

# Informatica di Base<sup>1</sup> — Linea 1

Jianyi Lin

Dipp. di Matematica e Scienze dell'Informazione  
Università degli Studi di Milano, Italia

[jianyi.lin@unimi.it](mailto:jianyi.lin@unimi.it)

a.a. 2011/12

---

<sup>1</sup>© 2011 J.Lin, M. Monga. Creative Commons Attribuzione-Condividi allo stesso modo 2.5 Italia License.  
<http://creativecommons.org/licenses/by-sa/2.5/it/>. Rielaborazione del materiale © 2009/10 © S. Mascetti.

## Lezione IV: Rappresentazione dell'informazione

# Informazione

L'informatica studia la **rappresentazione** e **elaborazione** dell'informazione. Per ora abbiamo solo accennato a come viene rappresentata l'informazione nei calcolatori:

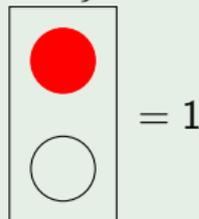
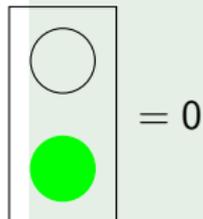
- Rappresentazione “numerica” (**codifica digitale**)
- In realtà i “numeri” sono solo un *alfabeto* di cifre usate per il loro valore simbolico (naturalmente è possibile codificare numeri, ma qualsiasi informazione viene rappresentata con il medesimo meccanismo)

La codifica digitale rende l'elaborazione una pura **manipolazione simbolica**: da una sequenza di simboli di partenza si produce una nuova sequenza di simboli.

## Esempio

### Informazione

Un semaforo può essere {rosso, verde}



### Elaborazione algoritmica

Se il semaforo è verde attraversa (= 1),  
altrimenti non attraversare (= 0)

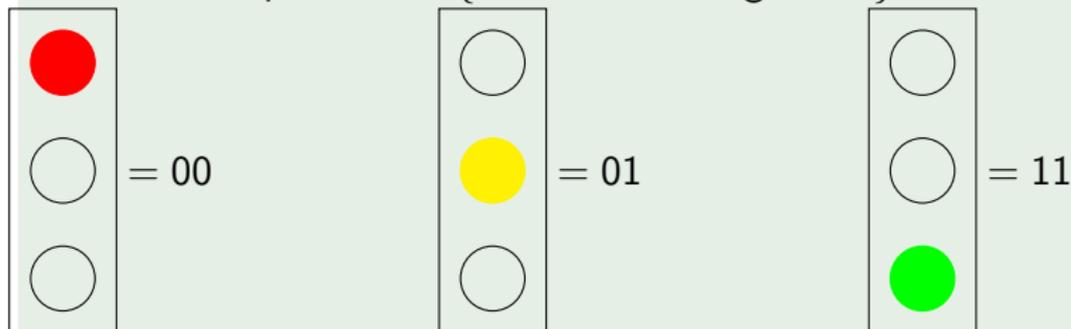
0  $\longrightarrow$  1

1  $\longrightarrow$  0

## Esempio

### Informazione

Un semaforo può essere {rosso, verde, giallo}.



### Elaborazione algoritmica

Se il semaforo è verde o giallo attraversa  
(= 1), altrimenti non attraversare (= 0)

00	→	0
01	→	1
11	→	1

# Rappresentazione binaria dell'informazione

Le componenti hardware sono spesso *bistabili*: Rappresentare l'informazione con solo **due** simboli (rappr. **binaria**)

0 1

00 01 10 11

000 001 010 011 100 101 110 111

0000.0001.0010.0011.0100.0101.0110.0111.1000.1001.1010.1011.1100.1101.1110.1111

0 1 2 3 4 5 6 7 8 9 A B C D E F

- 1 cifra binaria viene detta **bit**
- 1 bit è la quantità di informazione che permette di discernere fra due alternative
- con  $n$  bit è possibile distinguere  $2^n$  informazioni diverse, altrimenti equiprobabili
- Le stringhe di bit vengono spesso scritte in maniera più compatta usando i 16 simboli 0123456789ABCDEF, ognuno dei quali rappresenta un gruppo di 4 bit (notazione esadecimale)

## Esempi

- Quanti bit servono per codificare le sette note musicali?
- Quanti bit servono per distinguere gli abitanti del pianeta (circa 7 miliardi di persone)?
- Quanti bit servono per codificare i numeri da 0 a 255?
- Supponiamo che i nanetti di Biancaneve dicano sempre e solo la verità. Supponendo che inizialmente non abbia alcun sospetto specifico, quante domande sí/no dovrà fare Biancaneve per scoprire chi, fra i nanetti, s'è mangiato la torta?

## Codifica binaria di numeri

La rappresentazione abituale dei numeri usa dieci simboli, secondo una notazione posizionale (**numerazione decimale**)

$$1035 = 1 \times 10^3 + 0 \times 10^2 + 3 \times 10^1 + 5 \times 10^0$$

Possiamo fare la stessa cosa con qualsiasi numero di simboli, per esempio 2 (**numerazione binaria**)

$$(1035)_{10} = (10000001011)_2$$

# Trasformare da decimale a binario

Dato il numero che vogliamo convertire, dobbiamo:

- dividere per due e memorizzare il resto (il resto sarà 0 o 1)
- se il quoziente/quoto (il risultato della divisione precedente) non è uguale a zero, dividerlo per due e memorizzare il resto
- ripetere fintantoché il quoziente non diventa zero

Come rappresentare i numeri non interi? Come rappresentare i numeri negativi?

# Codifica dei caratteri

ASCII	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
ASCII	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
ASCII	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
ASCII	0	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1
0000	N <sub>U</sub>	S <sub>H</sub>	S <sub>X</sub>	E <sub>X</sub>	E <sub>T</sub>	E <sub>O</sub>	A <sub>K</sub>	B <sub>L</sub>	B <sub>S</sub>	H <sub>T</sub>	L <sub>F</sub>	V <sub>T</sub>	F <sub>F</sub>	C <sub>R</sub>	S <sub>O</sub>	S <sub>I</sub>
0001	D <sub>L</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	N <sub>K</sub>	S <sub>V</sub>	E <sub>Z</sub>	C <sub>N</sub>	E <sub>M</sub>	S <sub>B</sub>	E <sub>C</sub>	F <sub>S</sub>	G <sub>S</sub>	R <sub>S</sub>	U <sub>S</sub>
0010		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
0011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0101	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
0110	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0111	p	q	r	s	t	u	v	w	x	y	z	{		}	~	D <sub>T</sub>
1000	S <sub>O</sub>	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	I <sub>N</sub>	N <sub>L</sub>	S <sub>S</sub>	E <sub>S</sub>	H <sub>S</sub>	H <sub>J</sub>	V <sub>S</sub>	P <sub>D</sub>	P <sub>V</sub>	R <sub>I</sub>	S <sub>Z</sub>	S <sub>Z</sub>
1001	D <sub>C</sub>	P <sub>1</sub>	P <sub>2</sub>	S <sub>E</sub>	C <sub>C</sub>	M <sub>M</sub>	S <sub>P</sub>	F <sub>P</sub>	O <sub>S</sub>	O <sub>O</sub>	O <sub>A</sub>	C <sub>S</sub>	S <sub>T</sub>	O <sub>S</sub>	P <sub>M</sub>	A <sub>P</sub>
1010	A <sub>O</sub>	i	ç	£		¥		\$	''	©	♀	«	¬	-	®	—
1011	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
1100	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
1101	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
1110	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
1111	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Figura 6.6 Extended ASCII, l'American Standard Code for Information Interchange

# File di testo

- Un file di testo contiene solo caratteri (ASCII, ASCII esteso, UNICODE, ...)
- Un file prodotto da un word processor, generalmente **non** è un file di testo ([testo formattato](#))
- (Un file HTML è un file di testo, che viene interpretato dal browser per visualizzare un testo formattato)

## Codifica di colori

Uno dei metodi piú usati è quello cosiddetto **RGB** (Red, Green, Blue).

- Ogni colore è la composizione di 3 livelli 0–255: uno si riferisce al rosso, uno al verde, uno al blu;
- Il colore finale è ottenuto secondo il “modello additivo” dei colori primari (p.es. rosso + verde = giallo)
- Ogni livello è codificato indipendentemente come un numero (< 256 quindi bastano 8 bit)
- Gli 8 bit vengono generalmente scritti come due cifre esadecimali (204  $\rightsquigarrow$  CC)

Esempi: <http://www.visibone.com/colorlab/>