Rappresentazione di Caratteri, Stringhe ed altri dati

Fondamenti di Informatica I
Corso di laurea in Ingegneria Informatica e Automatica
Sapienza Università di Roma

A.A. 2018-19

Domenico Lembo, Paolo Liberatore,
Alberto Marchetti Spaccamela, Marco Schaerf

Rappresentazione di caratteri

- Il calcolatore è in grado di manipolare solo simboli elementari che noi codifichiamo normalmente in “0” ed “1”.
- Tutte le informazioni possono essere rappresentate con sequenze di “0” e “1”.
- Siamo quindi interessati a capire come possiamo realizzare tali rappresentazioni.
- Partiamo dalla rappresentazione dei caratteri:
  - ogni carattere si rappresenta con un numero (codificato in binario)
  - versioni più semplici: un numero senza segno a sette o otto bit
  - versione moderna più utilizzata: sequenza di numeri senza segno a otto bit

Codifica ASCII

- ASCII: American Standard Code for Information Interchange
- pubblicato dall'American National Standards Institute (ANSI) nel 1968.
- E' un sistema di codifica a 7 bit. Questo consente di utilizzare 2^7=128 numeri (da 0 a 127) per codificare altrettanti caratteri.
- **Esempi**:

  'z' → 1111010 (numero 122 - espresso in binario con 7 bit)

  ';' → 0111011 (numero 59)

  'E' → 1000101 (numero 69)

  '0' → 0110000 (numero 48)
- Invece di 7 bit è possibile usarne 8, ponendo il primo bit pari a 0 (quindi, ad esempio, la codifica di 'z' diventa 01111010, che rappresenta sempre il numero 122 espresso in binario con 8 bit − la rappresentazione binaria dei numeri naturali sarà oggetto di una lezione successiva)

Caratteri speciali

- alcuni numeri (da 0 a 31, ed il 127) non rappresentano veri caratteri, ma simboli speciali (non stampabili), comunemente detti **caratteri di controllo**
- Ad esempio
  - 0 (chiamato NUL) indica la fine di una stringa
  - 7 (chiamato BEL) indica un beep, inteso come suono!
  - 9 (chiamato TAB) indica una tabulazione orizzontale
  - 10 (chiamato LF, e cioè Line Feed) indica l'andata a capo
  - 13 (chiamato CR, e cioè Carriage Return) indica lo spostamento del cursore all'inizio della linea
  - 19 (chiamato DC3, dove DC sta per Device Control) indica la richiesta sospensione trasmissione
  - 17 (chiamato DC1) indica richiesta ripresa trasmissione

La tabella ASCII

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

Limiti della codifica ASCII

- La codifica ASCII cattura solo un limitato di caratteri. Ad esempio, mancano le lettere accentate italiane (à, è, é, ì, ò, ù), o la ñ spagnola, i caratteri tedeschi ä, ö, ü, ß.

- Ovviamente mancano molti altri simboli, come gli ideogrammi o i simboli matematici e chimici, o anche un simbolo di uso molto comune come il simbolo dell'euro.

- Altri codifiche sono quindi state proposte per ampliare l'uso dei caratteri rappresentabili.

- Nel seguito ci concentriamo in particolare sulle codifiche ISO-8859-1 e Unicode UTF-8

ISO-8859

- ISO-8859 è uno standard proposto da ISO (International Organization for Standardization) ed IEC (International Electrotechnical Commission) per la codifica di caratteri ad 8 bit.

- ISO-8859 è in realtà un insieme di standard (ISO-8859-i, con i nell'intervallo 1..16 e diverso da 12, essendo quest'ultimo abbandonato).

- In tutte le ISO-8859-i è possibile codificare $2^8 = 256$ caratteri. I numeri della codifica da 1 a 127 sono come in ASCII (codificato con 8 bit, con primo bit a sinistra pari a 0), mentre gli altri cambiano nei vari standard.

- ISO-8859-1 (Latin 1 Western European) codifica i caratteri usati nella maggior parte delle lingue europee occidentali, incluse l'Italiano, lo Spagnolo ed il Tedesco. E' probabilmente la parte di ISO-8859 più usata.

ISO-8859-1

130:										
140:										
150:										
160:		¡	¢	£	¤	¥	¦	§	¨	©
170:	ª	«	¬		®	¯	°	±	²	³
180:	´	µ	¶	·	¸	¹	º	»	¼	½
190:	¾	¿	À	Á	Â	Ã	Ä	Å	Æ	Ç
200:	È	É	Ê	Ë	Ì	Í	Î	Ï	Ð	Ñ
210:	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û
220:	Ü	Ý	Þ	ß	à	á	â	ã	ä	å
230:	æ	ç	è	é	ê	ë	ì	í	î	ï
240:	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù
250:	ú	û	ü	ý	þ	ÿ				

- *I numeri da 0 a 127 codificano caratteri come nel codice ASCII e non sono riportati in figura.*
- *I numeri da 128 a 159 non sono utilizzati*
- *Il numero 160 codifica lo spazio "non-breaking"*

Limiti della codifica ISO-8859-1

- Questo sistema va bene per lingue come l'italiano, l'inglese, lo spagnolo, il norvegese, ma copre solo in parte i simboli usati in altre lingue (ad es. lingue dell'est europa, il russo, l'arabo, il turco, ecc.).
- Per queste bisogna ricorrere agli altri standard ISO-8859
- Per esempio, ISO-8859-9 consente di codificare tutti i caratteri usati nella lingua turca, ed usa il numero 253 per la i senza punto ı invece che per la y con accento acuto ý come in ISO-8859-1.
- Quindi, le varie ISO-8859 sono incompatibili fra loro e bisogna sempre specificare quale ISO-8859 si sta usando
- Non si possono scrivere testi in più lingue insieme.
- alcune lingue, come il cinese, hanno più di 255 caratteri.

Unicode

- Unicode (anche detto Universal Coded Character Set, o UCS) è un sistema di codifica in grado di rappresentare i caratteri usati in quasi tutte le lingue vive e in alcune lingue morte, simboli matematici e chimici, cartografici, l'alfabeto Braille, ideogrammi ecc.
- Per rappresentare concretamente i caratteri, Unicode prevede tre possibili codifiche
  - UTF-8, sequenza fino a 4 unità da 8 bit
  - UTF-16, sequenza fino a 2 unità da 16 bit (è una evoluzione del precedente UCS-2, codifica di lunghezza fissa a 2 byte)
  - UTF-32 (nota anche come UCS-4), sequenza di esattamente 32 bit per carattere
- Delle tre, UTF-8 è la più efficiente nel gestire lo spazio, consentendo anche di usare una sola unità da 8 bit (cioè un byte) per rappresentare un carattere.

UTF-8

- i primi 128 caratteri, che sono gli stessi di ASCII, rappresentati come un singolo numero a otto bit, il cui primo bit è 0
- per gli altri caratteri, sequenza di numeri a otto bit che hanno tutti 1 come primo bit; il primo di questi numeri è nella forma 1...10..., e il numero di uni indica la lunghezza della sequenza.

Num Byte	Da num	A num	Byte 1	Byte 2	Byte 3	Byte 4
1	0	127	0xxxxxxx			
2	128	2047	110xxxxx	10xxxxxx		
3	2048	65535	1110xxxx	10xxxxxx	10xxxxxx	
4	65536	1114111	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

Nota: Potenzialmente, UTF-8 potrebbe usare sequenze fino a 6 byte, ma nel 2003 è stato limitato per coprire solo l'intervallo descritto formalmente nello standard Unicode, per il quale 4 byte sono sufficienti (massimo numero rappresentabile 1.111.4111 = U+10FFFF in esadecimale).

Stringhe

Le stringhe sono sequenze di caratteri

Ad esempio 'ciao a tutti! ' è una stringa che in Python possiamo stampare con il comando

```
print('ciao a tutti!')
```

sequenza fra virgolette = stringa = sequenza di caratteri

Ogni carattere è un numero. I numeri corrispondenti ai caratteri vengono memorizzati in sequenza. La stringa precedente è così rappresentate

99 105 97 111 32 97 32 116 117 116 116 105 33 0

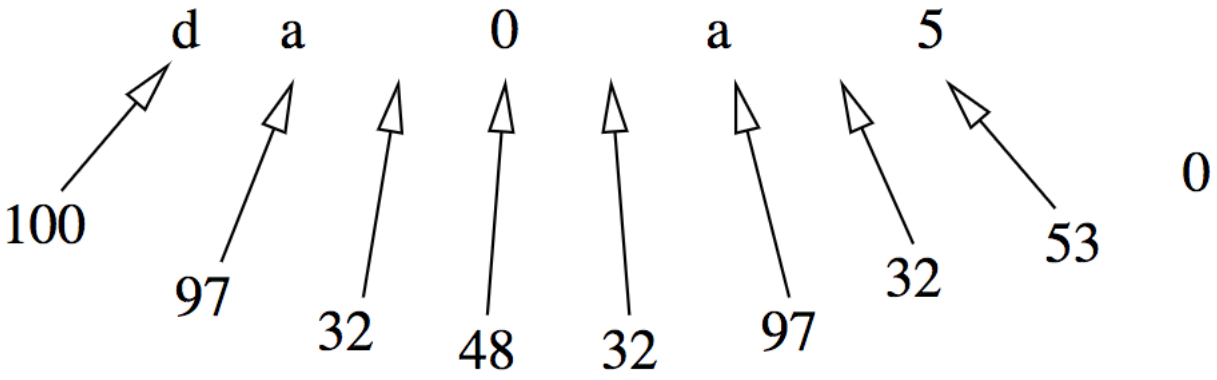
99 = c, 105 = i, 97 = a, …

32 = spazio

0 = fine stringa

Rappresentazione delle Cifre

- In ASCII, ISO-8859-1, UTF-8 le cifre ‘0’..’9’ sono rappresentate con i numeri 48..57.
- Quindi nell’esempio precedente lo 0 non corrisponde alla cifra ‘0’, ma è il terminatore di stringa
- Esempio: stringa ‘da 0 a 5’



Alternative all'uso del carattere speciale NUL

all'inizio viene specificato quanti caratteri ci sono nella stringa:

00013 99 105 97 111 32 97 32 116 117 116 116 105 33

00013 indica che la stringa è di 13 caratteri.

Il primo numero è a sedici bit invece che a otto come gli altri in modo da poter così venire usato per stringhe lunghe più di 256 caratteri (fino a 2^{16}-1=65535).

Ritorno a capo

Il ritorno a capo viene indicato con

- Il numero 10 (cioè Line Feed), oppure
- con la sequenza 10 13 (cioè Line Feed e Carriage Return)

C'è un motivo storico per questo, legato all'uso delle prime stampanti:

- 10 = avanzamento carta di una linea
- 13 = ritorno del carrello a inizio linea

Suoni

Sono onde di pressione dell'aria

E’ possibile fornirne una rappresentazione numerica:

- Pressione misurata a intervalli regolari (es. 48000 volte al secondo)
- Ciascun valore rappresentato in binario (es. a 16 bit)
- Suono = sequenza di questi valori

La sequenza è uguale all’originale

- variazioni fra una misurazione all'altra non vengono rilevate
- la pressione è un valore continuo

numero prefissato di bit = approssimazione

fedeltà all'originale = alta frequenza di campionamento + alto numero di bit

Alcuni protocolli (come il MIDI) definiscono come rappresentare suoni di strumenti musicali elettronici.

Colori e Immagini

- La maggior parte dei colori visibili all'occhio umano si possono considerare un miscuglio di quantità variabili di rosso, verde e blu

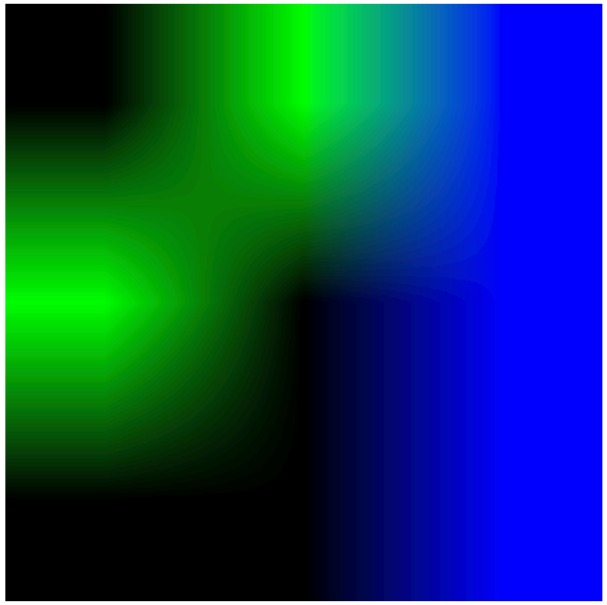
  Esempio: massimo di rosso, mezzo verde e niente blu 

Immagine = griglia di minuscoli quadrettini (pixel)

- ogni pixel viene considerata di un colore solo
- per ogni pixel, si rappresentano le quantità di rosso, verde e blu che contiene.

Rappresentazione delle immagini

Un semplice formato (ppm)

Immagine (ingrandita)	Rappresentazione
	P3 3 3 100 0 0 0 0 100 0 0 0 100 0 100 0 0 0 0 0 0 100 0 0 0 0 0 0 0 0 100

parte iniziale

identificativo del formato (P3)

larghezza e altezza dell'immagine (3×3)

massima intensità di colore (100)

matrice

i colori dei pixel, in sequenza (es. 0 100 0 = niente rosso, max verde, niente blu)

Immagini rappresentate come quadrati di pixel si dicono *raster*.

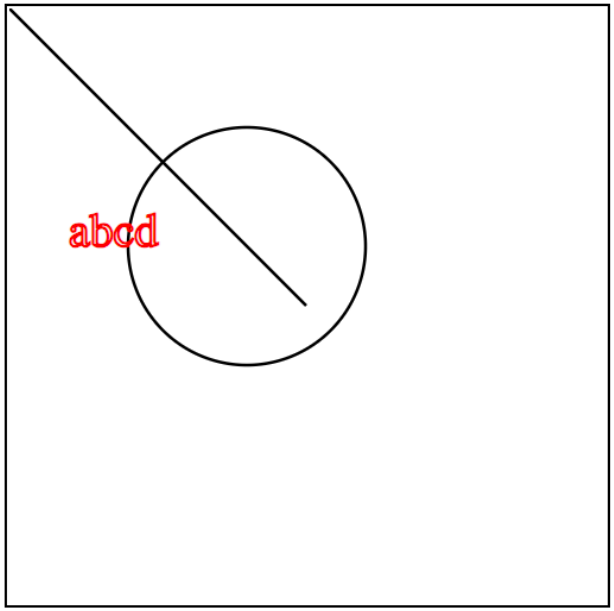
Approssimazioni e compressioni

- Come per i suoni, si tratta comunque di una rappresentazione approssimata, per i soliti due motivi: ogni pixel viene considerato di un colore unico (quindi variazioni di colore più piccole di un pixel vengono ignorate), e i colori vengono rappresentati con un numero finito di bit (per cui esiste una perdita di precisione).

- Esistono meccanismi di compressione, che permettono di ridurre lo spazio di memoria richiesto. Sono basati principalmente su sequenze che si ripetono (es. spesso pixel vicini = colori simili)

Immagini vettoriali

invece dei pixel: figure geometriche elementari

Immagine	Rappresentazione
	<?xml version="1.0" standalone="no"?> <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd"> <svg xmlns="http://www.w3.org/2000/svg"> <g style="stroke:#000000;fill:none;"> <polyline points="0,0 100,100" /> <circle cx="80" cy="80" r="40" /> <text x="20" y="80" stroke="#FF0000">abcd</text> </g> </svg>

immagini vettoriali: tag

<polyline points="0,0 100,100" /> segmento da coordinata 0,0 a 100,100

<circle cx="80" cy="80" r="40" /> cerchio

<text x="20" y="80" stroke="#FF0000">abcd</text> scritta abcd in rosso

origine: in alto a sinistra

Immagini vettoriali

immagini vettoriali: meccanismo

sequenze di caratteri (o numeri)

nell'esempio: da <?xml fino a </svg>

in genere si possono inserire come elementi (oltre a linee, cerchi, ecc.) anche delle immagini raster