Tegn og tekst

\yvind og]se N{rb} ?

Læreboka kapittel 2

29. August 2007

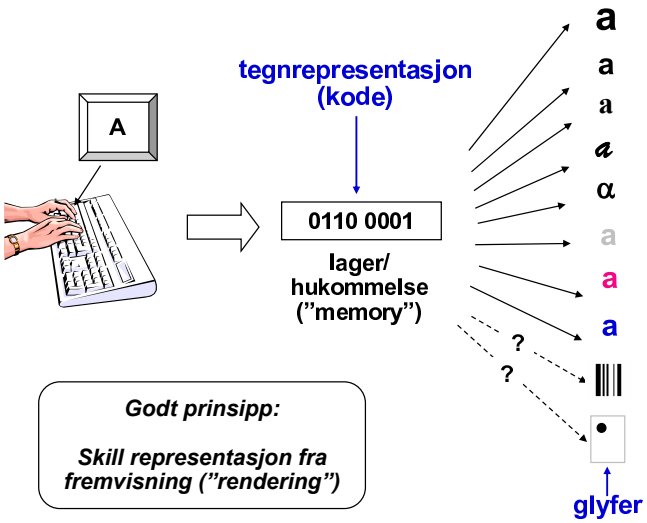
Problemstilling

- Utgangspunkt: Hvert tegn i teksten representeres av et unikt bitmønster.
  Eksempel: Med E = 01000101, H = 01001000, I = 01001001 får vi HEI = 01001000 01000101 01001001
- Sender (skriver) og mottaker (leser) må være enige om kodingen.
  - Vi trenger standarder!
- Aktuelle spørsmål:
  - Hvilke tegn skal representeres?
  - Hvor mange biter per tegn?
  - Fast eller variabelt antall biter per tegn?
  - Hvordan håndtere
    - » tegn som er varianter av andre tegn?
    - » ligaturer (sammensetninger)?
  - Bør det være noen form for systematikk i bitmønstrene?

Om tegn og glyfer

- **Tegn**
  Det bakenforliggende begrep for bestemte "strektegninger" på papir, skjerm, steintavler...
- **Glyf**
  Et tegn kan vises fram med ulike *glyfer*: A A *A* A …
- **Kontrollkode/kontrolltegn**
  "Tegn" som ikke vises fram i form av en glyf, men som brukes til å styre eller påvirke fremvisningsenheten eller dataoverføringen
  Eksempel: ASCII-koden 0000111_2 (Audible bell)
  får fremvisningsenheten til å gi lyd fra seg

Et representert tegn kan vises på flere måter



Tegnkoder og kodetabeller

- **Kode**
  Noe som representerer noe annet
  Eksempel: Heksadesimal notasjon, se neste lysark
  ASCII-koden 1000001_2 = 0x41
  representerer tegnet A
- **Kodepunkt**
  Et tegns "numeriske" verdi
  Eksempel: 0x41 er kodepunktet for A i ASCII
- **Kodetabell**
  En ordnet liste av koder og hva de representerer
- **Koding ("Encoding")**
  1. Oppsett av en kodetabell
  2. Kodingsprinsipp brukt i forbindelse med dataoverføring

kodepunkt	
A	0x41
B	0x42
C	0x43

Kodetabell

Fra binær til heksadesimal og vice versa

Fra binær til heksadesimal

- Grupper de binære sifrene 4 og 4 (bakfra)
- Erstatt hver gruppe med det tilsvarende heksadesimale sifferet
- Eksempel:
  $0110\ 0001_2 = 0x61$

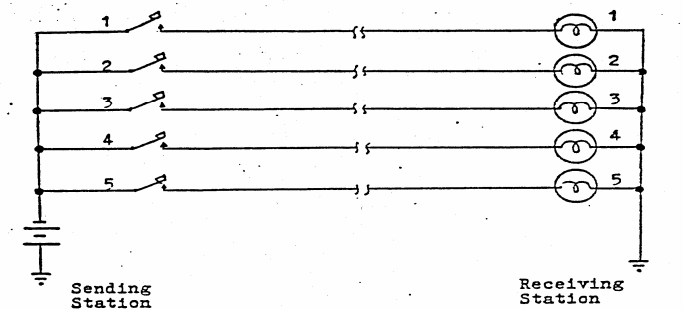
Fra heksadesimal til binær

- Bruk tabellen motsatt vei

binær siffergruppe	heksadesimalt siffer
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Baudots femlednings-system (1874)

- Fransk ingeniør (1845-1903)
- Brukte fem ledninger for å overføre ett tegn
- Første eksempel på fast antall biter per tegn i den elektriske verden
- Et genuint binært system!



Baudot kodetabell (1870 →)

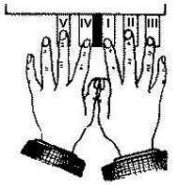
- 5-biters kode
  – 32 ulike koder
- To plan
  – LTRS og FIGS
  – statusavhengig koding
- Få kontrolltegn

LTRS	00	10
0	undef	LTRS
1	A	C
2	E	X
3	É	Z
4	I	S
5	O	T
6	U	W
7	Y	V
8	FIGS	DEL
9	J	K
A	G	M
B	H	L
C	B	R
D	C	Q
E	F	Z
F	D	P

FIGS	00	10
0	undef	LTRS
1	1	.
2	2	,
3	&	:
4	3	;
5	4	!
6	O	?
7	5	'
8	FIGS	DEL
9	6	(
A	7	)
B	H	=
C	8	-
D	9	/
E	F	№
F	0	%

Baudot - sendeenhet

- Kodetabellen utformet med tanke på håndens ergonomi



LTRS	none	IV	V	both
none	undef	FIGS	LTRS	DEL
I	A	J	C	K
II	E	G	X	M
I II	É	H	Z	L
III	I	B	S	R
III I	O	C	T	Q
III II	U	F	W	Z
III II I	Y	D	V	P

FIGS	none	IV	V	both
none	undef	FIGS	LTRS	DEL
I	1	6	.	(
II	2	7	,	)
I II	&	H	:	=
III	3	8	;	-
III I	4	9	!	/
III II	O	F	?	№
III II I	5	0	'	%

6-biters tegn kodetabell (1960 →)

- Mange ulike varianter – til og med på samme maskintype
  Eksempel her:
  UNIVACs 6-bits FIELDATA kodetabell
- Ofte bygd på FIELDATA (en US-Army standard)
- Meget utbredt på 60-tallets datamaskiner
- Gir rom for A – Z, 0 – 9, noen spesialtegn
- Ikke små bokstaver, ingen nasjonale tegn (Æ, Ø, Å)
- Tall og bokstaver systematisk plassert

	00	10	20	30
0	@	K	)	0
1	[	L	-	1
2	]	M	+	2
3	#	N	<	3
4	Δ	O	=	4
5	space	P	>	5
6	A	Q	&	6
7	B	R	$	7
8	C	S	*	8
9	D	T	(	9
A	E	U	%	’
B	F	V	:	;
C	G	W	?	/
D	H	X	!	.
E	I	Y	,	¤
F	J	Z	\	≠

ASCII (1963 →)

- ASCII
  – American Standard Code for Information Interchange
- 7-biters kode – 128 tegn
- Meget gjennomtenkt standard
  – brukes den dag i dag, og er inkludert i nyere standarder
- Gir rom for A – Z, a – z, 0 – 9, mange spesialtegn
- ... men ikke for internasjonale tegn (æ, ø, å)

se www.jimprice.com/jim-asc.htm

ASCII kodetabell (1963 →)

	00	10	20	30	40	50	60	70
0	NUL	DLE	space	0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	”	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

Hva betyr kontrolltegnene?

- NUL (Null): No character
- BEL (Bell): Used to control an alarm or attention device.
- BS (Back Space): Indicates the movement of the printing mechanism or display cursor one position backwards.
- HT (Horizontal Tab): Indicates the movement of the printing mechanism or display cursor forward to the next preassigned "tab" or stopping position.
- LF (Line Feed): Indicates movement of the printing mechanism or display cursor to the next line.
- CR (Carriage Return): Indicates movement of the printing mechanism or display cursor to the starting position of the same line.
- ESC (Escape): Intended to provide code extension in that it gives a specified number of contiguous following characters an alternate meaning.
- SP (Space): A nonprinting character used to separate words, or to move the printing mechanism or display cursor forward by one position.
- DEL (Delete): Used to obliterate unwanted characters (for example, on paper tape by punching a hole in every bit position).

se også http://www.tomrobinson.co.nz/work/iso8859.html

ISO 646-60 kodetabell

Bygger på ASCII, men

[\ ] { | }
er ofret til fordel for
Æ Ø Å æ ø å

- Lignende tilpasninger er gjort i tilsvarende standarder for andre språkmiljøer

	00	10	20	30	40	50	60	70
0	NUL	DLE	space	0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	”	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	Æ	k	æ
C	FF	FS	,	<	L	Ø	l	ø
D	CR	GS	-	=	M	Å	m	å
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

8 biter: Extended ASCII og ISO 8859 (1968 →)

- 8859-1: Latin Alphabet No. 1 (Vest-Europa)
- 8859-2: Latin Alphabet No. 2 (slavisk, ungarsk, romansk)
- 8859-3: Latin Alphabet No. 3 (esperanto, maltesisk)
- 8859-5: Latin/Cyrillic Alphabet
- 8859-6: Latin/Arabic Alphabet
- 8859-7: Latin/Modern Greek Alphabet
- 8859-8: Latin/Hebrew Alphabet
- 8859-9: Latin Alphabet No. 5 (moderne tyrkisk)
- 8859-13: Latin Alphabet No. 7 (islandsk, grønlandsk, baltisk, nordsamisk)
- 8859-14: Latin Alphabet No. 8 (keltisk)
- 8859-15: Latin Alphabet No. 9 (modernisert 8859-1, med bl.a. euro-tegn)

ISO 8859-1 (Latin-1) kodetabell

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
0	NUL	DLE	space	0	@	P	`	p			no break space	°	À	Ð	à	ð
1	SOH	DC1	!	1	A	Q	a	q			¡	±	Á	Ñ	á	ñ
2	STX	DC2	”	2	B	R	b	r			¢	²	Â	Ò	â	ò
3	ETX	DC3	#	3	C	S	c	s			£	³	Ã	Ó	ã	ó
4	EOT	DC4	$	4	D	T	d	t			¤	´	Ä	Ô	ä	ô
5	ENQ	NAK	%	5	E	U	e	u			¥	µ	Å	Õ	å	õ
6	ACK	SYN	&	6	F	V	f	v	un-		¦	¶	Æ	Ö	æ	ö
7	BEL	ETB	'	7	G	W	g	w	defined		§	·	Ç	×	ç	÷
8	BS	CAN	(	8	H	X	h	x			¨	¸	È	Ø	è	ø
9	HT	EM	)	9	I	Y	i	y			©	¹	É	Ù	é	ù
A	LF	SUB	*	:	J	Z	j	z			ª	º	Ê	Ú	ê	ú
B	VT	ESC	+	;	K	[	k	{			«	»	Ë	Û	ë	û
C	FF	FS	,	<	L	\	l	|		¬	¼	Ì	Ü	ì	ü
D	CR	GS	-	=	M	]	m	}			-	½	Í	Ý	í	ý
E	SO	RS	.	>	N	^	n	~			®	¾	Î	Þ	î	þ
F	SI	US	/	?	O	_	o	DEL			¯	¿	Ï	ß	ï	ÿ

ISO 8859-15 (Latin-15) kodetabell

ISO 8859-1 modernisert

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
0	NUL	DLE	space	0	@	P	`	p			no break space	°	À	Đ	à	ð
1	SOH	DC1	!	1	A	Q	a	q			¡	±	Á	Ñ	á	ñ
2	STX	DC2	”	2	B	R	b	r			¢	²	Â	Ò	â	ò
3	ETX	DC3	#	3	C	S	c	s			£	³	Ã	Ó	ã	ó
4	EOT	DC4	$	4	D	T	d	t			€	Ž	Ä	Ô	ä	ô
5	ENQ	NAK	%	5	E	U	e	u			¥	µ	Å	Õ	å	õ
6	ACK	SYN	&	6	F	V	f	v	un-		Š	¶	Æ	Ö	æ	ö
7	BEL	ETB	'	7	G	W	g	w	defined		§	·	Ç	×	ç	÷
8	BS	CAN	(	8	H	X	h	x			š	ž	È	Ø	è	ø
9	HT	EM	)	9	I	Y	i	y			©	¹	É	Ù	é	ù
A	LF	SUB	*	:	J	Z	j	z			ª	º	Ê	Ú	ê	ú
B	VT	ESC	+	;	K	[	k	{			«	»	Ë	Û	ë	û
C	FF	FS	,	<	L	\	l	|		¬	Œ	Ì	Ü	ì	ü
D	CR	GS	-	=	M	]	m	}			-	œ	Í	Ý	í	ý
E	SO	RS	.	>	N	^	n	~			®	Ÿ	Î	Þ	î	þ
F	SI	US	/	?	O	_	o	DEL			¯	¿	Ï	ß	ï	ÿ

8 biter – produsentspesifikke varianter

- IBMs EBCDIC (1963 –-)
- Microsoft Windows-1252: ISO 8859-1 pluss 28 tegn
- Macintosh – MacRoman Encoding

Designere av PCs, Macs og UNIX programvare har valgt ulike måter for representasjon av linjeskift:

PCs: 0x0D pluss 0x0A (CR pluss LF)

Macs: 0x0D

UNIX: 0x0A

Dette gjør filoverføring tricky!

Windows 1252 kodetabell

Bygger på ISO 8859-1

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
0	NUL	DLE	space	0	@	P	`	p	€	undef	no break space	°	À	Đ	à	ð
1	SOH	DC1	!	1	A	Q	a	q	undef	‘	¡	±	Á	Ñ	á	ñ
2	STX	DC2	”	2	B	R	b	r	‚	’	¢	²	Â	Ò	â	ò
3	ETX	DC3	#	3	C	S	c	s	ƒ	“	£	³	Ã	Ó	ã	ó
4	EOT	DC4	$	4	D	T	d	t	„	”	¤	´	Ä	Ô	ä	ô
5	ENQ	NAK	%	5	E	U	e	u	…	•	¥	µ	Å	Õ	å	õ
6	ACK	SYN	&	6	F	V	f	v	†	–	¦	¶	Æ	Ö	æ	ö
7	BEL	ETB	'	7	G	W	g	w	‡	—	§	·	Ç	×	ç	÷
8	BS	CAN	(	8	H	X	h	x	ˆ	˜	¨	¸	È	Ø	è	ø
9	HT	EM	)	9	I	Y	i	y	‰	™	©	¹	É	Ù	é	ù
A	LF	SUB	*	:	J	Z	j	z	Š	š	ª	º	Ê	Ú	ê	ú
B	VT	ESC	+	;	K	[	k	{	‹	›	«	»	Ë	Û	ë	û
C	FF	FS	,	<	L	\	l	|	Œ	œ	¬	¼	Ì	Ü	ì	ü
D	CR	GS	-	=	M	]	m	}	undef	undef	-	½	Í	Ý	í	ý
E	SO	RS	.	>	N	^	n	~	Ž	ž	®	¾	Î	Þ	î	þ
F	SI	US	/	?	O	_	o	DEL	undef	Ÿ	¯	¿	Ï	ß	ï	ÿ

ETSI GSM 03.38 kodetabell for SMS

	00	10	20	30	40	50	60	70
0	@	Δ	space	0	¡	P	¿	p
1	£	_	!	1	A	Q	a	q
2	$	Φ	”	2	B	R	b	r
3	¥	Γ	#	3	C	S	c	s
4	è	Λ	¤	4	D	T	d	t
5	é	Ω	%	5	E	U	e	u
6	ù	Π	&	6	F	V	f	v
7	ì	Ψ	'	7	G	W	g	w
8	ò	Σ	(	8	H	X	h	x
9	Ç	Θ	)	9	I	Y	i	y
A	LF	Ξ	*	:	J	Z	j	z
B	Ø	ESC	+	;	K	Ä	k	ä
C	ø	Æ	,	<	L	Ö	l	ö
D	CR	æ	-	=	M	Ñ	m	ñ
E	Å	undef	.	>	N	Ü	n	ü
F	å	É	/	?	O	$	o	à

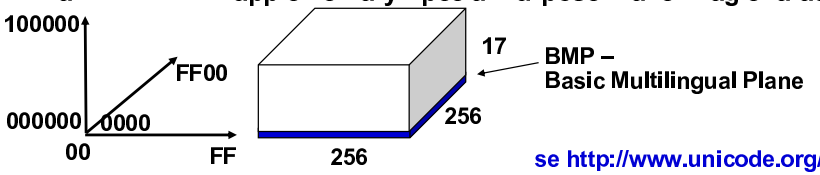
…pluss disse 10 ”escape”-sekvensene

€	ESC e
FF	ESC LF
[	ESC <
\	ESC /
]	ESC >
^	ESC Λ
{	ESC (
|	ESC @
}	ESC)
~	ESC =

Den endelige løsning? – Unicode og ISO 10646

- 21 biter, med mulighet for 1 114 112 bitmønstre
- Tegnsettet er delt opp i 17 plan med 2^{16} = 65536 bitmønstre i hvert plan
- Plan 0: BMP – Basic Multilingual Plane U+0000 to U+FFFF
- Plan 1: SMP – Supplementary Multilingual Plane
  – historiske språk (f. eks. egyptiske hieroglyfer), musikk
- Plan 2: SIP – Supplementary Ideographic Plane
  – sjeldne kinesiske tegn
- Plan 14: SPP – Supplementary Special Purpose Plane – tag characters

I Unicode skriver vi U+ istedenfor 0x



se http://www.unicode.org/

Unicode databasen

- For hvert tegn finnes
  - en representativ glyf
  - kodepunktet
  - et navn
  - klassifisering
  - Skriveretning

se http://www.unicode.org/charts/

- Vedtatte tegn med kodepunkter skal *aldri* endres
- Det er plass til 1 114 112 kodepunkter, herav ca 130 000 private.
  Ca 870 000 kodepunkter er ennå ikke brukt
- I BMP er det plass til 65 536 ulike bitmønstre.
- 21 000 bitmønstre er brukt for kinesiske ideografer,
  2 048 bitmønstre for surrogatpar,
  6 400 bitmønstre er private.
  6 700 bitmønstre er ennå ikke brukt
- Første 128 tegn er identisk med ASCII
- Første 256 tegn identisk med ISO 8859-1

Men hva er nå egentlig et tegn?

Noen eksempler:

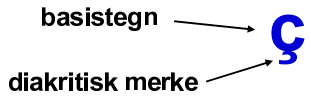
- Norsk: aa (f.eks. i navn) oppfattes som å
- Spansk: ll (eks. Mallorca) oppfattes som j
- Er ö ett tegn, eller o med ”tødler”?
- Er ½ ett tegn, eller tre?
- Ligaturer: ae →æ, ij → ij – to eller ett tegn?
- Er bokstaven Å og enheten Å to forskjellige tegn?
- Kontroversielt tema: Unifisering av CJK (tegnsettet brukt i kinesisk, japansk og koreansk)

Kombinerende tegnsekvenser

- I Unicode finnes mange tegnsekvenser som vises som en eneste, sammensatt glyf.
- Eksempel:
  Et tegn med diakritiske merker (se neste lysark) representeres som et basistegn etterfulgt av ett eller flere kombinasjonstegn som gir de diakritiske merkene
  Eksempel: Ö representeres ved hjelp av de to tegnene O og ¨, dvs. U+4F og U+A8
- For å være kompatibel med ASCII og ISO 8859-1 finnes det også såkalte forhåndssammensatte (”precomposed”) tegn, der basistegn og kombinasjonstegn oppfattes som ett tegn
  Eksempel: Ö kan også representeres direkte som forhåndssammensatt tegn, dvs. U+D6
- Samme tekst kan altså representeres på flere måter!

Eksempler på diakritiske merker

- acute accent – ´
- armstrong or ring above – °
- breve – ˘
- caron/háček – ˇ
- cedilla – ¸
- circumflex – ˆ
- diaresis/umlaut – ¨
- double acute accent – ˝
- grave accent – `
- macron – ¯
- ogonek – ˛



Tegn med endret utseende

- Er ² i teksten x² et spesielt tegn U+B2 = ², eller er det det vanlige tegnet U+32 = 2 som har vært gjenstand for en egnet formatering?
- Er ½ et spesielt tegn U+BD = ½, eller er det de tre tegnene U+31 = 1, U+2F = / og U+32 = 2 som har vært gjenstand for en egnet formatering?

Unicode normalisering

- For å kunne sammenlikne tekster, bør de være normalisert til samme form
- Den vanligste normaliseringsformen (form D):
  - Erstatte forhåndssammensatte tegn med basistegn pluss kombinasjonstegn
  - La tegn med endret utseende stå urørt
- Andre former kan imidlertid være nyttige i spesielle tilfeller

Koding av tegnstrømmer

Hvordan sende og lagre en sekvens av Unicode-tegn?

- Ulike formater, kalt UTF – Unicode Transformation Formats (Unicode) eller UCS – Universal Character Set (ISO 10646)
- UTF-32 = UCS-4:
  Bruker 32 biter for alle tegn (lite brukt).
- UTF-16 = UCS-2:
  - Tegn i BMP: 16 biter.
  - Tegn utenfor BMP: surrogatpar.
- UTF-8 = UCS-1:
  Bruker 8, 16, 24 eller 32 biters avhengig av tegnet.
  - For tegn i ASCII brukes 8-biter med bitmønsteret 0x00 – 0x7F.
  - UTF-8 er identisk med ASCII med en ekstra ledende 0 for tegn definert i ASCII!

Unicode UTF-32

- Føy til ledende 0-biter opp til 32 biter foran representasjonen av kodepunktet

Eksempler:

- Hva er representasjonen for A i UTF-32?

- Hva er representasjonen for Ø i UTF-32?

Unicode UTF-16

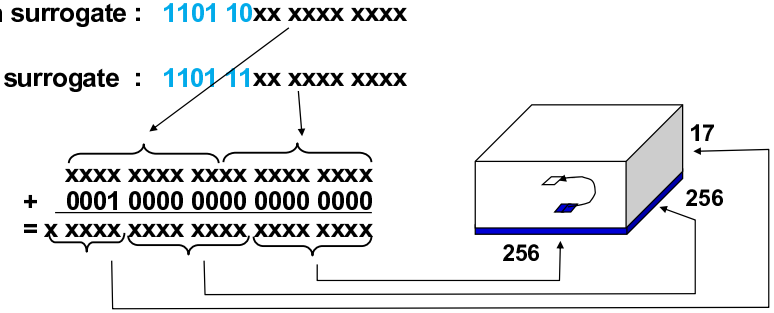
- Med 16 biter kan kodepunktene i BMP representeres direkte
- De 16 planene over BMP adresseres med *surrogatpar* fra BMP
- Surrogatpar
  - Bitmønstre i BMP som fungerer som "halvtegn" av 21-biters tegn
  - Første halvdel (high surrogate): U+D800 – U+DBFF
  - Siste halvdel (low surrogate): U+DC00 – U+DFFF
  - Brukes ikke for noe annet, og tar derfor plass som ellers kunne vært brukt for vanlige tegn i BMP
- Unicode UTF-16 har altså tegn med variabel lengde, men lengden kan fastslås ut fra bitmønsteret

S = (high – 0xD800) * 0x0400 + (low – 0xDC00) + 0x10000

- High surrogate – D800 – DBFF : 1101 10xx xxxx xxxx
- Low surrogate – DC00 – DFFF: 1101 11xx xxxx xxxx
- Gir indirekte tilgang til 2^{20} = 1 048 576 ekstra tegn ut fra BMP

High surrogate : 1101 10xx xxxx xxxx

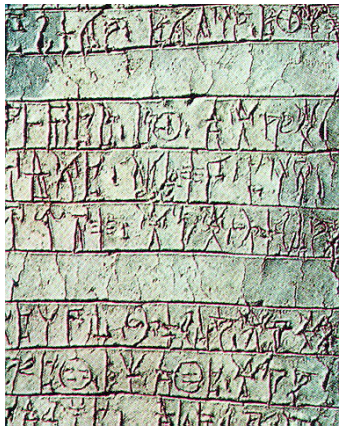
Low surrogate : 1101 11xx xxxx xxxx



Unicode UTF-16 – Surrogat-par - eksempel

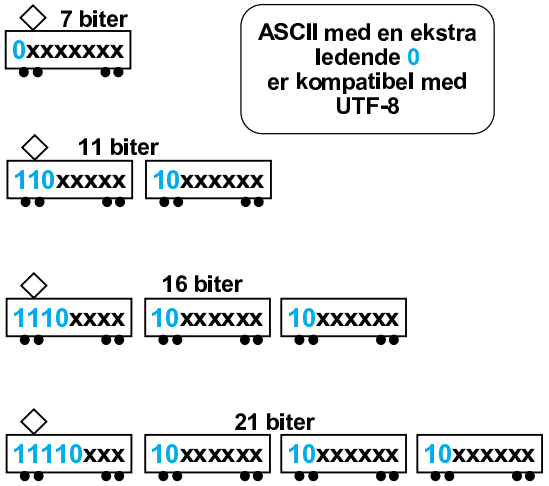
Hva er representasjonen for Linear B ⊕ i Unicode UTF-16?

- Kodepunktet for ⊕ er U+1000F



Unicode UTF-8

- Enslig motorvogn = 0 + ASCII-kode
- Motorvogn i tog begynner alltid med et antall 1er-biter etterfulgt av en 0
- Antall 1er-biter i motorvognen = antall vogner i toget
- Vognene begynner alltid med 10
- Disse bitmønstrene brukes ikke for vanlige tegn i UTF-8



http://www.ifi.unizh.ch/mml/mduerst/papers/PDF/IUC11-UTF-8.pdf

Unicode UTF-8 – eksempler

- Hva er representasjonen for A i Unicode UTF-8?
- Hva er representasjonen for Ø i Unicode UTF-8?

Veletablerte Unicode delmengder

Ambisiøst å skulle håndtere over 40000 tegn!

Vi kommer langt med delmengder:

- WGL4 (Windows Glyph List) – 650 tegn fra MS-DOS, Windows, Mac
- Europeiske UCS-delmengder
  - MES-1: 335 tegn – unionen av ISO 8859-1,2,3,4,9,10,15
  - MES-2: 1052 tegn – supermengde av MES-1 med latin/gresk/kyrillisk/armensk/georgisk pluss matematiske tegn
  - MES-3: 2819 tegn – supermengde av MES-2 og WGL4
- Lignende delmengder for Asia

se www.cl.cam.ac.uk/~mgk25/unicode.html

Big endian vs. Little endian

- I representasjoner som krever mer enn én byte, finnes det to mulige rekkefølger av bytene:
  - Starte med den mest signifikante (”Big endian”)
  - Starte med den minst signifikante (”Little/small endian”)
- Eksempel:
  UTF-16 Big endian for A er 0x 00 41
  UTF-16 Little endian for A er 0x 41 00
- Begge muligheter blir brukt i praksis, og dette kan gi problemer når data overføres fra et maskinmiljø til et annet!

Byte order mark (BOM)

- Et "Byte order mark" (BOM) er tegnet "Zero width no-break space" med kodepunkt U+FEFF i begynnelsen av en Unicode-fil.
- Siden det ikke finnes noe tegn med kodepunkt FFFE, kan BOM brukes til å finne filformatet (UTF-32, UTF-16, UTF-8 og Big eller Small endian) – se tabellen.

Koding	BOM-bitmønster
UTF-32, big-endian	0x 00 00 FE FF
UTF-32, little-endian	0x FF FE 00 00
UTF-16, big-endian	0x FE FF
UTF-16, little-endian	0x FF FE
UTF-8	0x EF BB BF

Plain vs. fancy tekst

- Unicode-standarden omfatter bare "plain text":
  Tekst *uten* typografiske virkemidler som bestemt skrifttype og skriftstørrelse, fet skrift, kursiv, bestemt linjeavstand, innrykk osv.
- "Fancy tekst" er Unicodes betegnelse på tekst *med* slike typografiske virkemidler.
- Disse må legges inn i teksten i form av formatteringskommandoer til fremvisningsenheten.

"Fancy tekst" er temaet for neste forelesning!