COM3110/4115/6115: Text Processing

Text Encoding

Rob Gaizauskas

Department of Computer Science
University of Sheffield

Overview

- Languages and Writing Systems
- Characters vs Glyphs
- The Challenge of Character Encoding
- A Note on Binary/Octal/Hexadecimal Representations
- A (Very) Brief History of Character Encoding

Languages and Writing Systems

- Physical faculties required for spoken language
  - Broca's area, region of brain associated with language
  - vocal apparatus

  arose in homo sapiens between 2 million and 300,000 years ago
- Written language only emerged between 3,500-3,100 BCE in 3 independent centres:
  - amongst the Harappans in the Indus Valley (Indus language)
  - amongst the Sumerians in Mesopotamia (cuneiform)
  - amongst the Egyptians (hieroglyphics)

  though some dispute about whether earlier evidence counts as writing

Languages and Writing Systems (cont)

- Not an accident that the emergence of written language coincides with the stunning acceleration of human culture, science, technology and population size over the last 3000-4000 years.
  - ◇ "Humankind is defined by language; but civilization is defined by writing." (Daniels and Bright, 1996)
- Today some 4000-5000 languages spoken (depending on, e.g. distinction made between language and dialect)
- However, only something like 170-180 writing systems in use + ∼10 undeciphered ones (according to `omniglot.com`)
  - ◇ Not clear how many languages lack writing systems

Languages and Writing Systems (cont)

- What is a writing system (also called a script)?

  "a system of more or less permanent marks used to represent an utterance in such a way that it can be recovered more or less exactly without the intervention of the utterer"

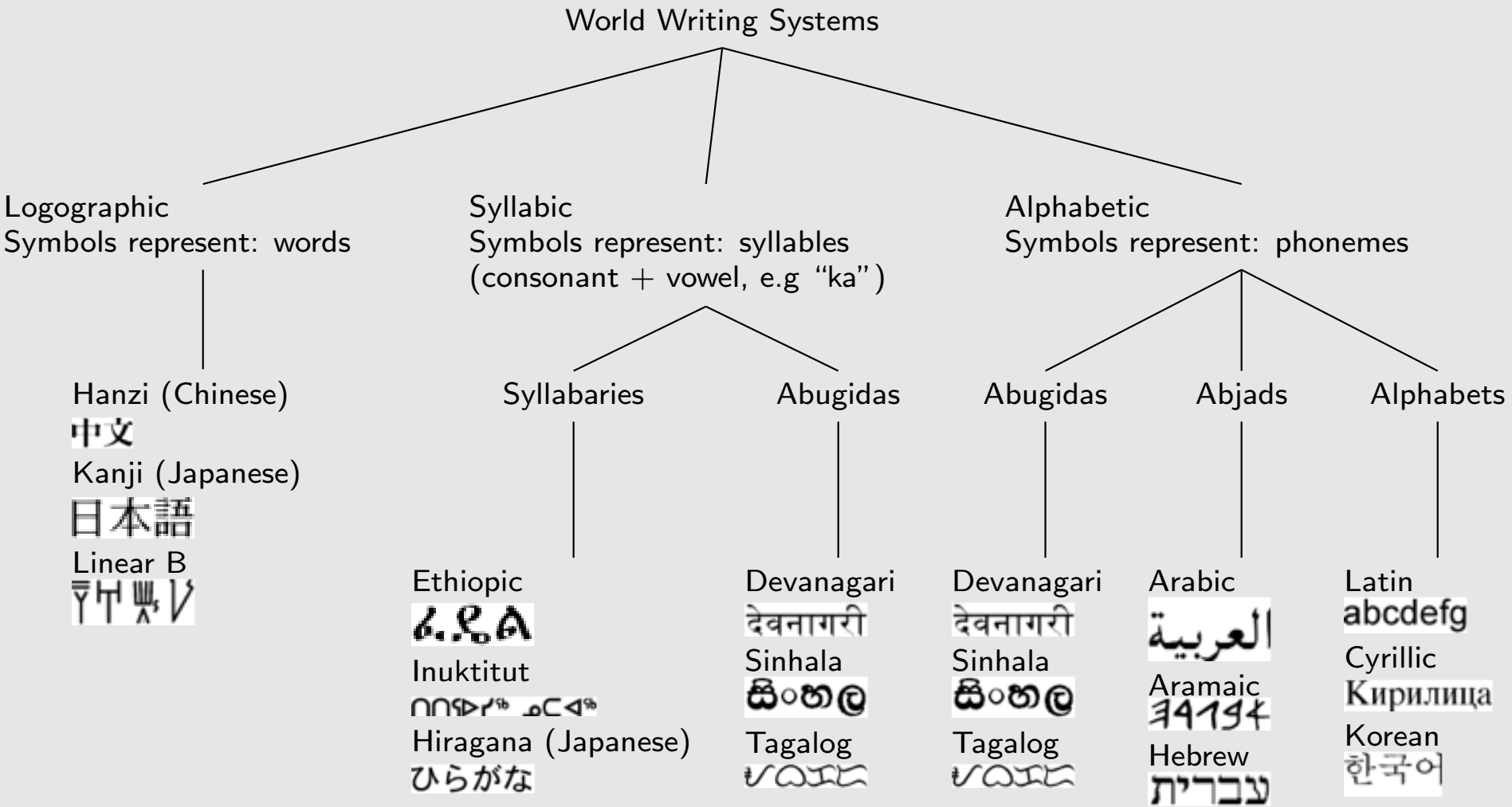
  Peter T. Daniels, *The World's Writing Systems*

- There is a many-to-many mapping between languages and scripts
  - ◇ many languages may share one script
    - e.g. most of the languages of Western Europe use the Roman script with some small variations (accents in French, some extra characters in Spanish, the Scandinavian languages)
  - ◇ some languages may have multiple scripts
    - Japanese – kanji (from Chinese); hiragana (for grammatical particles); katakana (for loan words from languages other than Chinese); arabic numerals

Languages and Writing Systems (cont)

- Writing systems may be classified along a number of dimensions:
  - Directionality
    - left-to-right – e.g. English, Russian
    - right-to-left – e.g. Arabic, Hebrew
    - top-to-bottom – e.g. Chinese
    - bottom-to-top – e.g. Mongolian
    - boustrophedon ("ox-turning") – e.g. (some) ancient Greek
  - Historical derivation
  - Relationship between symbols and sounds
    - *phonological* systems show a clear relationship between sounds of the language and symbols
    - *non-phonological* systems do not
- Sound/symbol relationship generally agreed to be the best way to classify languages
  - but no consensus among scholars as to the best set of sub-classifications

Taxonomy of Writing Systems



Characters vs Glyphs

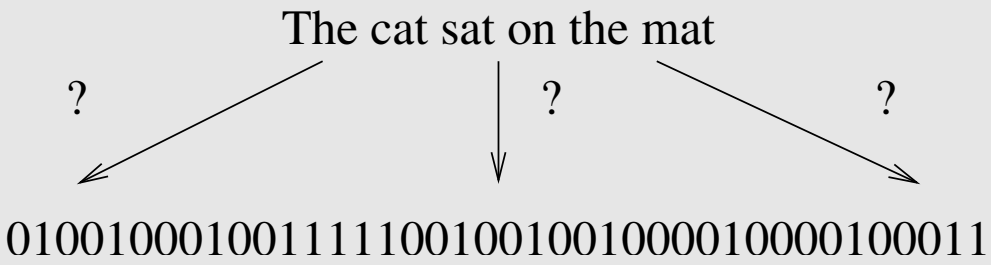
- A character is the smallest component of a writing system that has a semantic value.
  - I.e. changing one character to another – e.g. *bat* → *cat* – changes the meaning of the word in which it occurs
  - The word grapheme (by analogy with phoneme, the smallest sound unit in spoken language) is used
    - sometimes synomously with "character"
    - sometimes to indicate multiple characters that function like a character – e.g. á
- A glyph is a representation of a character or characters, as it/they is/are rendered or displayed.
  - E.g. **A**, A, $\mathcal{A}$ are different glyphs representing the Latin character A
  - A repertoire of glyphs of similar appearance makes up a font
  - Changing font does **not** change the basic meaning of the word(s) whose font is changed
    - Though changing font to italics or bold can change, e.g., emphasis, which is an aspect of meaning

Characters vs Glyphs (Cont)

- The relationship between glyphs and characters can be complex
  - for any character there may be many glyphs representing it (in different fonts)
  - a single glyph may correspond to a number of characters.
    - E.g. "ﬁ" is typically a single glyph used to represent "f" followed by "i" in typesetting English
  - there may be arbitrariness – should é
    - be stored as two characters and rendered using two glyphs?
    - be stored as two characters and rendered by one glyph?
    - be stored as one character and rendered by one glyph?
- When deciding on the set of characters for a language for purposes of representing them in a computer, it is extremely important (e.g. for sorting, indexing purposes) to separate out characters from glyphs
  - the underlying representation of a text should contain the character sequence only
  - the final appearance of the text is the responsibility of the rendering process

The Challenge of Character Encoding

- All data in held in a digital computer is ultimately stored in binary form – i.e. as sequences of 0's and 1's
- Therefore, to store character data, i.e. text, in a digital computer it is necessary to agree an **encoding scheme**



- Various issues arise in designing a character encoding scheme:
  - ◇ Generality How many languages are you aiming to deal with?
  - ◇ Character Set Specification What is the character set(s) to be encoded?
  - ◇ Hardware Issues What are the minimal units of addressable memory?
  - ◇ Variable/Fixed Width Should every character occupy the same number of bits? Or should more frequently occurring characters occupy fewer?
  - ◇ Interoperability Who do you want to play with? Do they already have a different scheme? (**standards**)

A Note on Binary, Octal & Hexadecimal Representations

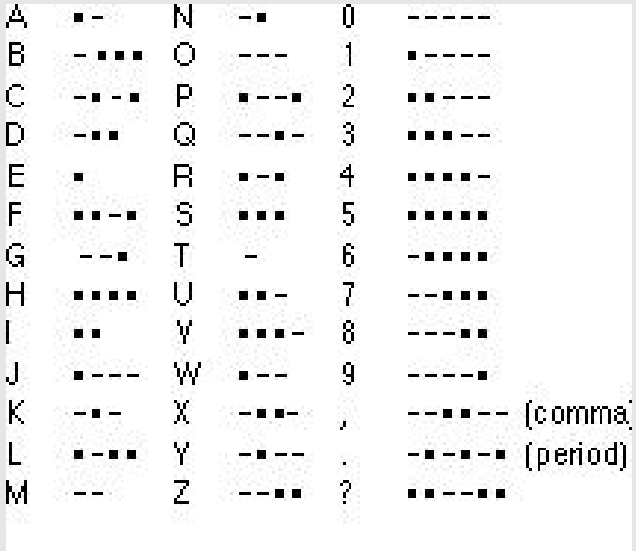
- One byte = 8 bits.
- 8 bits can represent $2^8 = 256$ distinct values – 0 - 255.
- In binary notation, these range from

$$\begin{array}{l} 00000000 \\ 00000001 \\ 00000010 \\ \vdots \\ 11111111 \end{array}$$

- In octal (base 8) notation, they range from: 000 to 377.
  - $o377 = (3 * 64) + (7 * 8) + 7)$.
- In hexadecimal (base 16) notation, they range from 00 to *FF*
  - $xFF = (F * 16) + F$.
- Two bytes = 16 bits, and can represent $2^{16} = 65,536$ distinct values.
- Two byte values are most commonly written in hexadecimal; range from 0000 - *FFFF*
  - $d65535 = xFFFF = (15 * 4096) + (15 * 256) + (15 * 16) + 15$

A (Very) Brief History of Character Encoding

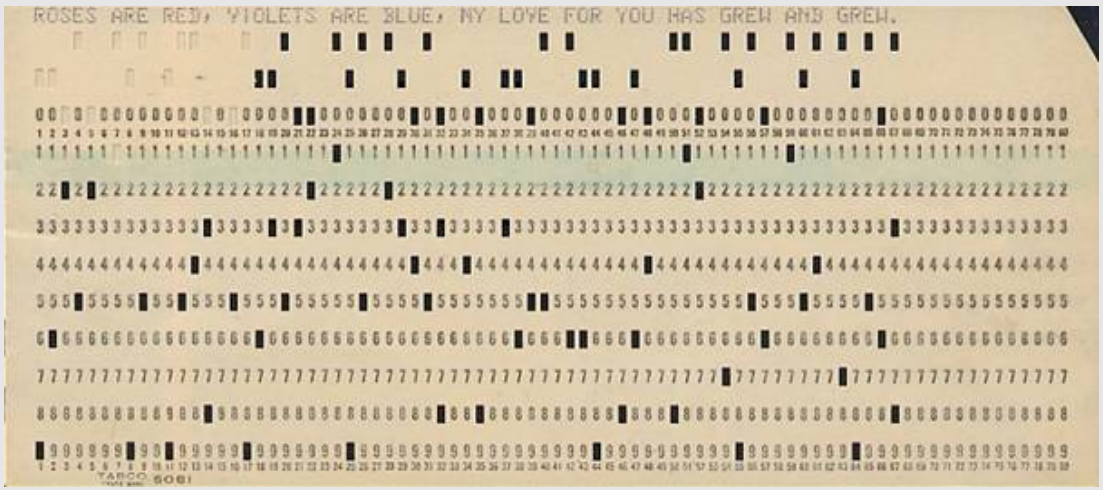
- "Standards are a good thing; and the good thing about standards is that there are so many to chose from ..."
- **Telegraphy and the Morse and Baudot Codes**
  - Telegraph invented by Wheatstone + Cooke (UK), 1837
  - Samuel Morse (US) invents simpler system + code, 1838
    - Variable length code, based on 2 values – dot's or dashes



  - Baudot (Fr) invents printing telegraph ("teleprinter"), 1874
    - used first binary code for textual data – 5 bit Baudot code
    - used a shift code to double code space; i.e., $2 * 2^5$

A (Very) Brief History of Character Encoding (cont)

- **Hollerith and Punched Cards**
  - Herman Hollerith (US) invents punched card tabulating machines to process 1890 US census data
  - code based on holes punched in 12 (row) $\times$ 80 (col) cards
  - Hollerith commercialises invention – Tabulating Machine Co (1896) – becomes International Business Machines Corp (IBM) in 1924



From: `http://wvegter.hivemind.net/abacus/CyberHeroes/Hollerith_files/image006.jpg`

A (Very) Brief History of Character Encoding (cont)

- **ASCII**
  - American Standards Association (ASA – later changed to American National Standards Institute [ANSI]) proposed a 7-bit code – the American Standard Code for Information Interchange (ASCII) (1963, finalised 1968)

ASCII

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	U+0000 NUL 0	U+0001 SOH 1	U+0002 STX 2	U+0003 ETX 3	U+0004 EOT 4	U+0005 ENQ 5	U+0006 ACK 6	U+0007 BEL 7	U+0008 BS 8	U+0009 HT 9	U+000A LF 10	U+000B VT 11	U+000C FF 12	U+000D CR 13	U+000E SO 14	U+000F SI 15
1_	U+0010 DLE 16	U+0011 DC1 17	U+0012 DC2 18	U+0013 DC3 19	U+0014 DC4 20	U+0015 NAK 21	U+0016 SYN 22	U+0017 ETB 23	U+0018 CAN 24	U+0019 EM 25	U+001A SUB 26	U+001B ESC 27	U+001C FS 28	U+001D GS 29	U+001E RS 30	U+001F US 31
2_	U+0020 SP 32	U+0021 ! 33	U+0022 " 34	U+0023 # 35	U+0024 $ 36	U+0025 % 37	U+0026 & 38	U+0027 ' 39	U+0028 (40	U+0029) 41	U+002A * 42	U+002B + 43	U+002C , 44	U+002D - 45	U+002E . 46	U+002F / 47
3_	U+0030 0 48	U+0031 1 49	U+0032 2 50	U+0033 3 51	U+0034 4 52	U+0035 5 53	U+0036 6 54	U+0037 7 55	U+0038 8 56	U+0039 9 57	U+003A : 58	U+003B ; 59	U+003C < 60	U+003D = 61	U+003E > 62	U+003F ? 63
4_	U+0040 @ 64	U+0041 A 65	U+0042 B 66	U+0043 C 67	U+0044 D 68	U+0045 E 69	U+0046 F 70	U+0047 G 71	U+0048 H 72	U+0049 I 73	U+004A J 74	U+004B K 75	U+004C L 76	U+004D M 77	U+004E N 78	U+004F O 79
5_	U+0050 P 80	U+0051 Q 81	U+0052 R 82	U+0053 S 83	U+0054 T 84	U+0055 U 85	U+0056 V 86	U+0057 W 87	U+0058 X 88	U+0059 Y 89	U+005A Z 90	U+005B [91	U+005C \ 92	U+005D] 93	U+005E ^ 94	U+005F _ 95
6_	U+0060 ` 96	U+0061 a 97	U+0062 b 98	U+0063 c 99	U+0064 d 100	U+0065 e 101	U+0066 f 102	U+0067 g 103	U+0068 h 104	U+0069 i 105	U+006A j 106	U+006B k 107	U+006C l 108	U+006D m 109	U+006E n 110	U+006F o 111
7_	U+0070 p 112	U+0071 q 113	U+0072 r 114	U+0073 s 115	U+0074 t 116	U+0075 u 117	U+0076 v 118	U+0077 w 119	U+0078 x 120	U+0079 y 121	U+007A z 122	U+007B { 123	U+007C | 124	U+007D } 125	U+007E ~ 126	U+007F DEL 127

From: `http://www.gammon.com.au/unicode/`

  - ASCII adopted by all US computer manufacturers except market-leaders IBM – proposed their own code – Extended Binary Coded Decimal Interchange Code (EBCDIC)
  - Remains the most widely used coding scheme for English language text

A (Very) Brief History of Character Encoding (cont)

- **ISO/IEC 8859**
  - ASCII insufficient for Western European languages
  - International Standards Organisation proposed a derivative with 10 codes left for national variants – ISO 646 (1967)
  - Has been rationalised and extended to cover most needs of Western and Eastern Europe – ISO 8859 (revised up until 2001)
  - Despite ISO 8859 many American PC companies built OS-specific extensions of ASCII for European languages that were not compatible with ISO 8859
  - Not maintained extended since 2001 – now subsumed by Unicode/UCS
  - See `https://en.wikipedia.org/wiki/ISO/IEC_8859`.
- Paralleling develops in North America and Europe, efforts to develop standardised character codes have gone on in Asia
  - for Chinese, Japanese and Korean (CJK) codes – e.g JIS
  - for South Asian languages – e.g. ISCII

A (Very) Brief History of Character Encoding (cont)

- So far have discussed codes as developed to meet needs of individual countries. But may need to
  - process multiple languages in one document
  - reuse software for products dealing with multiple languages
- By mid-1980s it became clear there was a need for multilingual coding schemes
- Early efforts
  - some pioneering efforts by Xerox (Xerox Character Code Standard – XCCS) and IBM in early 1980's
  - TRON project (tronweb.super-nova.co.jp/homepage.html)
    - ongoing in Japan (U. Tokyo + Japanese industry) since 1984
    - uses escapes to shift between character planes – "limitlessly extensible"
    - built into BTRON operating system that runs on PCs
- By late 1980's two major efforts underway to build character encoding schemes to embrace all the world's languages – past, present and future
  - Unicode started as initiative of US software industry in 1988
  - The Universal Coded Character Set (UCS) aka ISO/IEC 10646 started in 1989 as an initiative of European and Japanese researchers

A (Very) Brief History of Character Encoding (cont)

- Unicode (`www.unicode.org`; `en.wikipedia.org/wiki/Unicode`)
  - committed to 16 bit codes and to avoiding escape sequences
  - supported by, e.g. Apple, IBM, Microsoft, Google, ...
  - now part of XML and HTML standards, Java, Perl, Python, ...
  - current version (Version 10.0.0) contains a repertoire of 136,755 characters covering 139 modern and historic scripts, as well as multiple symbol sets
  - has been criticised in East Asia for not meeting technical needs (character coverage insufficient; no way to expand without moving beyond 16 bit codes)
    - proponents: "the universal character encoding scheme for written characters and text" and "the foundation for global software"
    - opponents: "little more than an exercise in Western cultural imperialism"
- ISO/IEC10646 (`en.wikipedia.org/wiki/Universal_Coded_Character_Set`)
  - initially (1990) proposed a scheme involving 128 groups of 256 planes of 256 rows of 256 cells – could encode 679,477,248 characters
  - Supported four byte, two byte and variable length (1-5 byte) encodings
  - Was opposed by Unicode as too complex/requiring too much storage – wanted a two byte scheme
  - Later Unicode was forced to admit 16 bytes was insufficient and added "surrogate pairs" (see below)
- Unicode and UCS have now converged, are maintained together, and are code-for-code identical

References

- Crystal, D. The Cambridge Encyclopedia of Language. Cambridge University Press, 1987.
- Daniels, P. T. and Bright, W. eds, The World's Writing Systems. Oxford University Press, 1996.
- Steven J. Searle. "A Brief History of Character Codes in North America, Europe and Asia". Available at: `http://tronweb.super-nova.co.jp/characcodehist.html`. Site last visited 06/11/17.
- Unicode Consortium. The Unicode Standard Version 11.0.0. Available at `www.unicode.org`. Site last visited 23/10/18.
- Unicode Technical Report # 17: Character Encoding Model. Available at `www.unicode.org`. Site last visited 06/11/17.