InformaticaUmanistica

Lezione 5
La codifica dei caratteri

Claudio Gennaro

ISTI - CNR

UNIVERSITÀ DI PISA

Sommario

- **Introduzione (terminologia)**
- **Standards**
  - ASCII (e varianti)
  - La famiglia ISO 8859
  - ISO/IEC 10646, UCS e Unicode
- **Cenni sui Glifi e Fonts**

Teminologia

- **Nel mondo delle codifiche dei caratteri esistono tre concetti completamente separati:**
  - Repertorio di caratteri
  - Codice
  - Codifiche di caratteri

Repertorio di caratteri

- **Per repertorio di caratteri si intende un insieme di caratteri.**
- **Esso può essere determinato in base alla lingua che rappresenta o essere indipendente dalla lingua.**
- **L'alfabeto latino è un buon esempio di repertorio indipendente dalla lingua che rappresenta.**

Codici dei caratteri

- I codici dei caratteri esprimono una corrispondenza uno-a-uno (spesso rappresentata in forma tabulare) tra i caratteri del repertorio interi non-negativi.
- In pratica ad ogni elemento del repertorio di caratteri viene associato un codice numerico chiamato *code position*.

La codifica dei caratteri

- La codifica è il modo in cui il code position viene ridotto a bit ossia in una sequenza di bytes (algoritmo di mappatura tra code position e una o più sequenze di bit).
- Nel caso più semplice ogni ad carattere può essere associato un intero nell'intervallo 0 - 255 (vale a dire un byte), ma questo funziona solo quando il repertorio ha al più 256 caratteri.
- Per insiemi più grandi bisogna inventarsi codifiche più complesse.

Gli Standards

ASCII

- L'ASCII (American Standard Code for Information Interchange) è sicuramente la codifica più nota nel mondo dell'informatica.
- È uno standard ANSI (X3.4 - 1968) che definisce valori per 128 caratteri, ovvero 7 bit su 8. Nello standard originale il primo bit non è significativo ed è pensato come bit di parità.
- In ASCII si riferisce contemporaneamente, ad un repertorio di caratteri, e alla loro codifica.

Curiosità: le persone di lingua anglosassone pronunciano la parola ASCII come "ASCHI"

ASCII

- **ASCII possiede 33 caratteri (0-31 e 127) di controllo, tra cui molte ripetizioni inutili. Ad esempio:**
  - Backspace (sposta la testina indietro di un carattere, utile nelle telescriventi - 08 [0x08]) e
  - Delete (cancella tutti i buchi di un carattere in una scheda perforata, cioè tutti buchi, 1111111 - 127 [0x7F]).
  - Carriage Return (riporta la testina all'inizio di riga - 13 [0x0C]) e
  - Form Feed (gira il carrello di una riga - 14 [0x0D]) che causano molte confusioni nei sistemi moderni.

ASCII

- **Gli altri 95, composti da caratteri dell'alfabeto latino, maiuscole e minuscole, numeri e punteggiatura sono qui di seguito rappresentati (il primo e l'ultimo sono spazi bianchi):**

```
  ! " # $ % & ' ( ) * + , - . /
0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z [ \ ] ^ _
` a b c d e f g h i j k l m n o
p q r s t u v w x y z { | } ~
```

Variazioni dell'ASCII

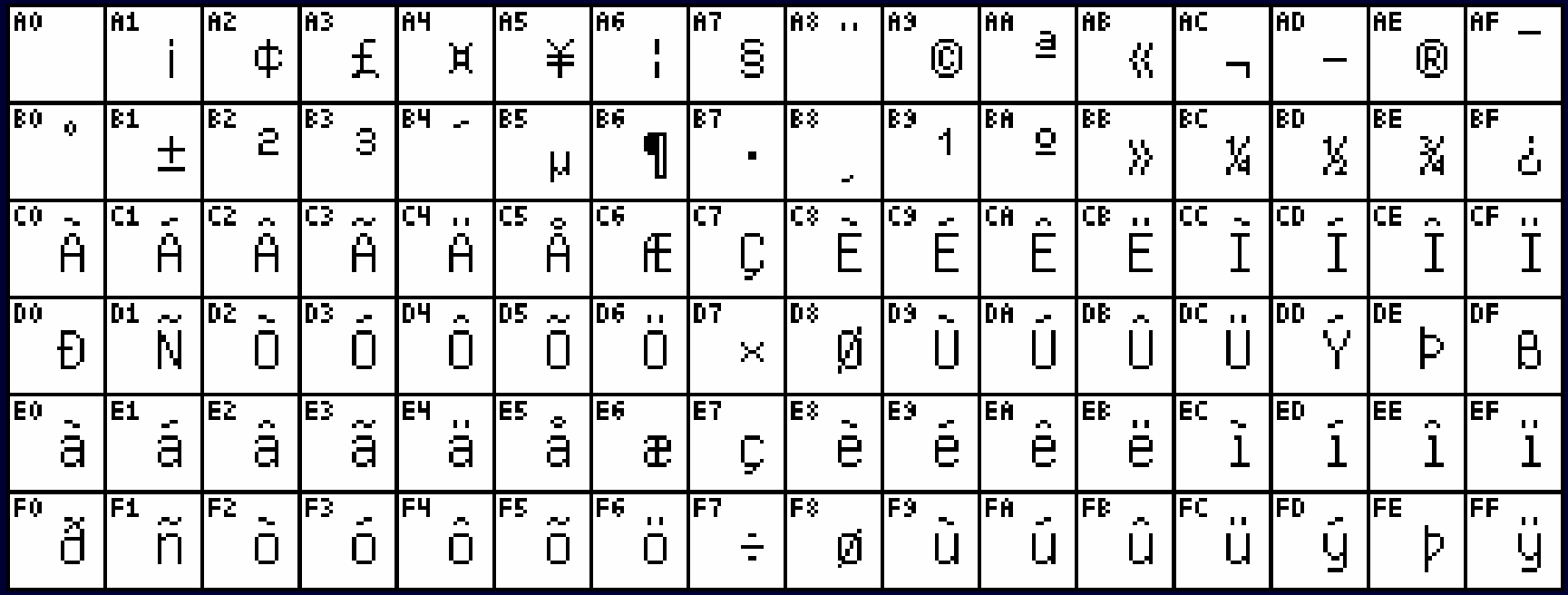
- Esistono molte variazioni internazionali dell'ASCII.
- In queste variazioni alcuni caratteri sono sostituiti da simboli speciali propri della nazione che ha creato la variante.
- Quindi lo standard ASCII (X3.4 - 1968) viene spesso indicato chiamato **US-ASCII**.
- Qualche volta si parla dell'ASCII a "8 bit" questo nome è utilizzato per indicare varie codifiche che sono estensioni dell'ASCII nel senso che contengono ASCII come sottoinsieme ma utilizzano l'intervallo 128-255 per estendere l'insieme.
- Ma in generale il vero ASCII è quello a 7 bit dello standard X3.4 - 1968.

ISO Latin 1

- Una estensione dell'ASCII standard che comprende alcuni caratteri un certo numero di caratteri degli alfabeti europei come accenti, ecc. è l'ISO 8859-1 (della famiglia ISO 8859) che comprende il repertorio di caratteri "Latin alphabet No. 1", noto come ISO Latin 1.
- Questo standard è usato automaticamente da HTTP e qualche sistema operativo.
- Ovviamente ISO Latin 1 è compatibile all'indietro con ASCII, di cui è un estensione per i soli caratteri >127.

ISO Latin 1

- **I caratteri dell'ISO Latin 1 che occupano le posizioni da 160 a 255 sono qui di seguito rappresentati:**



- **Si noti come alcuni caratteri potrebbero non apparire se il font utilizzato non supporta tutto il l'insieme dei caratteri.**

Windows code page 1252

- In ISO Latin 1 le posizioni dal 128-159 sono esplicitamente riservate per caratteri di controllo, e quindi non corrispondono a caratteri rappresentabili graficamente.
- I cosidetto Windows character set (WinLatin1 o Windows code page 1252 per l'esattezza) utilizza questo intervallo per inserire caratteri stampabili come il simbolo dell'euro il del copyright, ecc.
- Comunque questo set non è identico allo standard ISO Latin 1 ed è spesso chiamato *ANSI character set* dalla Microsoft anche se non è mai stato approvato come standard ma semplicemente perché la Microsoft si era basata su una bozza di uno standard ANSI.

Windows code page 1252

Differenza con ISO-Latin 1

8x	€ 128	129	‚ 130	ƒ 131	„ 132	… 133	† 134	‡ 135	ˆ 136	‰ 137	Š 138	‹ 139	Œ 140	141	Ž 142	143
9x	144	' 145	' 146	" 147	" 148	• 149	– 150	— 151	˜ 152	™ 153	š 154	› 155	œ 156	157	ž 158	Ÿ 159
Ax	NBSP 160	¡ 161	¢ 162	£ 163	¤ 164	¥ 165	¦ 166	§ 167	¨ 168	© 169	ª 170	« 171	¬ 173	SHY 173	® 174	¯ 175
Bx	° 176	± 177	² 178	³ 179	´ 180	µ 181	¶ 182	· 183	¸ 184	¹ 185	º 186	» 187	¼ 188	½ 189	¾ 190	¿ 191
Cx	À 192	Á 193	Â 194	Ã 195	Ä 196	Å 197	Æ 198	Ç 199	È 200	É 201	Ê 202	Ë 203	Ì 204	Í 205	Î 206	Ï 207
Dx	Ð 208	Ñ 209	Ò 210	Ó 211	Ô 212	Õ 213	Ö 214	× 215	Ø 216	Ù 217	Ú 218	Û 219	Ü 220	Ý 221	Þ 222	ß 223
Ex	à 224	á 225	â 226	ã 227	ä 228	å 229	æ 230	ç 231	è 232	é 233	ê 234	ë 235	ì 236	í 237	î 238	ï 239
Fx	ð 240	ñ 241	ò 242	ó 243	ô 244	õ 245	ö 246	÷ 247	ø 248	ù 249	ú 250	û 251	ü 252	ý 253	þ 254	ÿ 255

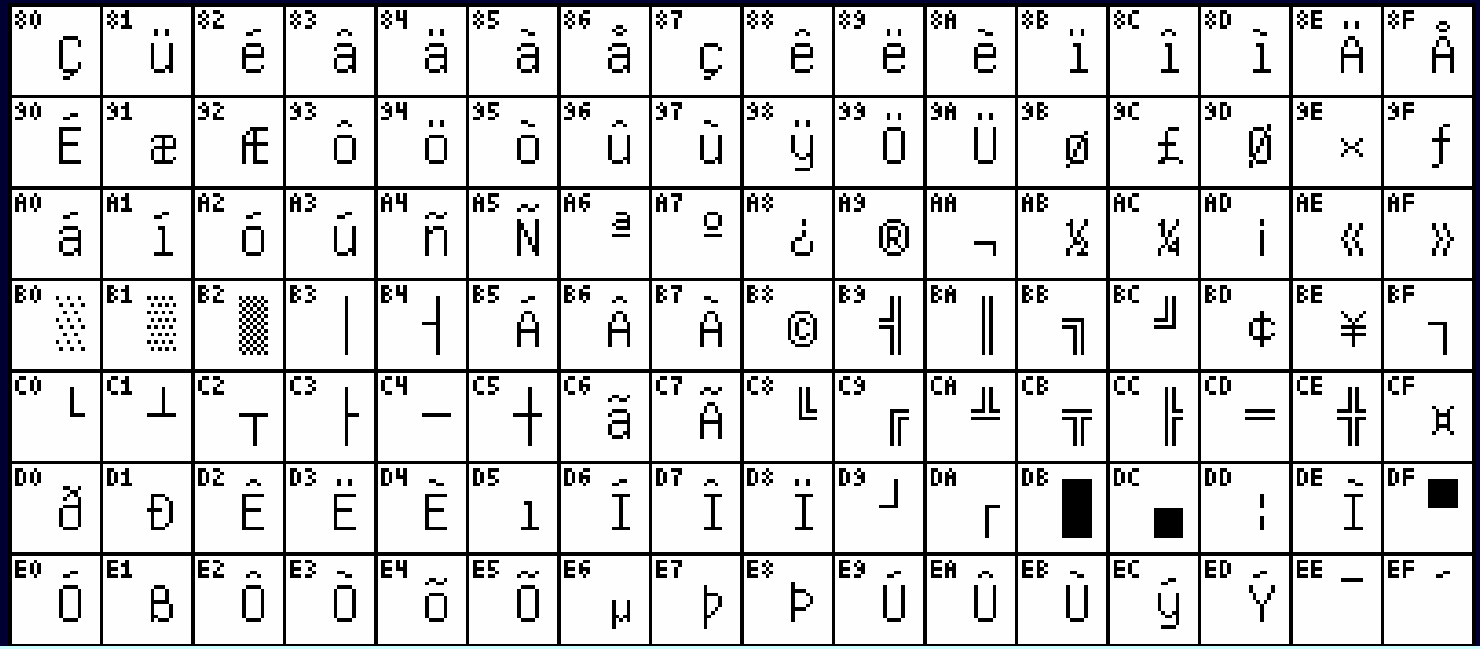
DOS character codes & character code del Macintosh

- Altre estensioni dell'ASCII sono che utilizzano l'intervallo 0-255 (8 bit) sono il DOS character codes, chiamato anche code pages (CP) e il character code del Macintosh.
- Il primo caso si riferisce al sistema operativo DOS che permetteva di installare differenti set di caratteri (code pages appunto), come ad esempio il CP 437 che introduce i caratteri della lingua Greca, simboli matematici, e simboli pseudo-grafici.
- Più tardi divenne famoso il CP 850 che contiene le lettere per le lingue dell'Europa occidentale, praticamente le stesse contenute in ISO Latin 1 ma in posizioni differenti.

DOS character codes & character code del Macintosh

- Si noti che le code pages del DOS sono abbastanza differenti da le Windows character set sebbene si utilizzi spesso nomi come cp-1252 (che somiglia al windows 1252!!!).
- A complicare le cose la Microsoft preferisce usare il nome “OEM code page” per i set di caratteri di una particolare regionalizzazione.
- Per quanto riguarda la codifica del Macintosh è molto più uniforme di quello del mondo dei PC (benché anche in questo caso esistano delle regionalizzazioni). Questo insieme contiene un mix di ASCII, lettere accentate, simboli matematici, ed altri caratteri.

DOS CodePage 850



Curiosità: in windows, aprendo un editore come il notepad o una finestra DOS e componendo la sequenza ALT-xxx (dove xxx sono tre cifre del tastierino), si ottiene la battitura di un carettere corrispondente al CP 850 (e non quello di windows 1252!!!)

La famiglia ISO 8859

- La famiglia ISO 8859 estende il repertorio di caratteri dell'ASCII introducendo caratteri speciali a seconda delle lingua usata.
- Come già detto a questa famiglia appartiene l'ISO 8859-1 (ISO Latin 1) che contiene i caratteri principali delle lingue occidentali con alfabeti latini, ed è usato da molte applicazioni su Internet.
- Ad esempio ISO 8859-2 (ISO Latin 2) contiene i caratteri per le lingue l'Europa Centrale e dell'Est (come Ungherese, Ceco, Polacco, etc.).

La famiglia ISO 8859

- **La seguente tabella riassume la famiglia ISO 8859.**

iso-8859-1, Latin1, L1	Europa dell'Ovest
iso-8859-2, latin2, L2	Europa centrale e dell'Est
iso-8859-3, latin3, L3	Esperanto, Maltese
iso-8859-4, latin4, L4	Lingue baltiche
iso-8859-5, cyrillic	Bulgaro, Bielorusso, Macedone, Russo, Serbo
iso-8859-6, arabic	Arabo
iso-8859-7, greek, greek8	Greco
iso-8859-8, hebrew	Ebraico
iso-8859-9, latin5, L5	Turco
iso-8859-10, latin6, L6	Lingue nordiche
iso-8859-13	Lingue baltiche
iso-8859-14, latin8, L8	Lingue Celte
iso-8859-15	Europa dell'Ovest (Simbolo dell'Euro €)

ISO/IEC 10646, UCS e Unicode

- Negli anni '90 sono state avviate due iniziative parallele per sistemare in modo definitivo il problema della rappresentazione dei caratteri:
- la prima gestita dal consorzio Unicode (una organizzazione no-profit in cui convergono numerosi produttori di sistemi informatici), e la seconda dall'ISO.
- Da queste iniziative sono nati dal consorzio Unicode e ISO 10646/UCS, due codifiche di caratteri che sono per fortuna perfettamente allineati (le differenze riguardano solo aspetti tecnici) e che, con qualche ragione, hanno la presunzione di definirsi 'universali'. UCS sta infatti per “Universal Character Set” e corrisponde ad un vastissimo repertorio di caratteri e il suo corrispondente insieme di codici.

ISO/IEC 10646, UCS e Unicode

- La differenza tra i due standard è che Unicode definisce uno standard completamente compatibile con ISO/IEC 10646, ma a differenza di quest'ultimo (che è più astratto Unicode) impone vincoli addizionali sull'implementazione dei caratteri in modo da garantirne la compatibilità tra differenti piattaforme e applicazioni.
- Quindi per dirla breve i due standard non sono esattamente la stessa cosa anche se sono in qualche modo allineati.
- Ogni versione di standard di Unicode identifica la corrispondente versione nello standard ISO/IEC 10646.
- La pagina ufficiale di Unicode http://www.unicode.org/versions/ permette di identificare per ogni versione se e come è allineata ad una versione di ISO/IEC 10646.

Unicode

- In pratica le persone generalmente parlano di Unicode invece che di ISO 10646, in parte perché si preferisce usare i nomi invece di numeri, in parte perché Unicode è più esplicito sul significato dei caratteri e le informazioni possono essere facilmente recuperate su internet.
- La versione 1.0 di Unicode usa nomi abbastanza differenti per i caratteri rispetto a ISO 10646. Nella versione 2.0 di Unicode i nomi sono stati fatti coincidere con quelli di ISO 10646. Nuove versioni hanno aggiunto principalmente nuovi caratteri dalla 3.0 (con 49 194 caratteri) fino alla 4.1.0 con 97 720 caratteri.

Unicode

- Lo standard Unicode definito dallo Unicode Consortium era originalmente progettato per essere un codice a 16 bit (che equivale a 2^{16}=65536 caratteri), ma fu esteso in modo da permettere codici nell'intervallo esadecimale 0..10FFFF vale a dire 1 114 112 caratteri.

- Tipicamente un carattere Unicode viene identificato con l'abbreviazione U+xxxx dove xxxx è un numero esadecimale a quattro cifre. Ad esempio la lettera A viene indicata in Unicode come U+0041 (65 in decimale).

Unicode

- Unicode e ISO/IEC 10646 hanno definito diverse forme di codifica dei loro comuni repertorio:
- UTF-8, UCS-2, UTF-16, UCS-4 e UTF-32.
- Spesso, per comodità, le codifiche UTF sono associate a Unicode mentre le codifiche UCS a ISO/IEC 10646.

UCS-2 e UTF-16

- UCS-2 e UTF-16 sono i nomi due codifiche di caratteri quasi identiche.
- UCS-2 è uno schema a due byte ed è sostanzialmente un'estensione di ISO Latin 1.
- Ad esempio nel caso della lettera maiuscola A rappresentata in ISO Latin 1 con il codice 41 esadecimale in UCS-2 verrebbe rappresentato da 2 bytes 00 41.

UCS-2: *big endian* e *little endian*

- Esistono due varianti di della codifica UCS-2, big endian e little endian.
- La differenza sta nell'ordine dei due byte.
- In big endian il byte più significato viene rappresentato per primo, quindi per esempio la lettera A corrisponde al codice 4100 esadecimale,
- viceversa little endian rappresenta prima il byte meno significativo, quindi A è rappresentato da 0041 esadecimale.

UCS-2: *big endian* e *little endian*

- Per permettere ai parser di interpretare correttamente testi codificati in UCS-2, si introduce all'inizio del testo un carattere speciale FEFF esadecimale, che in gergo è chiamato carattere Zero-Width No-Break Space (ZWNBSP)
- Si chiama così perché è un carattere che può essere usato in qualunque contesto di whitespace (cioè ovunque tranne in mezzo alle parole) senza modificare il significato dei testi.
- comunemente noto come byte order mark contrassegno dell'ordine dei byte (BOM).
- Se il parser riceve FEFF all'inizio del flusso allora il testo viene interpretato come big endian, se riceve FFFE il testo è interpretato come litte endian.

UCS-2: *svantaggi*

- **L'uso di UCS-2 comporta, però, tre svantaggi:**
  - il file occupano il doppio di spazio rispetto a quelli codificati in ISO Latin 1 (due byte invece di uno);
  - UCS-2 non è compatibile all'indietro con ASCII (programmi che si aspettano testi codificati con byte singoli non possono leggere testi un UCS-2);
  - UCS-2 è limitato a rappresentare massimo 65.536 caratteri.

UTF-16 e surrogati

- La soluzione all'ultimo dei tre problemi è stato affrontato dalla codifica UTF-16 attraverso l'uso dei cosiddetti *surrogati*.
- Di fatto la codifica UTF-16 è identica alla codifica UCS-2 ad eccezione di quest'ultimo aspetto.

UTF-16 e surrogati

- La codifica UTF-16 riserva 1024 codici, detti surrogati alti (high surrogates) e altrettanti detti surrogati bassi (low surrogates).
- Ogni possibile sequenza formata da un surrogato alto seguito da uno basso codificava un cosiddetto carattere surrogato: diventava dunque possibile codificare 1024 X 1024 nuovi caratteri, cioè 1 048 576.
- I surrogati alti sono i codici che vanno da D800 a DBFF esadecimale mentre quelli bassi, da DC00 a DFFF. Questi codici non si prestano a confusione in quanto non erano definiti nello standard UCS-2.
- Ad esempio il carattere 10FFFD esadecimale, corrispondente al limite superiore di Unicode, viene rappresentato con la sequenza DBFF, DFFD.
- Unicode non assegna a nessun carattere un valore compreso tra D800 e DFFF, evitando in questo modo che i singoli elementi di una coppia surrogata possano essere confusi con un carattere Unicode valido.

UTF-16 e surrogati

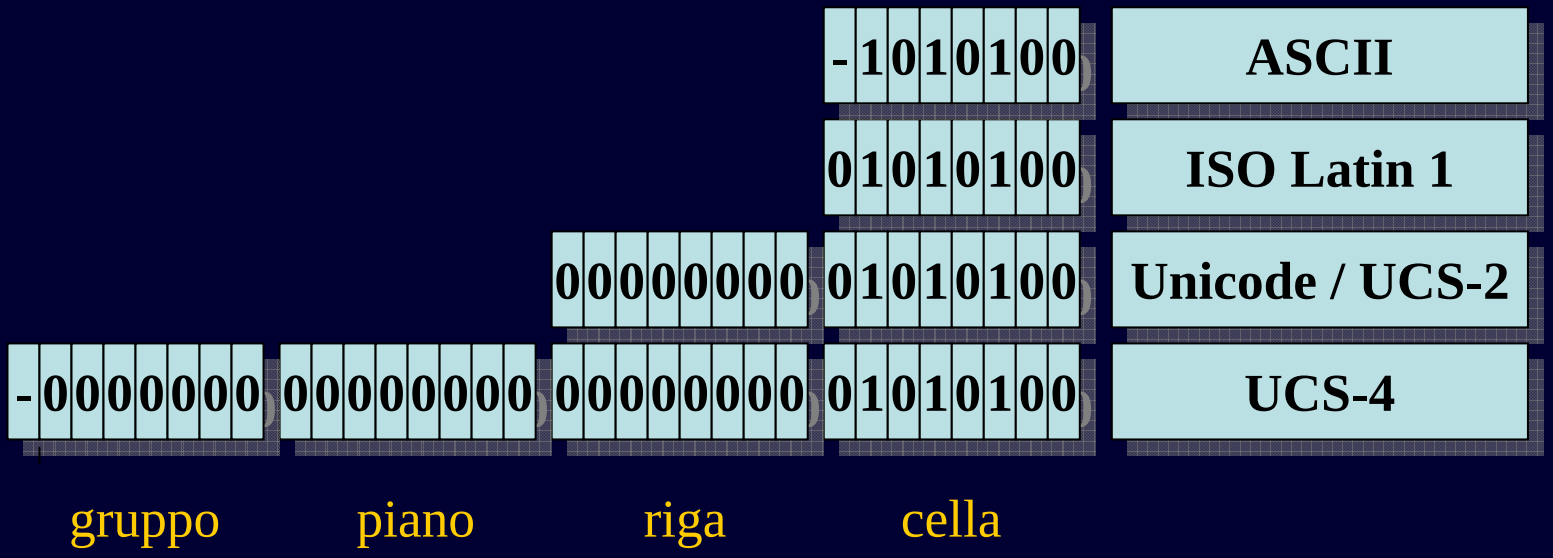
- **Ad esempio il primo carattere fuori dall'intervallo 0000-FFFF U+10000 è codificato in UTF-16 con i due surrogati:**
  - D800 DC00
- **Mentre l'ultimo carattere U+10FFFD verrebbe codificato con i due surrogati:**
  - DBFF DFFD

UTF-16 e UCS-2

- **Dato che nessun software utilizza i surrogati di UTF-16 la distinzione con UCS-2 è puramente accademica.**

UCS-4 e UTF-32

- **UCS-4 è è una estensione di UCS-2 che utilizza uno schema a 31 bit in 4 byte fissi. UCS-4 diviso in gruppi, piani, righe e celle.**



UCS-4

- In UCS-4 esistono dunque 32768 piani di 65536 caratteri ciascuno.
- Il primo piano, o piano 0, è noto come BMP (Basic Multilingual Plane) ed è ovviamente equivalente a UCS-2 quando il gruppo è 0.
- I gruppi vanno dallo 0 al 7F esadecimale (e non FF in quanto si utilizzano solo 31 bit).
- Solo il gruppo 0 è utilizzato attualmente nello standard gli altri (quello da 1-7F) sono riservati.

UCS-4

- **Per quanto riguarda il gruppo 0, sono definiti caratteri sono nei seguenti piani:**
  - Piano 0 (BMP o Basic Multilingual Plane): tutti gli alfabeti moderni.
  - Piano 1 (SMP o Supplementary Multilingual Plane): tutti gli alfabeti antichi.
  - Piano 2 (SIP o Supplementary Ideographic Plane): ulteriori caratteri ideografici CJK (chinese, japanese, Korean) non presenti in BMP.
  - Piano 14 (SSP o Supplementary Special-purpose Plane): Caratteri tag

UTF-32

- Considerando solo lo spazio dei codici da 0 a 10FFFF, UTF-32 è identico UCS-4.
- Però, allo scopo di rendere le due codifiche interoperabili il documento “Principles and Procedures” del JTC1/SC2/WG2 ha dichiarato che i codici oltre 10FFFF non di UCS-4 non saranno più utilizzati, rendendo in pratica lo standard UCS-4 un alias di UTF-32.

UTF-8

- UTF-8 è una codifica a lunghezza variabile di Unicode che risolve i primi due problemi dell'UCS-2 di cui si è discusso sopra.
- I caratteri da 0 a 127, ovvero il set di caratteri ASCII, vengono codificati con un byte ciascuno, esattamente come avviene nella codifica ASCII.
- In ASCII, il byte con valore 65 rappresenta la lettera A e anche in UTF-8 il byte 65 rappresenta la lettera A. Quindi esiste un'identità uno-a-uno tra i caratteri ASCII e i byte di UTF-8.
- Questo significa che i file scritti in ASCII puro risultano anche accettabili come file UTF-8.

UTF-8

- **UTF-8 permette di accedere a tutti i caratteri definiti di UCS-4, ma utilizza un numero compreso tra 1 e 4 byte per farlo. In particolare:**
  - i codici compresi tra 0 - 127 (ASCII a 7 bit) richiedono un byte, in cui ci sia 0 al primo bit;
  - i codici derivati dall'alfabeto latino e tutti gli script non-ideografici richiedono 2 byte;
  - i codici ideografici (orientali) richiedono 3 byte;
  - i codici dei piani alti richiedono 4 byte.

UTF-8

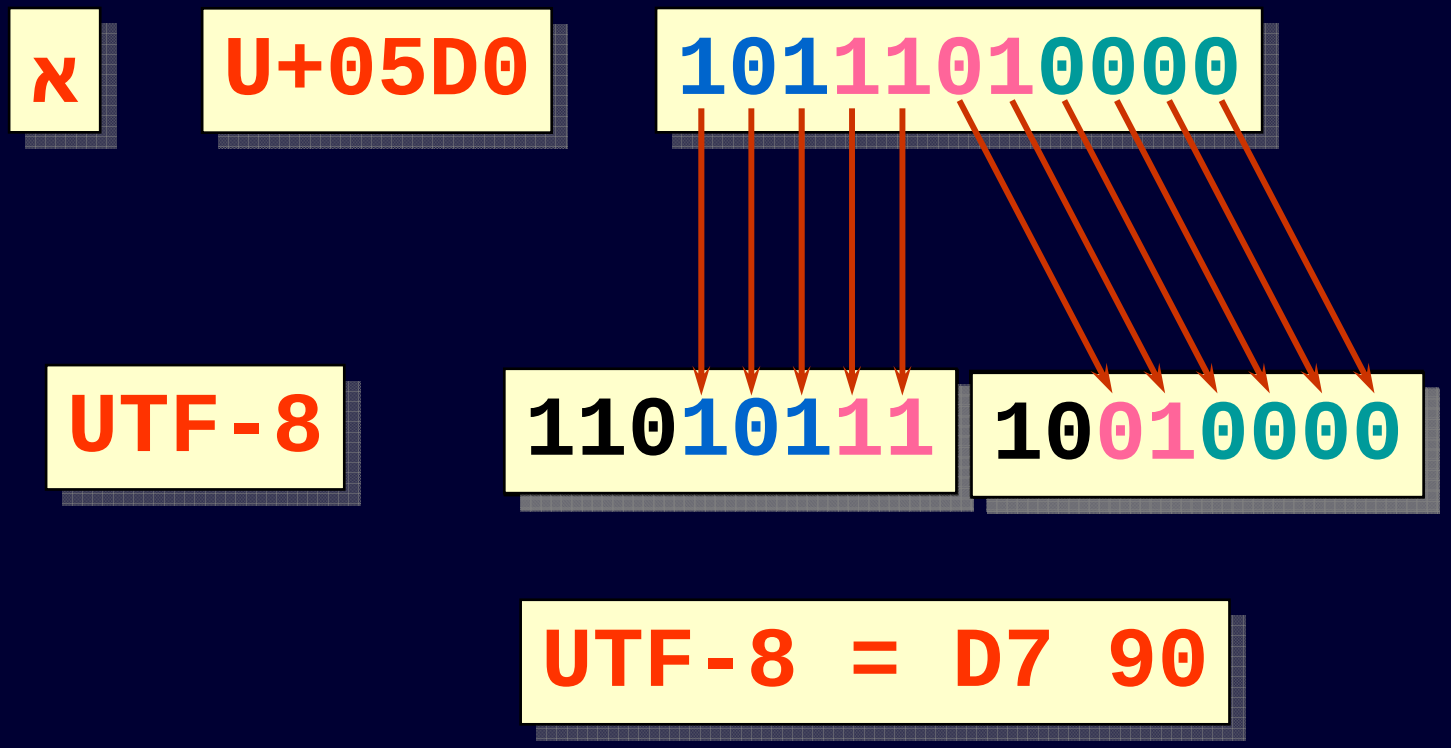
- **La seguente tabella mostra la corrispondenza dei codici di caratteri**

U-00000000 – U-0000007F:	0*xxxxxxx*
U-00000080 – U-000007FF:	110*xxxxx* 10*xxxxxx*
U-00000800 – U-0000FFFF:	1110*xxxx* 10*xxxxxx* 10*xxxxxx*
U-00010000 – U-001FFFFF:	11110*xxx* 10*xxxxxx* 10*xxxxxx* 10*xxxxxx*

UTF-8

- Per esempio, il carattere alef א, corrispondente all'Unicode U+05D0 esadecimale, viene rappresentato in UTF-8 con questo procedimento:
- ricade nell'intervallo da 0x0080 a 0x07FF. Secondo la tabella verrà rappresentato con due byte. 110xxxxx 10xxxxxx
- l'esadecimale 05D0 equivale al binario 101-1101-0000
- gli undici bit vengono copiati in ordine nelle posizioni marcate con "x". 110-10111 10-010000
- il risultato finale è la coppia di byte 11010111 10010000, o in esadecimale D7 90.

UTF-8



UTF-8

- Per concludere, UTF-8 è probabilmente il tipo di codifica Unicode supportato in maniera più diffusa.
- Per esempio, le stringhe dei file Java class sono memorizzate utilizzando tale codifica.
- Di conseguenza, la codifica UTF-8 è la codifica di default che viene assunta da un parser XML a meno che non ne venga specificata una utilizzando una dichiarazione dl codifica.
- Esistono ottime possibilità che un programma che dichiara di salvare file in Unicode, utilizzi la codifica UTF-8.

Glifi e Fonts

- **Per mostrarci un testo sullo schermo o stamparlo su carta, il computer ha bisogno di codificare due tipi di informazioni ben diverse:**
  1. il puro e semplice contenuto del testo, l'equivalente di un originale dattilografato
  2. la forma o layout del testo, cioè una serie di codici che ne definiscono il formato (numero di colonne, larghezza dei margini, eccetera) e che indicano i punti dove iniziano e terminano determinati attribuiti grafici (il tipo di font, la dimensione del testo, il suo colore, eccetera).

Glifi e Fonts

- È bene precisare che Unicode (e qualsiasi altro standard per la codifica del testo) si occupa solo e unicamente del punto 1.
- Un testo digitale che contenga solo la codifica del contenuto si definisce testo semplice (plain text);
- se invece contiene anche informazioni sul formato tipografico si definisce testo ricco (rich text).
- Unicode si occupa di codificare il plain text. Le informazioni aggiuntive necessarie al rich text vengono definite da Unicode protocollo di livello superiore (higher level protocol), e la loro codifica viene ignorata in quanto considerata materia di pertinenza di altri standard.

Glifi e Fonts

- **Proprio a causa della loro indeterminatezza grafica, i caratteri di Unicode si definiscono come *caratteri astratti*.**
- **I caratteri visibili sullo schermo si definiscono invece come *glifi* e l'esatta definizione della loro forma dipende dal font utilizzato e dalle scelte estetiche fatte da chi l'ha progettato.**

Glifo

- Il glifo è una particolare forma con cui un carattere può essere rappresentato sullo schermo o su carta.
- Per esempio, il carattere Z potrebbe essere rappresentato come grassetto **Z** o corsivo *Z*.
- D'altra parte, il carattere minuscolo z è definito come un carattere separato, il quale a sua volta potrebbe essere associato ad un glifo diverso.
- La decisione di uno standard di mantenere in un repertorio forme diverse dello stesso carattere come maiuscolo e minuscolo come caratteri separati è completamente arbitraria.
- Infatti, in Unicode ci sono diversi esempi di caratteri che potrebbero essere considerate semplicemente delle varianti tipografiche di uno stesso carattere, ma che per varie ragioni sono tenuti come caratteri separati.

Font

- Un repertorio di glifi costituisce un font.
- Più tecnicamente un font è un insieme numerato di glifi. Il numero corrisponde al codice del carattere (rappresentato dal glifo).
- Quindi in certo senso il font è dipendente dal codice del carattere.
- È possibile che un glifo utilizzato per un dato carattere venga utilizzate per altri caratteri. Ad esempio, benché i caratteri della lettera A latina, A (alfa) greca siano distinti in Unicode questi vengono rappresentati con lo stesso glifo.