Systemy pisma i problemy ich komputerowego przetwarzania

Unicode

Jolanta Bachan

Prezentacja oparta na slajdach Kevina Scannella „*Unicode for Linguists*”, INNET Summer School, Gniezno, Polska, 4 września 2013

Unicode

Unicode – komputerowy zestaw znaków mający w zamierzeniu obejmować wszystkie pisma używane na świecie. Definiują go dwa standardy – Unicode oraz ISO 10646: *Universal Coded Character Set (UCS)*. Znaki obu standardów są identyczne.

Wikipedia

Podstawy kodowania

- Kodowanie: mapowanie znaków ↔ liczby
- Komputery dobrze przechowują liczby!
- Kodowaniom nie zależy na tym, jak wstawiamy znaki
- Kodowania nie są związane z prezentacją znaków
- Nie ma czcionek, formatowania, pogrubień – tylko tekst!
- "Modularność" – trzy odrębne warstwy
  - glyf(kształt)+znak(nazwa)+punkt kodowy(liczba)
- Wcześniej było zamieszanie (klawiatury i czcionki połączone)

~~Czcionki~~

- Poprawnie zakodowany tekst jest odrębnym bytem
- Oczywiście, czcionki umożliwiają prezentację tekstu na ekranie
- Jest wiele czcionek, które zawierają wiele znaków
- Ale nadal zdarzają się wyjątki:

  I I I I I I I I I I I I I I I

- A niektóre czcionki mają błędy...
- *Szczególnie jeśli zawierają* znaki diakrytyczne i ligatury

Zalety zwykłego tekstu

- Unicode koduje "zwykły tekst" (i nic więcej)
- Zwykły tekst jest portatywny pomiędzy różnymi platformami
- Zwykły tekst jest długowieczny
- Zwykły tekst to najlepszy otwarty format
- Nie są potrzebne żadne narzędzia zastrzeżone do manipulowania tekstem
- Zwykły tekst jest łatwo czytelny i edytowalny
- Pliki HTML czy XML to też zwykły tekst

Kodowanie: Trochę historii

- ASCII (1963); A-Za-z, 0-9, itd., kodowanie 0-127
- EBCDIC (1964); mapowanie do 0-255 (z brakami)
- serie ISO 8859 (1987-); 0-255, rozwinięcie ASCII
- Big5 (1984); tradycyjny Chiński
- Shift JIS; Japoński
- GB2312/18030; uproszczony Chiński
- Tysiące innych
- Dlaczego jest aż tak źle?

Alfabetyczna zupa

IBM1124, CP424, ISO_10367-BOX, ECMA-128, ISO8859-1, KOI-7, CSIBM1137, EBCDIC-GREEK, ISO-IR-99, GOST_1976874, CSISOLATINCYRILLIC, CP901, IBM933, IBM-1142, CP1026, IBM4909, UTF32BE, CSISO19LATINGREEK, CSIBM4971,MSGREEK, MAC, ISO_8859-10:1992, TS-5881, WINBALTRIM, IBM-1161, NF_Z_62-010, CSIBM420, CSISO89ASMO449, CP1144,ISO_6937-2, CP-AR, CP281, T.61-8BIT, EUCTW, CSIBM500, ISO-IR-141, INIS, CP10007, IBM16804, PT2, ISO-8859-13, IBM12712,IBM-1162, CSISO86HUNGARIAN, SHIFT_JIS, CUBA, ISO-IR-138, EBCDIC-ES-S, ISO-IR-4, ISO_8859-2:1987, OSF10020359,EBCDIC-CP-FI, CP1251, ISO_8859-4:1988, ISO-IR-103, GB18030, EBCDICATDE, OSF00010020, CSPC862LATINHEBREW,ISO_8859-14, 8859_5, CSISO5427CYRILLIC, CSA_T500-1983, EBCDIC-CP-WT, EBCDIC-IS-FRISS, CSIBM1364, CP274,ISO885914, EBCDIC-CP-AR1, IBM1132, CSIBM863, CSIBM1123, BIG-FIVE, ISO8859-2, 904, IBM1129, ISO-8859-11, SE,CSIBM1026, CP285, UTF16LE, IBM1141, CSIBM297, CP1097, IBM856, IBM277, 1046, NF_Z_62-010_1973, HPTHAI8,CSISO15ITALIAN, CP1141, IBM-1147, CSIBM902, IBM1026, CP819, ISO-8859-9E, CSIBM1129, EBCDICDKNO, CP937, WINDOWS-31J, ARMSCII-8, EBCDIC-AT-DE, ISO_8859-7:1987, IBM9448, CP4909, GB_1988-80, CP278, DECMCS, IBM273, 8859_7,NC_NC00-10:81, EBCDIC-CP-SE, IBM1160, CSISOLATIN2, IBM-1166, OSF10020365, IBM1388, OSF10020111, RK1048, ISO88598,CSISO150GREEKCCITT, CP9448, ISO-IR-69, BIG5, IBM904, ASMO_449, CSISOLATIN4, MACUK, CSISO122CANADIAN2,CSIBM1148, ISO2022CN, IBM866NAV, CSIBM903, ASMO-708, CP1004, IBM1158, CP297, CSISO141JUSIB1002, CP437, MS-EE,CP771, CP1255, IBM1143, CP772, DEC, OSF1002035D, DS2089, MSCP949, ISO-2022-JP-2, TIS-620, ISO88592,CSISO27LATINGREEK1, CP1161, DE, 855, ISO-2022-JP-3, CP1256, CSIBM11621162, CP1390, MS-TURK, NATSDANO, CP500,1026, IBM1137, IBM284, ISO885916, OSF10020352, CSIBM1390, IBM1163, ES2, ISO_8859-5, CP1160, HPGREEK8, IBM-933,MACINTOSH, UCS4, CP891, LATIN3, CSISO69FRENCH, CP949, IBM-1124, EBCDIC-AT-DE-A, CSISOLATIN5, NAPLPS, IBM868,EUCCN, INIS-8, CSIBM939, ISO885913, CSIBM860, ISO-IR-86, NF_Z_62-010_(1973), ISO8859-7, ANSI_X3.110, ISO-IR-89,CP1364, ISO_8859-9, JIS_C6229-1984-B, ISO_8859-3:1988, CP903, MAC-UK, 437, CSIBM866, WINDOWS-1258, ISO646-CA2,CP939, OSF10020354, KOI8R, CSA7-1, IBM-1122, CP4517, IBM855, ISO_6937-2:1983, GEORGIAN-PS, CSIBM935, UCS-2LE,IBM-16804, CP1081, IBM-1163, CP1124, LATIN10, WINDOWS-1257, CP874, IBM916, ISO_9036, CSIBM803, ISO8859-5, IBM-1046,CSIBM1097, EBCDIC-CP-CA, ISO_69372, CSISO60DANISHNORWEGIAN, OSF10020115, CSEBCDICFISE, EBCDIC-JP-KANA,CP1282, CSIBM1112, IBM874, IBM-1143, BALTIC, CSIBM1025, INIS8, EBCDIC-CP-NO, CP902, BIG-5, CP1254, CSIBM855,EBCDICESA, JP-OCR-B, TCVN5712-1:1993, ISO646-FR1, TSCII, IBM-9066, CSIBM1132, ISO2022JP2, EBCDICDKNOA, IBM1164,IBM4517, UTF-16BE, CP1025, ISO-IR-111, IBM-4971, SJIS-WIN, MAC-CENTRALEUROPE, CP1137, IBM-1008, CSMACINTOSH,EBCDIC-CP-CH, CP905, CP273, VISCII, OSF00010008, ROMAN9, IBM-9448, OSF0001000A, 864, ISO_8859-10, IBM857, LATIN9,OSF10020357, ISO-IR-37, CSIBM273, EBCDIC-CP-HE, LATIN6, ISO88591, CSIBM1164, TIS620.2529-1, CSISO143IECP271, ISO-8859-9, CSIBM1399, ...

Ogromna niejednoznaczność

- Znak á odpowiada 225 w ISO-8859-1
- Ale 225 to ß w kodowaniu CP770
- I w cyrylicy ц (c) w CP771
- A w arabskim lam (ل) w CSIBM9448
- I ţ (małe i z ogonkiem) w ISO-8859-13
- Itd...

Wielojęzyczność

- Żadne kodowanie nie pokryło wielu systemów pisma
- Wielojęzyczne dokumenty były niemożliwe do stworzenia
- Zobacz plik `ranodomsample.txt`
- Jedno losowo wybrane zdanie w 1500+ językach
- Całkowicie niemożliwe 25 lat temu!

Luki

- Wiele systemów pisma nie miało kodowania wcale
- Ludzie wymyślali kodowania "ad hoc"
- Przeważnie brakowało tylko kilku znaków
- np. walijski korzysta z ISO 8859-1 + ŵ, ŷ
- Rozwiązanie: stwórz nowe kodowanie (ISO 8859-14)
- Mongolski korzysta z cyrylicy + ө, ҮҮ
- Rozwiązanie: "triki czcionkowe"
  - przerysuj є, ї jako ө, ґ
- Albo czcionki, które przerysowują *wszystko* (np. język czirokeski - ᏣᎳᎩ – *Tsalagi*)

Unicode!

- Pierwsze starania, aby stworzono uniwersalne kodowanie, późne lata 80'.
- Konsorcjum Unicode, ISO, 1991-1992
- Pierwsza wersja 1.0.0 w październiku 1991, 7161 znaków
  - Wersja 7.0 – czerwiec 2014, 113.021 znaków
  - Wersja 8.0 – czerwiec 2015, 120.672 znaków
  - Wersja 9.0 – czerwiec 2016,128,172 znaków
  - Wersja 10.0 – czerwiec 2017, 136,690 znaków
  - Wersja 11.0 – czerwiec 2018, 137,374 znaków
  - Wersja 12.1 – maj 2019, 137,929 znaków
- Każdy znak ma przypisany numer
- Te znaki nazywają się "punktami kodowymi"

Unicode!

Obejrzyjcie Unicode na stronie:
http://www.unicode.org/charts/

Niejednoznaczność wyjaśniona

- Znak á jest mapowany do punktu kodowego 225
- ß odpowiada punktowi kodowemu 223
- ц (c) mapuje do 1089
- Arabskie Lam (ل) mapuje do 1604
- ị (małe i z ogonkiem) to 303

System binarny i szesnastkowy

- Zatem á, ale widzimy U+00E1
- Komputery przechowują dane w systemie binarnym o podstawie 2.
- 225=128+64+32+1=11100001
- Dwójkowy system liczbowy
- Szesnastkowy system liczbowy

binarny vs. szesnastkowy

- 0 = 0000
- 1 = 0001
- 2 = 0010
- 3 = 0011
- 4 = 0100
- 5 = 0101
- 6 = 0110
- 7 = 0111
- 8 = 1000
- 9 = 1001
- 10 = 1010 = “A”
- 11 = 1011 = “B”
- 12 = 1100 = “C”
- 13 = 1101 = “D”
- 14 = 1110 = “E”
- 15 = 1111 = “F”

Ćwiczenie (1)

- Co to za liczba w systemie dziesiętnym?
  - 1011
  - 10010
  - 1010011
  - 21
  - 34
  - A2
  - C30

Ćwiczenie (1)

- Co to za liczba w systemie dziesiętnym?
  - 1011 = 8 + 0 + 2 + 1 = 11
  - 10010 = 16 + 0 + 0 + 2 + 0 = 18
  - 1010011 = 64 + 0 + 16 + 0 + 0 + 2 + 1 = 83
  - 21 = $2 \times 16^1 + 1 \times 16^0 = 32 + 1 = 33$
  - 34 = $3 \times 16^1 + 4 \times 16^0 = 48 + 4 = 52$
  - A2 = $10 \times 16^1 + 2 \times 16^0 = 160 + 2 = 162$
  - C30 = $12 \times 16^2 + 3 \times 16^1 + 0 \times 16^0 = 3072 + 48 + 0 = 3120$

Ćwiczenie (2)

- Wybierz 3 znaki ze znaków specjalnych i podaj ich punkty kodowe w systemie dziesiętnym.

Przelicznik

Kolejna niejednoznaczność

- Różne sposoby reprezentacji znaku
- Å =U+212B=ANGSTROM SIGN
- Å = U+00C5=LATIN CAPITAL LETTER A WITH RING ABOVE
- A can also be encoded as U+0041 followed by U+030A
- LATIN CAPITAL LETTER A, COMBINING RING ABOVE
- U+00C5: “forma złożona”, U+030A: “znak diakrytyczny łączący”
- Może to powodować problemy z wyszukiwaniem, tworzeniem list frekwencyjnych, itd.
- Takie znaki należy znormalizować

Demo

- https://cs.slu.edu/~scannell/pub/innet.html
- To są znaki U+212B, U+00C5, U+0041, U+030A
- Skopiuj pierwszy znak, następnie szukaj go (Ctrl+F)
- Powtórz, kopiując znak drugi i trzeci
- Teraz zmień przeglądarkę i spróbuj jeszcze raz!

Rodzaje znaków

- Litery: a,A,A,d,ʰu,ǂ,∫,Σ,ম,, ᥰ ,𝐲,Ӟ, め , 儆
- Liczby: 0,1,2,৫,৬,৩,৪,২,⅙,Ⅶ,⑤
- Interpunkcja: !,?,□,ᨷ,ᨶ, X,≺
- Symbole: £ ,},c,⇏,∇,⌟,♻,✓,
- Znaki diakrytyczne (łączące)

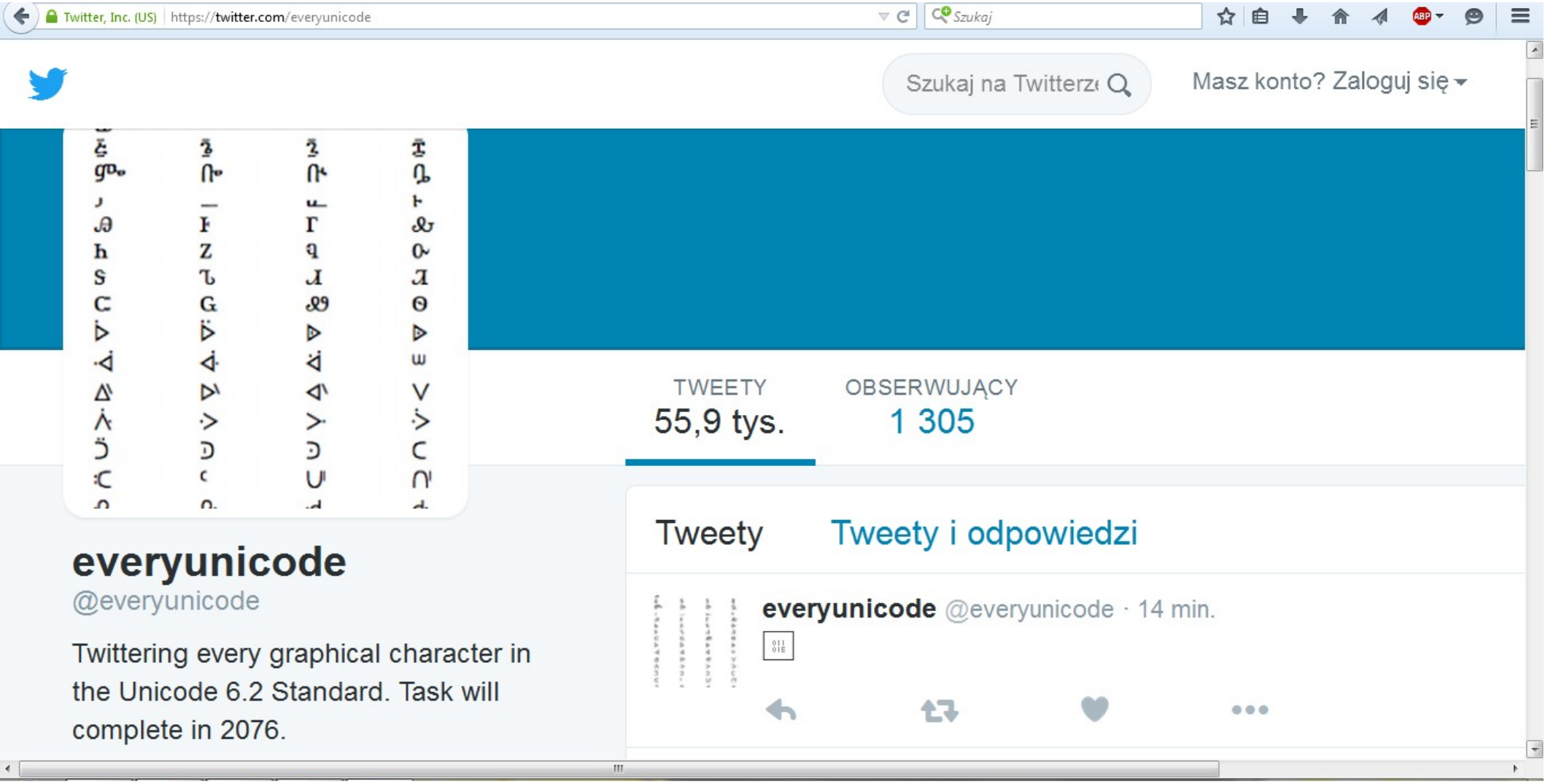
Jeśli czegoś brakuje?

- http://scriptsource.org/
- SIL International: http://www.sil.org/about
- 269 systemy pisma
- 163 mają standard ISO
- wiele języków jeszcze czeka...
- http://www.linguistics.berkeley.edu/sei/
  - http://www.linguistics.berkeley.edu/sei/scripts-not-encoded.html
- Polski:http://scriptsource.org/cms/scripts/page.php?item_id=language_detail&key=pol

Zamieszanie!

- http://www.unicode.org/Public/security/revision-06/confusables.txt
  - SL - Single-Script, Lowercase Confusables
  - SA - Single-Script, Anycase Confusables
  - ML - Mixed-Script, Lowercase Confusables
  - MA - Mixed-Script, Any-Case Confusables
- A = U+0041 = LATIN CAPITAL LETTER A
- A = U+0391 = GREEK CAPITAL LETTER ALPHA
- A = U+0410 = CYRILLIC CAPITAL LETTER A
- ᐃ = U+15C5 = CANADIAN SYLLABICS CARRIER GHO
- Ꭰ = U+13AA = CHEROKEE LETTER GO
- ...
- Jak znaleźć i naprawić nieprawidłowe znaki w istniejącym tekście?
- Jak dokonać prawidłowego wyboru? ɨ → i̠ (U+0268 vs. U+0069, U+0335)
- Odpowiedź: Znormalizuj tekst i znajdź odpowiednie znaki!

https://twitter.com/everyunicode



Ćwiczenie na rozluźnienie

- Przetestujcie tę aplikację:
  http://shapecatcher.com/

Do zobaczenia za tydzień!