

Calcolo numerico e **programmazione**

Rappresentazione delle informazioni

Tullio Facchinetti

<tullio.facchinetti@unipv.it>

23 marzo 2012

09:19

<http://robot.unipv.it/toolleeo>

Codifica di dati non numerici: definizioni

- i calcolatori lavorano soltanto con i numeri
- devono poter trattare anche altre entità
 - testo
 - immagini
 - suoni

per questa ragione sono stati inventati i codici

Alfabeto usato dai calcolatori

alfabeto

è un insieme finito di simboli (caratteri) distinti

esempi:

- $U = \{1\}$ alfabeto unario
- $A = \{0, 1\}$ alfabeto binario
- $B = \{A, B, C, \dots, Z\}$ alfabeto inglese
- $C = \{A, B, \dots, Z, 0, 1, \dots, 9\}$ alfabeto alfanumerico

parola

è una successione finita di caratteri di un alfabeto

- 1011001101 è una parola di A
- SFSJKGH è una parola di B
- SF120C è una parola di C

Alfabeto usato dai calcolatori

i calcolatori lavorano utilizzando due simboli:

- interruttore: aperto/chiuso
- transistor: conduce/non conduce
- livello di tensione: alto/basso
- riflettività di un'areola: alta/bassa
- dominio di magnetizzazione: $\uparrow \downarrow$

Alfabeti usati dall'uomo

- $\{A, B, C, D, \dots, Z\}$
- $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- $\{ ; , . ! ? : () ' " \dots \}$
- { Caratteri giapponesi }
- { Caratteri arabi }

è necessario stabilire delle regole di corrispondenza, dette **codifiche**

la codifica mette in corrispondenza biunivoca ogni **simbolo** appartenente all'alfabeto più ricco con una **stringa** di simboli appartenenti all'alfabeto più ridotto

Codifica

- sia dato un insieme I di elementi qualsiasi
- sia dato un alfabeto A

si dice codifica di I mediante parole di A un procedimento che permette di stabilire una corrispondenza biunivoca tra gli elementi di I e le parole di un sottoinsieme Q dell'insieme delle parole di A

esempio: codifica di lettere tramite cifre decimali

$A \rightarrow 01, B \rightarrow 02, C \rightarrow 03, D \rightarrow 04, Z \rightarrow 26$

Codifica binaria

un codice binario è la codifica dei simboli di un alfabeto S mediante stringhe di bit

- sia C è la cardinalità di S (n. di elementi)
- sia n il numero di bit da utilizzare

allora deve valere $C \leq 2^n$, ossia:

$$n \geq \lceil \log_2 C \rceil = M$$

teorema

dati due alfabeti $A = \{a_1, a_2, \dots, a_k\}$, $B = \{b_1, b_2, \dots, b_h\}$ con $h, k > 1$ è sempre possibile codificare l'insieme delle parole di A mediante parole di B

Codici non ridondanti e ridondanti

- se $n = M$ il codice è non ridondante (usa il minimo numero di bit per codificare tutti i simboli di S)
- se $n > M$ il codice è ridondante (usa più bit del necessario per codificare tutti i simboli di S)

si definisce distanza di Hamming (H) di un codice il minimo numero di bit di cui differiscono due parole qualsiasi del codice:

- se $n = M \rightarrow H = 1$
- se c'è ridondanza $\rightarrow H > 1$

Codici non ridondanti e ridondanti

aggiungendo opportunamente la ridondanza alle parole, si possono ottenere codici a rilevazione e/o a correzione di errore

si può dimostrare che

- per i codici a sola rilevazione, il numero di bit errati rilevati è $R = H - 1$
- per i codici a correzione, il numero di bit errati corretti è $C \leq (H - 1)/2$

Parità

sono codici con distanza di Hamming $H > 1$ e possono avere la caratteristica di rivelare o correggere uno o più bit errati

- è un codice con $H = 2$
- si ottiene dal codice non ridondante aggiungendo 1 bit in modo che il numero complessivo di bit uguali a “1” sia pari (parità pari) o dispari (parità dispari)
- avendo $H = 2$ può solo rilevare gli errori, e solo se questi accadono singolarmente o in un numero dispari nella stessa parola di codice

Parità pari

codice non ridondante a 7 bit in cui il bit di parità diventa il bit più significativo (bit aggiuntivo in posizione b_7)

esempio:

● 0 1 1 1 0 1 1 → 1 0 1 1 1 0 1 1

● 0 1 0 1 0 1 1 → 0 0 1 0 1 0 1 1

Codifica delle cifre decimali

vi sono diverse codifiche delle cifre decimali 0123456789:

codice BCD (Binary Coded Decimal)

- ogni cifra decimale viene tradotta con 4 bit
- è un codice pesato perché il valore di ogni cifra viene ottenuto eseguendo una somma pesata delle quattro cifre binarie che lo compongono

codice ad eccesso tre

- è basato sul codice BCD
- ogni codifica si ottiene sommando 3 alla codifica BCD
- non è un codice pesato
- il complemento di ogni cifra si ottiene complementando le cifre binarie della sua codifica

Codifica delle cifre decimali

codice 2 su 5

- ogni configurazione presenta sempre due 1

codice 2421

- i numeri indicano ordinatamente i pesi delle 4 cifre binarie
- come l'eccesso tre, è complementabile

codice Gray

- le diverse configurazioni sono adiacenti
- ognuna differisce dalla precedente solo per un bit

Codifica delle cifre decimali

	Gray	2 su 5	BCD	Eccesso 3	2421
0	0000	00110	0000	0011	0000
1	0001	00011	0001	0100	0001
2	0011	00101	0010	0101	0010
3	0010	01001	0011	0110	0011
4	0110	01010	0100	0111	0100
5	0111	01100	0101	1000	1011
6	0101	10001	0110	1001	1100
7	0100	10010	0111	1010	1101
8	1100	10100	1000	1011	1110
9	1101	11000	1001	1100	1111

- sono tutti codici che lasciano alcune configurazioni inutilizzate
- il codice 2 su 5 permette di rilevare errori
- il codice Gray elimina il problema di transizioni spurie

Codifica di informazioni alfanumeriche

per convenzione, si definisce alfabeto esterno di un calcolatore l'insieme dei caratteri che è in grado di leggere e stampare mediante i dispositivi di I/O

esempio codice Fielddata:

- le 26 lettere dell'alfabeto inglese maiuscole
- le 10 cifre decimali
- 28 caratteri vari (spazio, segni di punteggiatura)

Codifica di informazioni alfanumeriche

in genere si utilizzano codici a 7 o 8 bit

EBCDIC (Extended Binary Coded Decimal Interchange Code)

- codice a 8 bit
- le prime configurazioni corrispondono con il BCD

UNICODE

- rappresenta con 16 bit tutti i caratteri della lingua parlata dall'uomo
- usato inizialmente da Java
- adottato poi da HP, IBM, Microsoft, Oracle, ...

Codice ASCII: American Standard Code for Information Interchange

ne esistono due versioni

7 bit

- 128 simboli
- in questo caso l'ottavo bit è usato a volte come ridondanza ($H=2$) per la rilevazione di un errore (bit di parità)

8 bit

- 256 simboli (ASCII esteso)
- codifica anche lettere accentate e caratteri grafici
- l'estensione non è standardizzata e quindi fra i 128 simboli aggiunti può succedere che alla stessa codifica corrispondano simboli diversi

Codice ASCII

è il codice più usato per la codifica dei caratteri alfanumerici (lettere e cifre) oltre che per simboli di interpunzione e vari

ad esempio:

. ; : , () [] { } / \ < > = ? ! ~ | # \$ % & * ^ - @ ecc.

Codice ASCII

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

fonte Wikipedia

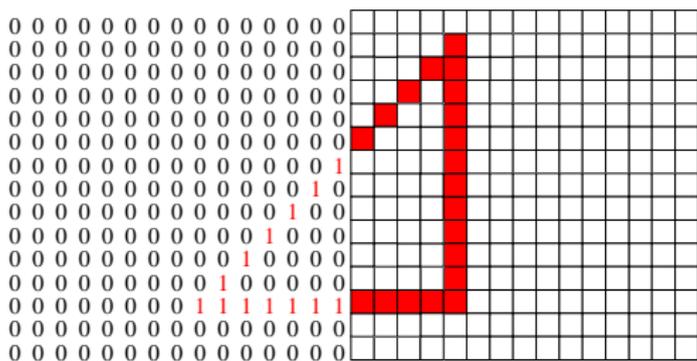
Codifica delle immagini

- in un calcolatore tutto è discreto
- l'immagine è un insieme continuo di informazioni (luce, colore) in due dimensioni
- quando si memorizzano immagini pittoriche o fotografiche si scompone artificialmente l'immagine in una sequenza di elementi di informazione codificati con sequenze binarie



Codifica bitmap di immagini

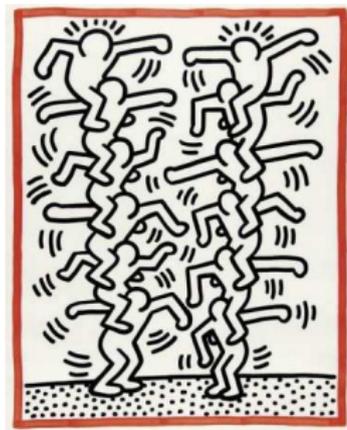
- l'immagine viene suddivisa in un reticolo di punti detti pixel (picture element)
- ogni pixel viene codificato con una sequenza di bit



la qualità dell'immagine dipende dal numero di pixel per unità di lunghezza e dal numero di bit utilizzati per codificare ogni pixel

Codifica bitmap di immagini

problema: non è possibile ingrandire a piacimento un'immagine perché non si aumenta il dettaglio, ma si distinguono i pixel



la soluzione di aumentare il numero di pixel non è possibile perché questo aumenterebbe lo spazio necessario per memorizzare una immagine

Codifica dei pixel: immagini in bianco e nero

- raramente si usa un solo bit
- ogni pixel di solito si codifica con 8 bit per rappresentare diversi livelli di grigio ($2^8 = 256$ livelli)



8 bit – 256 livelli



2 bit – 4 livelli



1 bit – 2 livelli

Codifica dei pixel: immagini a colori

- i colori vengono ottenuti tramite la combinazioni di almeno 3 colori base, detti primari
- la composizione può avvenire tramite la tecnica di sintesi sottrattiva o additiva in funzione del tipo di dispositivo utilizzato (stampante, monitor o televisore)

Codifica dei pixel: immagini a colori

- per codificare un pixel si devono codificare i tre colori primari (es. RGB), la cui combinazione consente di ottenere il colore del pixel stesso
- per ciascun colore primario spesso si usano 8 bit e quindi in totale per codificare ciascun pixel servono 24 bit
- ciò consente di codificare $2^{24} \sim 16$ milioni di colori diversi
- con il termine **profondità di colore** si intende il numero di bit utilizzati per codificare i pixel

Occupazione di memoria di un'immagine

l'occupazione di memoria di un'immagine dipende da:

- definizione o risoluzione dell'immagine (intesa come il numero di pixel per unità di lunghezza: dot per inch = DPI)
- numero dei colori (dipende a sua volta dal numero di bit usati per codificarli: 8, 16, 24 bit)

tipo immagine	definizione	colori	occupazione
televisiva	720 × 650	256 (8 bit)	~ 440 KB
monitor PC	1024 × 768	65536 (16 bit)	~ 1.5 MB
fotografica	15000 × 10000	16.000.000 (24 bit)	~ 430 MB

Formati di file bitmap

TIFF (Tagged Image File Format)

profondità fino a 24 bit e vari metodi di compressione

GIF (Graphics Interchange Format)

consente di memorizzare più immagini in uno stesso file

JPEG (Joint Photographers Expert Group)

- formato standard che usa due tipi di compressione molto utilizzato su Internet
- prevede la tecnica di visualizzazione interallacciata (l'immagine viene composta per righe alterne per cui viene caricata velocemente, ma diventa più nitida man mano che viene completata)

Formati di file bitmap

BMP (BitMaP)

usata con Windows supporta immagini con profondità 1, 4, 8 e 24 bit

PNG (Portable Network Graphics)

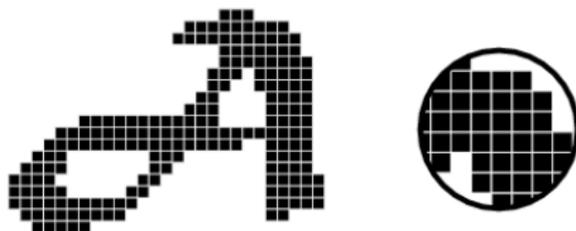
- creato per l'uso su Internet: adottato come formato standard
- nato a causa dell'imposizione di royalties sul formato GIF
- visualizzazione progressiva dell'immagine mediante interlacciamento
- informazioni ausiliare di qualsiasi natura accluse al file
- immagini a colori fino a 48 bpp (bit per plane);
- immagini in scala di grigio sino a 16 bpp;

Immagini vettoriali

- si usa in applicazioni di progettazione meccanica, architettonica, elettronica
- l'immagine viene costruita con elementi di alto livello quali linee, cerchi, poligoni, archi, colori, ecc.
- ogni oggetto è codificato attraverso un identificatore (es. polyline, circle, etc.) e alcuni parametri quali le coordinate del centro e la lunghezza del raggio (per la circonferenza) o le coordinate dei vertici (per il poligono)

Immagini vettoriali

al contrario delle immagini bitmap in cui l'elemento di informazione è il pixel, in quelle vettoriali gli elementi dell'immagine sono gli oggetti grafici che la compongono



Formato vettoriale per le immagini

vantaggi:

- indipendenza dal dispositivo di visualizzazione e dalla sua risoluzione
- gli elementi grafici sono indipendenti l'uno dall'altro e si possono elaborare distintamente
- minore occupazione di memoria rispetto alla grafica bitmap

svantaggi:

- applicabilità limitata: un'immagine fotografica non si può scomporre in elementi primitivi

Formati di file vettoriali

PostScript

- incorporato in molte stampanti, si è evoluto nel tempo e consente di rappresentare immagini vettoriali e bitmap anche a colori con diverse tecniche di compressione
- formati derivati del PostScript sono EPS e PDF

DXF (Drawing Interchange Format)

- usato da molti programmi di disegno memorizza i vettori in formato testo
- la versione binaria DXB è più compatta ma meno diffusa

Formati di file vettoriali

SVG (Standard Vector Graphics)

- formato basato sul linguaggio derivato dall'XML
- formato vettoriale standard per il Web
- formato di file aperto
- supporto nativo da parte di molti browser
- permette di gestire figure dinamiche e interattive

un formato di file aperto è un specifica pubblica per la descrizione e l'archiviazione di dati digitali solitamente gestita da un ente di standardizzazione non proprietario e libera da restrizioni legali per il suo utilizzo

Rappresentazione di immagini in movimento

- immagini in movimento vengono rappresentate attraverso sequenze di immagini fisse (frame) visualizzate ad una frequenza sufficientemente alta da consentire all'occhio umano di ricostruire il movimento (24, 25 o 30 immagini al secondo)
- lo standard multimediale per le immagini in movimento, MPEG, codifica ciascun frame separatamente secondo lo standard JPEG
- si utilizzano tecniche di compressione in quanto un minuto di filmato a 25 fotogrammi al secondo richiederebbe uno spazio di memorizzazione di 644 MByte (un CD contiene 680 MByte)
- alcuni tipici formati di file video sono avi, mpg, 3gp, mp4, flv e mov, webm