Bytes.
- Transparent for binary numbers from 0 to 127, coded with one byte with most significant bit zero.
  - ➡ US-ASCII-downward-compatible
- Multi-Byte-sequences have a leading byte and 1-5 continuation bytes.
  - Number of leading „1“ in leading byte (succeeded by one „0“) yields the number of used bytes for the corresponding code position.
  - Continuation bytes start with the bit sequence „10“.
  - The remaining bits are used to encode the code positions (padded with leading „0“ where necessary).

§2.3 Unicode

UTF-8 Encoding

Bytes	UTF-8 representation		Bits	Largest code position
	Leading byte	Continuation bytes		
1	0xxxxxxx		7	7F
2	110xxxxx	10xxxxxx	11	7FF
3	1110xxxx	10xxxxxx 10xxxxxx	16	FFFF
4	11110xxx	10xxxxxx 10xxxxxx 10xxxxxx	21	1FFFFF
5	111110xx	10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx	26	3FFFFFF
6	1111110x	10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx	31	7FFFFFFF

UTF-8

- Pros
  - Self-synchronizing since the beginning of a code position can be at most five bytes earlier.
    - At an interruption, bytes starting with „10“ are ignored.
    - If „0...“ or „11...“ is detected, a new code position starts.
  - Automatic detection of UTF-8.
    - ➡ Usage for the internet.
- Cons
  - There are multiple code units for the same character:
    - E.g.: „a“ is coded as *0***1100001** or erroneously as *110*00001 *10***100001**
  - ➡ Only the shortest code unit is valid.
  - ➡ There are certain invalid bit-/byte-sequences.

§2.3 Unicode

UTF-16

- Represents every valid code position in the range of 0-FFFF canonically by two bytes
  - ➡ UCS-2 downward compatible.
- Represents code positions in the range of 10000 to 10FFFF (20 bits) with two surrogate positions (surrogate of 2 byte)
  - Upper surrogate range D800 to DBFF: 110110xxxxxxxxxx
  - Lower surrogate range DC00 to DFFF: 110111xxxxxxxxxx
- E.g.:

Character	Unicode	UTF-16BE binary	UTF-16BE hex
y	0079	00000000 01111001	00 79
ä	00E4	00000000 11100100	00 E4
€	20AC	00100000 10101100	20 AC
𝄞	1D11E	11011000 00110100 11011101 00011110	D8 34 DD 1E

Goals

- What is the code space and what is an encoding scheme?
- What is text coding?
- What is ASCII and what is the size of its code space?
- What is the Unicode BMP?