La Rappresentazione dell'Informazione

Maurizio Palesi
Salvatore Serrano

L'Informatica

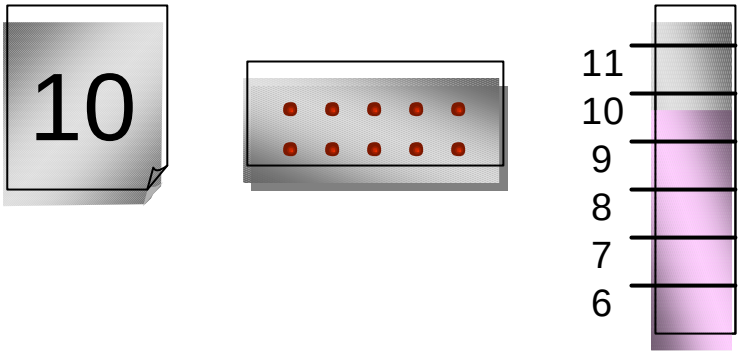
- Cos'è l'Informatica?
  - ➔Studio sistematico degli algoritmi che descrivono e trasformano l'informazione: la loro teoria, analisi, progetto, efficienza, realizzazione e applicazione [*ACM - **A**ssociation for **C**omputing **M**achinary*]
  - ➔Scienza della rappresentazione e dell'elaborazione dell'informazione.
- Attenzione
  - ➔Non si parla di tecnologia degli elaboratori!
  - ➔Si attribuisce ruolo centrale al concetto di informazione!

Non Esiste Informazione senza Supporto

- L'informazione è "portata da" o "trasmessa su" o "memorizzata in" o "contenuta in" qualcosa
  - ➔Questo qualcosa NON è l'informazione stessa
- Ogni supporto ha le sue caratteristiche in quanto mezzo su cui può essere scritta dell'informazione
  - ➔Alcuni supporti sono particolarmente adatti alla trasmissione dell'informazione, ma non alla sua memorizzazione (aria)
  - ➔Per altri supporti vale il viceversa (compact disk)

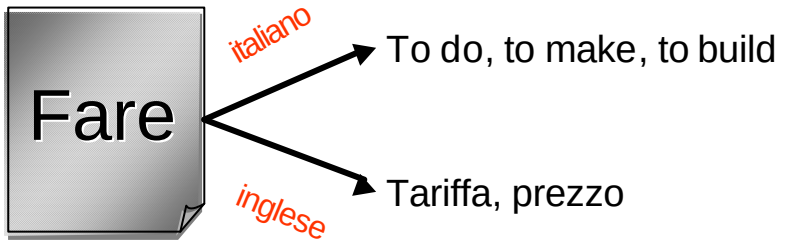
Informazione e Supporti (1)

- La stessa informazione può essere scritta su supporti differenti



Informazione e Supporti (2)

- Lo stesso supporto può portare informazioni diverse

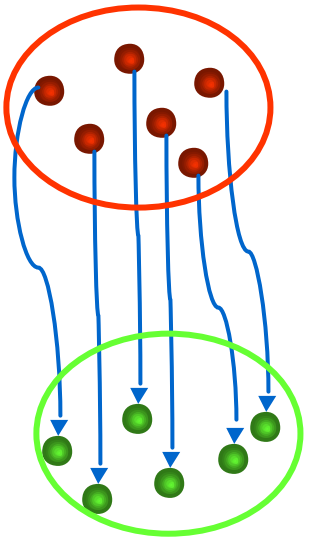


Simboli ed Alfabeto

- Per formalizzare dati (numeri, caratteri, immagini, suoni, …) si utilizzano successioni di simboli scelti da un insieme finito detto alfabeto
  - ➔Alfabeto: {A,B,C,…,Z}, Simboli: A, B, …
  - ➔Alfabeto: {0,1,2,…,9}, Simboli: 0, 1, …
  - ➔Alfabeto: {✂✍🔔⚐☠💣}, Simboli: 💣🔔⚐
- Ad ogni alfabeto è associato un insieme di regole di composizione
  - ➔Regole per la composizione dei numeri
  - ➔Regole per la composizione delle parole
- Successione ben formate di simboli di un alfabeto sono associate a dati mediante codici

Definire un Codice

- Indentificare
  - ➔{ configurazioni }
  - ➔{ entità informazione }
- Associare gli elementi dei due insiemi



Informazione e Incertezza

- La presenza di informazione è condizionata dal fatto che il supporto sia in grado di assumere diverse configurazioni
- Se la nostra incertezza circa l'effettiva configurazione del supporto viene ridotta dall'accesso al supporto, allora
  - ➔Tale atto ci ha portato dell'informazione
- Se potessimo misurare l'incertezza prima e dopo la lettura, la quantità di informazione potrebbe essere definita dalla differenza tra tali gradi di incertezza

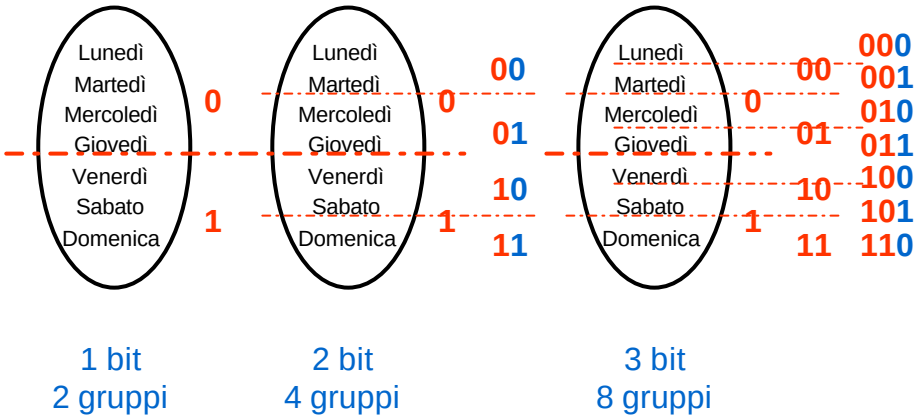
Informazione e Incertezza

- Ipotesi
  - ➔Si conoscono a priori le configurazioni che il supporto può assumere
  - ➔Non si sa quali di esse sia quella effettivamente assunta
- La quantità di informazione che si ottiene selezionando una configurazione da un insieme che ne contiene due è l'unità di informazione elementare (bit).

Codifica Binari

- Alfabeto
  - ➔2 simboli: {**0**, **1**}, {**on**, **off**}, {☺, ☹}
- Problema
  - ➔Assegnare un **codice univoco** a tutti gli oggetti appartenenti ad un insieme predefinito
- Quanti oggetti posso codificare con k bit?
  - ➔1 bit ⇒ (0, 1) ⇒ 2 oggetti
  - ➔2 bit ⇒ (00, 01, 10, 11) ⇒ 4 oggetti
  - ➔3 bit ⇒ (000, 001, 010, …, 111) ⇒ 8 oggetti
  - ➔…
  - ➔k bit ⇒ (...) ⇒ 2^k oggetti
- Quanti bit mi servono per codificare N oggetti?
  - ➔$N \leq 2^K \Rightarrow K \geq \log_2 N \Rightarrow K = \lceil \log_2 N \rceil$

I Giorni della Settimana in Binario



1 bit
2 gruppi

2 bit
4 gruppi

3 bit
8 gruppi

Codifica Binaria dei Caratteri

- Quanti sono gli oggetti compresi nell'insieme?
  - ➔26 lettere maiuscole + 26 lettere minuscole ⇒ 52
  - ➔10 cifre
  - ➔Circa 30 segni d'interpunzione
  - ➔Circa 30 caratteri di controllo (EOF, CR, LF, …)
  - ➔Circa 120 oggetti complessivi ⇒ $K = \lceil \log_2 120 \rceil = 7$
- Codice ASCII
  - ➔Utilizza 7 bit e quindi può rappresentare al massimo $2^7=128$ caratteri
  - ➔Con 8 bit (=byte) rappresento $2^8=256$ caratteri (ASCII esteso)
  - ➔Si stanno diffondendo codici più estesi (es. UNICODE) per rappresentare anche i caratteri delle lingue orientali

ASCII su 7 bit

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
010	sp	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
101	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
110	`	a	b	c	d	e	f	g	h	I	j	k	l	m	n	o
111	p	q	r	s	t	u	v	w	x	Y	z	{	|	}	~	canc

Bit, Byte, Kbyte, Mbyte, …

- Bit = solo due stati, ‘0’ oppure ‘1’
- Byte = 8 bit, quindi $2^8 = 256$ stati
- KiloByte [KB] = 2^{10} Byte = 1024 Byte ~ 10^3 Byte
- MegaByte [MB] = 2^{20} Byte ~ 10^6 Byte
- GigaByte [GB] = 2^{30} Byte ~ 10^9 Byte
- TeraByte [TB] = 2^{40} Byte ~ 10^{12} Byte
- PetaByte [PB] = 2^{50} Byte ~ 10^{15} Byte
- ExaByte [EB] = 2^{60} Byte ~ 10^{18} Byte

Numeri Naturali

- Sistema di numerazione posizionale in base ***b***
  - ➔ $c_K c_{K-1} \ldots c_0$ rappresenta $c_K \times b^K + c_{K-1} \times b^{K-1} + \ldots + c_0 \times b^0$
  - ➔ $1231_{10} = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10^1 + 1 \times 10^0$
- Conversione Binario ⇒ Decimale
  - ➔ Basta scrivere il numero secondo la notazione posizionale utilizzando già il sistema decimale
  - ➔ $1101_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

Conversione Decimale Þ Binaria

- Si calcolano i resti delle divisioni per 2

18 : 2 = 9	resto 0
9 : 2 = 4	resto 1
4 : 2 = 2	resto 0
2 : 2 = 1	resto 0
1 : 2 = 0	resto 1

10010

137 : 2 =	68	resto 1
68 : 2 =	34	resto 0
34 : 2 =	17	resto 0
17 : 2 =	8	resto 1
8 : 2 =	4	resto 0
4 : 2 =	2	resto 0
2 : 2 =	1	resto 0
1 : 2 =	0	resto 1

10001001

Numeri Interi

- Alfabeto binario
  - ➔ Anche il segno è rappresentato con 0 e 1
  - ➔ E' indispensabile indicare il numero **K** di bit utilizzati
- Modulo e Segno (MS)
  - ➔ 1 bit di segno (0 positivo, 1 negativo)
  - ➔ K-1 bit di modulo
    - ✓ Es. $+6_{10} = 0110_{MS}$, $-6_{10} = 1110_{MS}$
    - ✓ Si rappresentano i valori da $2^{K-1}+1$ a $2^{K-1}-1$
    - ✓ Con 4 bit i valori da -7 a +7
    - ✓ Con 8 bit i valori da -127 a +127
  - ➔ Attenzione
    - ✓ Doppia rappresentazione dello 0

Diverse Codifiche/Interpretazioni

Codice	Nat	MS
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7

Codice	Nat	MS
1000	8	-0
1001	9	-1
1010	10	-2
1011	11	-3
1100	12	-4
1101	13	-5
1110	14	-6
1111	15	-7

Ottali ed Esadecimali

- Utili per rappresentare sinteticamente valori binari
- Ottale (base **b=8**)
  - Alfabeto: cifre comprese tra 0 e 7
    - ✓ $354_8 = 3\times8^2 + 5\times8^1 + 4\times8^0 = 236_{10}$
  - Ogni cifra ottale corrisponde a 3 cifre binarie
    - ✓ $11101100_2 = [11][101][100] = 354_8$
- Esadecimale (base **b=16**)
  - Alfabeto: cifre comprese tra 0 e 9 e lettere tra A ed F
    - ✓ $EC_{16} = 14\times16^1 + 12\times16^0 = 236_{10}$
  - Ogni cifra esadecimale corrisponde a 4 cifre binarie
    - ✓ $11101100_2 = [1110][1100] = EC_{16}$

Numeri Razionali

- Rappresentazione in virgola fissa
  - $0.1011_2 = 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3} + 1\times2^{-4} = 0.6875_{10}$
  - $11.101_2 = 1\times2^1 + 1\times2^0 + 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3} = 3.625_{10}$
  - Il numero di cifre prima e dopo la virgola è fisso!
- Rappresentazione in virgola mobile (float)
  - Usata spesso anche in decimale per rappresentare numeri o molto grandi o molto piccoli (0.1357×10^{64})
    - ✓ mantissa: parte frazionaria compresa tra 0 e 1 (0.1357)
    - ✓ esponente: numero intero
  - Utilizza 1 bit per il segno (s), h bit per l'esponente (e) e k bit per la mantissa (m): $\mathbf{R = s \times m \times 2^e}$

Numeri Interi in Complemento a 2

- Alfabeto binario
  - ➔ Anche il segno è rappresentato con 0 e 1
  - ➔ E' indispensabile indicare il numero **K** di bit utilizzati
- Complemento a 2 (C2)
  - ➔ **X** corrisponde al binario naturale di $2^K + X$
    - ✓ $+6_{10} = 2^4 + 6 = 22$ Þ [1]0110 Þ 0110_{C2}
    - ✓ $-6_{10} = 2^4 + (-6) = 10$ Þ [0]1010 Þ 1010_{C2}
  - ➔ Si rappresentano i valori da -2^{K-1} a $+2^{K-1}-1$
    - ✓ Con 4 bit i valori da **-8 a +7**
    - ✓ Con 8 bit i valori da **-128 a +127**
  - ➔ Conversione **C2 Þ Decimale**
    - ✓ $b_k\, b_{k-1}\, b_{k-2} \ldots\, b_0 = -b_k \times 2^K + (b_{K-1} \times 2^{K-1} + \ldots + b_0 \times 2^0)$
  - ➔ C'è una sola rappresentazione dello 0

Il Complemento a 2

- Metodi alternativi per calcolare la rappresentazione di **-X** a partire da **X**
  - ➔ Effettuare il complemento di ogni bit di X e aggiungere 1
    - ✓ Rappresentazione di $+6_{10}=0110_{C2}$ (N.B. ci vogliono 4 bit!)
    - ✓ Complemento di tutti i bit Þ 1001_{C2} (corrisponderebbe a -7_{10})
    - ✓ Aggiungere 1 Þ 1010_{C2} (che corrisponde a -6_{10})
  - ➔ Partendo da destra e andando verso sinistra lasciare incariati tutti i bit fino al primo 1 compreso, complementare tutti gli altri bit
    - ✓ Rappresentazione di $+6_{10}=0110_{C2}$ (N.B. ci vogliono 4 bit!)
    - ✓ Gli ultimi due bit (**_._.1.0**) rimangono invariati
    - ✓ Gli altri due bit vengono complementati (**1.0.1.0**$_{C2}$)

Diverse Codifiche/Interpretazioni

Codice	Nat	MS	C2
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7

Codice	Nat	MS	C2
1000	8	-0	-8
1001	9	-1	-7
1010	10	-2	-6
1011	11	-3	-5
1100	12	-4	-4
1101	13	-5	-3
1110	14	-6	-2
1111	15	-7	-1