Informatica

La codifica di numeri, caratteri, immagini e suoni

Codifica dell’informazione

Rappresentazione dei simboli usati dall’uomo nel suo linguaggio naturale (alfabeto esterno) in simboli adatti per essere interpretati dall’elaboratore (alfabeto interno).

L’alfabeto esterno è composto da almeno 64 simboli:

- lettere dell’alfabeto inglese,
- cifre decimali → numeri naturali, interi, reali,
- segni di interpunzione.

L’alfabeto interno è composto dalle cifre binarie.

Codifica dell'informazione

Teorema:

Dati due alfabeti $A = \{a_1, a_{2,} \dots a_n\}$ e $B = \{b_1, b_{2,} \dots b_p\}$ con n e $p \geq 1$, è sempre possibile codificare (rappresentare) tutte le parole di A usando i simboli di B, anche quando $p<n$.

Il metodo per passare dai simboli di A a quelli di B è detto **sistema di codifica**.

Codifica dell’informazione

I codici possono essere:

- a lunghezza fissa o variabile,
- ridondanti,
- numerici o alfanumerici.

Codifica dell’informazione

Codici numerici e alfanumerici

Sono numerici quei codici che ad una stringa di bit fanno corrispondere esclusivamente numeri, mentre sono alfanumerici quei codici che ad una stringa di bit fanno corrispondere numeri e simboli alfabetici.

Codici numerici

I numeri naturali

Si possono rappresentare in binario puro con **formato fisso** (numero di bit occupabili deciso a priori e non modificabile). Se il formato è di n bit si possono rappresentare i primi 2^n-1numeri naturali. Se la codifica non riempie tutti i bit si colma con 0 non significativi.

Codici numerici

I numeri naturali

La codifica in **formato fisso** è di difficile interpretazione, soprattutto per numeri alti. Si privilegiano, quindi, altre codifiche che siano più comode per le operazione di I/O o per l’esecuzione delle operazioni.

Codici numerici

Aritmetica finita

Quando si eseguono operazioni su numeri codificati su una quantità finita e fissa di cifre può capitare che il risultato ecceda le cifre disponibili per la sua memorizzazione. Questa condizione prende il nome di **overflow.**

Esempio: In una codifica a 8 bit voglio eseguire

$$
\begin{array}{r}
1000\ 0010\ + \\
1001\ 0001 = \\
\hline
\mathbf{1}\ 0001\ 0011
\end{array}
$$

Codici numerici

Rappresentazione dei numeri negativi

I numeri negativi vengono generalmente rappresentati in

- Modulo e segno
- Complemento a 2 (alla base)
- Complemento a 1 (alla base diminuita)
- Eccesso M

Codici numerici

Rappresentazione in modulo e segno

In una codifica a n bit il bit più a sinistra (MSB Most Significant Bit), detto anche bit di segno, rappresenta il segno (0 = + e 1 = -) ed i restanti n-1 bit rappresentano il valore assoluto del numero.

Se non si riempiono tutti gli n-1 bit del valore assoluto si colma con 0 non significativi.

Codici numerici

Rappresentazione in modulo e segno

In una rappresentazione a n bit, uno è destinato al segno e n-1 al modulo, quindi i numeri rappresentabili sono tutti gli interi appartenenti all'intervallo $[-2^{n-1}+1, 2^{n-1}+1]$.

Codici numerici

Rappresentazione in modulo e segno

Esistono due 0 uno positivo (00…0) e l'altro negativo (10…0).

Problemi con le operazioni, in particolare non è vero che x-x=0. E' necessario gestire separatamente il bit di segno ed i bit del modulo.

Codici numerici

Rappresentazione in modulo e segno

Esempi:

Overflow sul bit di segno

In una codifica a 4 bit la somma di numeri positivi potrebbe generare un negativo

0 100 + 0 100 = 1 000

La somma di un numero e dell'opposto non dà 0.

+5 + (-5) = 00000101 + 10000101 = 10001010 ≠ 0

Codici numerici

Complemento a uno

Il complemento a uno di un numero binario si calcola invertendo tutte le cifre del numero (mettiamo 1 dove c'è 0 e viceversa).

Esempio:

1 0 0 1 1 0 1

0 1 1 0 0 1 0 $\boxed{\textbf{Complemento a uno}}$

Codici numerici

Rappresentazione in complemento a 1

Supponiamo di avere a disposizione n bit.
I numeri interi positivi vengono rappresentati mediante codifica binaria pura su n-1 bit e completati con 0. Si ottiene la medesima rappresentazione ottenuta col modulo e segno.

I numeri interi negativi vengono rappresentati col complemento a 1 dell'opposto.
(Esempio: per rappresentare -5 calcolo il complemento a 1 di +5)

Codici numerici

Rappresentazione in complemento a 1

In questo modo si ha una rappresentazione posizionale, in cui la prima cifra vale $-2^{n-1}+1$ e le altre 2^{n-1}, ma non si risolve il problema del doppio 0.

L'intervallo di rappresentazione è $[-2^{n-1}+1, 2^{n-1}-1]$.

Codici numerici

Complemento a due

Il complemento a due di un numero binario si calcola aggiungendo 1 al complemento a uno del numero.

$$
\begin{array}{r}
1\ 0\ 0\ 1\ 1\ 0\ 1 \\
0\ 1\ 1\ 0\ 0\ 1\ 0\ + \\
1\ = \\
\hline
0\ 1\ 1\ 0\ 0\ 1\ 1
\end{array}
$$

Complemento a uno

Complemento a due

Codici numerici

Rappresentazione in complemento a 2

Supponiamo di avere a disposizione n bit.
I numeri interi positivi vengono rappresentati mediante codifica in formato fisso su n-1 bit e completati con 0. Si ottiene la medesima rappresentazione ottenuta col modulo e segno.

I numeri interi negativi vengono rappresentati col complemento a 2 dell'opposto.

Il bit più significativo esprime il segno del numero.

Codici numerici

Rappresentazione in complemento a 2

In una rappresentazione a 3 bit si possono avere le seguenti combinazioni:

Decimale	Complemento a 2
0	000
1	001

Decimale	Complemento a 2
-4	100
-3	101

Codici numerici

Rappresentazione in complemento a 2

C'è un unico zero, positivo.
L'intervallo di rappresentazione è $[-2^{n-1}, 2^{n-1}-1]$.

ATTENZIONE:
Al numero di bit da usare nella rappresentazione.
Al segno del numero da rappresentare (<u>si complementano solo i negativi</u>).

Codici numerici

Rappresentazione in complemento a 2

Esempi: In una rappresentazione a 6 bit si ha

+21 = 0 10101

-24 = 1 01000

+7 = 0 00111

-7 = 1 11001

Codici numerici

Rappresentazione in eccesso M

Dato un numero N espresso in base b la sua rappresentazione in eccesso M sarà

E(N)=N+M

con M espresso in base b e generalmente uguale a 2^{n-1} o 2^{n-1} -1, dove n è il numero di bit scelto per la rappresentazione.

Al variare del numero di bit si ottiene una rappresentazione diversa.

Codici numerici

Rappresentazione in eccesso M

I numeri interi verranno rappresentati tutti in eccesso M (prima si determina E(N) poi si trasforma in binario).

In questo modo, all’intervallo $[-\ 2^{n-1}, 2^{n-1}\ -1]$ rappresentabile usando n bit si sostituisce l’intervallo $[0, 2^n]$.

Su una rappresentazione a 8 bit si ha $M = 2^7 = 128$ e

$+4 \rightarrow E(+4)=+132 \rightarrow 10000100$

$-4 \rightarrow E(-4) = +124 \rightarrow 01111100$

Codici numerici

Rappresentazione in eccesso M

Facilità ad eseguire le somme ma non le altre operazioni.

Viene mantenuto l’ordinamento, quindi è una rappresentazione comoda per somme, operazioni relazionali e logiche.

Codici numerici

Rappresentazione dei numeri decimali

I numeri razionali e reali (approssimati) potrebbero essere rappresentati con i metodi precedentemente descritti, a meno dell'introduzione del punto decimale ma richiederebbero troppi bit per poter rappresentare numeri significativamente grandi o piccoli.

Per questo motivo si usano le rappresentazioni
- in virgola fissa
- in virgola mobile

Codici numerici

Rappresentazione in virgola fissa

Una parte dei bit a disposizione è deputata a rappresentare la parte intera del numero e la parte di bit restante viene usata per rappresentare la parte decimale del numero. I due contributi, essendo interi, vengono rappresentati con una delle modalità viste in precedenza.

La posizione della virgola è fissa e sottintesa.

Codici numerici

Rappresentazione in virgola fissa

Se si hanno a disposizione n bit si avrà p+q=n-1

S	p bit per la parte intera	q bit per la parte frazionaria

In questo modo l'intervallo di rappresentazione è

$[-2^{p}-1, 2^{q}-1 \; ; +2^{p}-1, 2^{q}-1]$

C'è una rappresentazione doppia dello 0.

Codici numerici

Rappresentazione in virgola fissa

A causa della necessaria approssimazione l'aritmetica in virgola fissa non è esatta sebbene sia rapida.

Esempio: $\frac{1}{3} + \frac{1}{3} + \frac{1}{3} = 1$

$0{,}3333 + 0{,}3333 + 0{,}3333 = 0{,}9999 \neq 1$

Non è utilizzabile per applicazioni scientifiche o commerciali ma in tutti quei casi in cui si preferisce la velocità alla precisione (riproduzione audio-video).

Codici numerici

Rappresentazione in virgola mobile

Ogni numero razionale e le approssimazioni dei reali possono essere rappresentati in base b come

$$\mathbf{N = m * b^{e}}$$

dove m è detta mantissa ed e è detto esponente.

Codici numerici

Rappresentazione in virgola mobile

Ogni numero N ha infinite rappresentazioni. Per esempio

$6{,}043 = 6043 * 10^{-3} \rightarrow (6043, -3)$

$= 60{,}43 * 10^{-1} \rightarrow (60{,}43, -1)$

$= 0{,}6043 * 10 \rightarrow (0{,}6043, 1)$

Per ogni classe di equivalenza si sceglie come rappresentante la rappresentazione avente la mantissa di una sola cifra.

Codici numerici

Rappresentazione in virgola mobile

Lo standard IEEE 754 prevede 3 formati:

1 bit	8 bit	23 bit	
S	Esponente	Mantissa	**Singola precisione** 32 bit

1 bit	11 bit	52 bit	
S	Esponente	Mantissa	**Doppia Precisione** 64 bit

1 bit	15 bit	64 bit	
S	Esponente	Mantissa	**Precisione estesa** 80 bit

Codici numerici

Rappresentazione in virgola mobile

Lo standard IEEE 754 prevede che ogni numero N, in base 2, sia rappresentato come:

- Il rappresentante ha parte intera pari a 1.
- La mantissa viene rappresentata in modulo e segno.
- L’esponente viene rappresentato in eccesso 127.

$$(-1)^{\text{segno}} * (1 + \text{mantissa}) * 2^{(\text{esponente} - 127)}$$

Codici numerici

Rappresentazione in virgola mobile

Esempio: Rappresentare il numero -5,75 in virgola mobile-singola precisione.

1)Convertire in base 2 → 101,11
2)Normalizzare → 1,**0111** * 2^2
3)Tronchiamo le eventuali cifre periodiche della mantissa a 23 o colmiamo con 0 non significativi le cifre mancanti.
4)Convertire l'esponente (eccesso 127)

1	10000001	**0111**0000000000000000000

Codici numerici

Rappresentazione in virgola mobile

Esempio: Determinare l'equivalente decimale del seguente numero rappresentato in virgola mobile-singola precisione.

0	01111110	**101**00000000000000000000

$$(-1)^{\text{segno}} * (1 + \text{mantissa}) * 2^{(\text{esponente} - 127)}$$

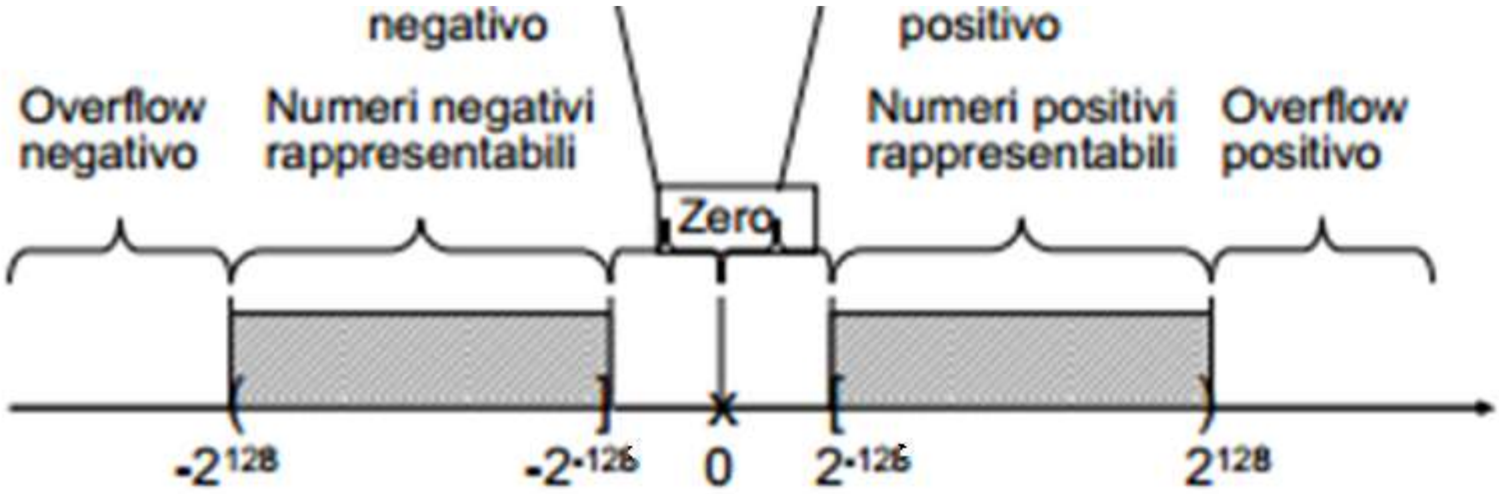
$$(-1)^{0} * (1 + 0,101) * 2^{(126 - 127)}$$

$$1,101*2^{-1} = 0,1101 = 0,8125$$

Codici numerici

Rappresentazione in virgola mobile

Gli estremi dell'intervallo di rappresentazione dipendono dal numero di cifre destinate all'esponente.



2^{-127} è riservato a NaN (Not a Number), usato, per esempio quando si cerca di calcolare la radice pari di un numero negativo.

Codici numerici

Aritmetica in virgola mobile

L'aritmetica in virgola mobile può presentare dei problemi:

- + e * non sono associative
- Non vale la proprietà distributiva di * rispetto a +
- Non unicità degli elementi neutri di + e *
- Assorbimento di numeri piccoli da numero molto grandi
- Cancellazione (0 come differenza tra numeri diversi ma molto vicini)
- Overflow
- Underflow
- Errori di arrotondamento

Codici alfanumerici

Per codificare i 64 simboli dell'alfabeto esterno occorrono un egual numero di sequenze di simboli dell'alfabeto interno. Poiché i simboli dell'alfabeto interno sono 2, per poter codificare 64 simboli sono necessari almeno 6 bit, infatti $2^6 = 64$.

Poiché i bit vengono generalmente raggruppati in byte, abbiamo una codifica che ci consente di rappresentare $2^8 = 256$ simboli, avendo la possibilità di aggiungere a quelli già citati alcuni simboli speciali.

Codici alfanumerici

Il codice ASCII

Il codice ASCII (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange) è uno dei codici alfanumerici usati dai calcolatori.

Assegna un codice a ciascun carattere.

I codici sono contigui e crescenti per lettere dell’alfabeto maiuscole, minuscole e cifre decimali.

I primi 32 simboli sono riservati a caratteri di controllo.

Dec	Sym	Dec	Char	Dec	Char	Dec	Char
0	NUL	32		64	@	96	`
1	SOH	33	!	65	A	97	a
2	STX	34	"	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL	39	'	71	G	103	g
8	BS	40	(	72	H	104	h
9	TAB	41	)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	,	76	L	108	l
13	CR	45	-	77	M	109	m
14	SO	46	.	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DC1	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3	51	3	83	S	115	s
20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	ETB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z
27	ESC	59	;	91	[	123	{
28	FS	60	<	92	\	124	|
29	GS	61	=	93	]	125	}
30	RS	62	>	94	^	126	~
31	US	63	?	95	_	127	⌂

Tabella ASCII Standard

Dec	Char	Dec	Char	Dec	Char	Dec	Char
128	Ç	160	á	192	+	224	Ó
129	ü	161	í	193	-	225	ß
130	é	162	ó	194	-	226	Ô
131	â	163	ú	195	+	227	Ò
132	ä	164	ñ	196	-	228	õ
133	à	165	Ñ	197	+	229	Õ
134	å	166	ª	198	ã	230	µ
135	ç	167	º	199	Ã	231	þ
136	ê	168	¿	200	+	232	Þ
137	ë	169	®	201	+	233	Ú
138	è	170	¬	202	-	234	Û
139	ï	171	½	203	-	235	Ù
140	î	172	¼	204	¦	236	ý
141	ì	173	¡	205	-	237	Ý
142	Ä	174	«	206	+	238	¯
143	Å	175	»	207	¤	239	´
144	É	176	_	208	¶	240	Û
145	æ	177	_	209	Đ	241	±
146	Æ	178	_	210	Ê	242	_
147	ô	179	¦	211	Ë	243	¾
148	ö	180	¦	212	È	244	¶
149	ò	181	Á	213	ı	245	§
150	û	182	Â	214	Í	246	÷
151	ù	183	À	215	Î	247	¸
152	ÿ	184	©	216	Ï	248	°
153	Ö	185	¦	217	+	249	¨
154	Ü	186	¦	218	+	250	·
155	ø	187	+	219	_	251	¹
156	£	188	+	220	_	252	³
157	Ø	189	¢	221	¦	253	²
158	×	190	¥	222	Ì	254	_
159	ƒ	191	+	223	_	255	

Tabella ASCII estesa

Codici alfanumerici

Il codice ASCII

Un testo scritto in ASCII attraverso un semplice editor (es. Notepad per Windows o Vi per Linux) è codificato da una sequenza di byte, ognuno dei quali rappresenta un simbolo.

Word processor più complessi aggiungono altri caratteri in testa al file, che rappresentano le caratteristiche grafiche del testo.

Codici alfanumerici

Il codice ASCII

Esempio: Un file scritto con Notepad e contenente la parola *ciao* ha dimensione di 4 byte.

Lo stesso file scritto in Word ha dimensione 12.631 byte.

Aprendo file .doc con Notepad è possibile visualizzare tutti quei caratteri di controllo che fanno aumentare la dimensione del file.

Codici alfanumerici

Il codice UNICODE

E’ un codice a 16 bit che consente di codificare 65636 caratteri, creato e pubblicizzato dall’ Unicode Consortium. E’ indipendente dalla lingua e dalla piattaforma.

Contiene lettere e cifre delle principali lingue vive del modo (anche gli ideogrammi ed il Braille) e di alcune lingue morte (es. cuneiforme e fenicio), simboli matematici e chimici.

Codifica delle immagini

Le immagini che sono per noi continue devono essere discretizzate, al fine di poter essere memorizzate o trattate dall’elaboratore. La scomposizione avviene tramite un reticolo di ‘quadratini’ detti pixel (**pic**ture **el**ements). Il reticolo è detto bitmap.

Ogni pixel può essere rappresentato con uno o più bit.

Più pixel ci sono (e quindi minore è la loro dimensione) e più l’immagine appare continua.

Codifica delle immagini

Si dice risoluzione di una immagine la sua qualità, dipendente dal numero di pixel e dalla dimensione dell'immagine.

A parità di pixel → + risoluzione = -grandezza

A parità di grandezza → + risoluzione = +pixel

Risoluzione ~ numero di pixel / grandezza

Codifica delle immagini

Bianco e nero

Ogni pixel è codificato con 1 bit, consentendo di rappresentare solo il bianco e il nero.

Codifica delle immagini

Scala di grigio

Ogni pixel è codificato con 8 bit, consentendo di rappresentare 256 tonalità di grigio, dal bianco al nero.

Codifica delle immagini

Immagini a colori

CMYK: elimina (sottrae) dalla luce bianca le frequenza relative ai colori primari (ciano, magenta, giallo). Usata dalle stampanti per sovrapposizione di diversi strati di colore in diverse percentuali.

Codifica delle immagini

Immagini a colori

RGB: Si sommano le lunghezze d'onda del rosso, verde e blue.

Usata nei monitor.

Si usano 8 bit per colore per un totale di 24 bit = 3 byte.

Nero = tutto 0

Bianco puro = tutto 1

Codifica delle immagini

Immagini a colori

E’ necessario fissare il numero di pixel e la quantità di colori rappresentabili (palette, sottoinsieme di tutti i colori possibili). Ogni pixel è di un singolo colore ed è codificato dalla sequenza di bit che rappresenta quel colore.

Si occupa meno spazio.

Codifica dei suoni

I suoni sono onde rappresentabili come funzioni continue del tempo e per essere rappresentati devono diventare dati discreti, quindi c'è perdita di informazione.

1) trasduzione: onda sonora → segnale elettrico

2) discretizzazione

3) quantizzazione: si approssima il valore della funzione nei punti di discretizzazione con valori discreti. A seconda della quantità di bit si avranno più o meno livelli di quantizzazione.

Codifica dei suoni

Wav: standard per la rappresentazione dei suoni su CD.

Valore di campionamento di 44.100 Hertz e quantizzazione a 16 bit.

Mp3, Mp4: standard per la trasmissione di file audio su internet. Taglia i suoni oltre la soglia di udibilità effettuando di fatto una compressione.