UNIVERSITÀ DEGLI STUDI DI SALERNO

Università di Salerno
Dipartimento di
Ingegneria Industriale

Fondamenti di Informatica

Linguaggi, Codifica e Rappresentazione dell'Informazione

Prof. Raffaele Pizzolante

A.A. 2016/17

Cosa abbiamo visto la volta scorsa

- Gli elaboratori sono strumenti per risolvere (o aiutare a risolvere) problemi basati sulle informazioni
- Ma come ciò avviene?
  1. Abbiamo bisogno di **codificare** e memorizzare opportunamente **dati** e **informazioni**
  2. Abbiamo bisogno di **impartire** le giuste **istruzioni per risolvere** correttamente **problemi**

Cosa vedremo oggi

- Gli elaboratori sono strumenti per risolvere (o aiutare a risolvere) problemi basati sulle informazioni
- Ma come ciò avviene?
  1. **Abbiamo bisogno di codificare e memorizzare opportunamente dati e informazioni**
  2. Abbiamo bisogno di **impartire** le giuste **istruzioni per risolvere** correttamente **problemi**

Che Lingua parla l'Elaboratore?

- Come rendere dati e informazioni comprensibili ad un elaboratore?
- **Informazioni** e **dati** per essere trattati da un elaboratore devono essere opportunamente **codificati**

Linguaggio

- **Alfabeto**
  - Collezione di simboli grafici, aventi di solito un ordine ben preciso, che servono a rappresentare le parole di una lingua
- **Vocabolario (o lessico)**
  - Insieme delle parole ammissibili di una lingua
- **Grammatica**
  - Insieme di regole utili alla corretta costruzione di frasi, parole, …
    - Il termine si riferisce allo studio delle suddette regole
- **Semantica**
  - Studia il significato delle parole (semantica lessicale), degli insiemi delle parole, delle frasi (semantica frasale) e dei testi

La Funzione dei Linguaggi

- I linguaggi (verbali, non verbali, orali, scritti, etc.) sono strumenti che contribuiscono a
  - **Rappresentare le informazioni**
    - Concetti, pensieri, emozioni, etc., vengono formalizzati attraverso i linguaggi per poter essere memorizzati, trasferiti ed elaborati
  - **Memorizzare le informazioni**
    - La scrittura
  - **Trasferire le informazioni**
    - La comunicazione
  - **Elaborare le informazioni**
    - Le deduzioni nella logica

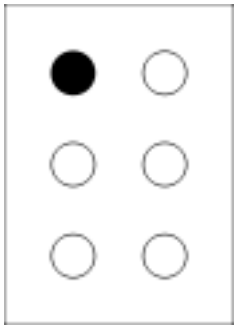
Problemi relativi ai Linguaggi

- **Accordo sui simboli**
  - a b c d e f g …
- **Accordo sul lessico**
  - casa, gatto, automobile, vado, …
- **Accordo sulla grammatica**
  - <soggetto verbo complemento>
- **Accordo sulla semantica**
  - La nonna chiude la porta (**OK**)
  - La porta chiude la nonna (**NO**)

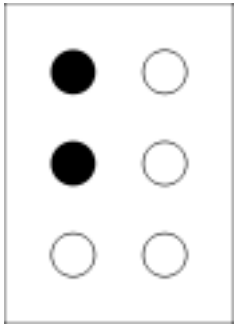
- **Accordo sulla codifica**
  - Regole per trasformare simboli, parole e frasi di un linguaggio in una nuova rappresentazione con possibilità di effettuare in maniera corretta anche l'operazione inversa
    - "a" in codice Morse (Samuel Morse, pittore e storico inglese) è " . – "
    - "b" in codice Morse è " – . . . "

Codice Braille

- Lettera “a”



- Lettera “b”



Codifica dei Numeri

- **Linguaggio di partenza**
  - I numeri
- **Codifica 1**
  - Numerazione decimale
    - 5, 45, 670
- **Codifica 2**
  - Numerazione binaria
    - 101, …

I Linguaggi Naturali: Ambiguità

- Per comunicare tra loro gli uomini hanno sviluppato i **linguaggi naturali**
  - Italiano, inglese, francese, etc
- Una **caratteristica negativa** di tali linguaggi è la loro inerente **ambiguità**
  - Una qualsiasi **frase** formulata è potenzialmente **polisemica**
  - Il significato che viene dato alla frase da chi riceve il messaggio può essere diverso da quello datogli dal mittente

I Linguaggi Naturali nella Comunicazione con i Calcolatori

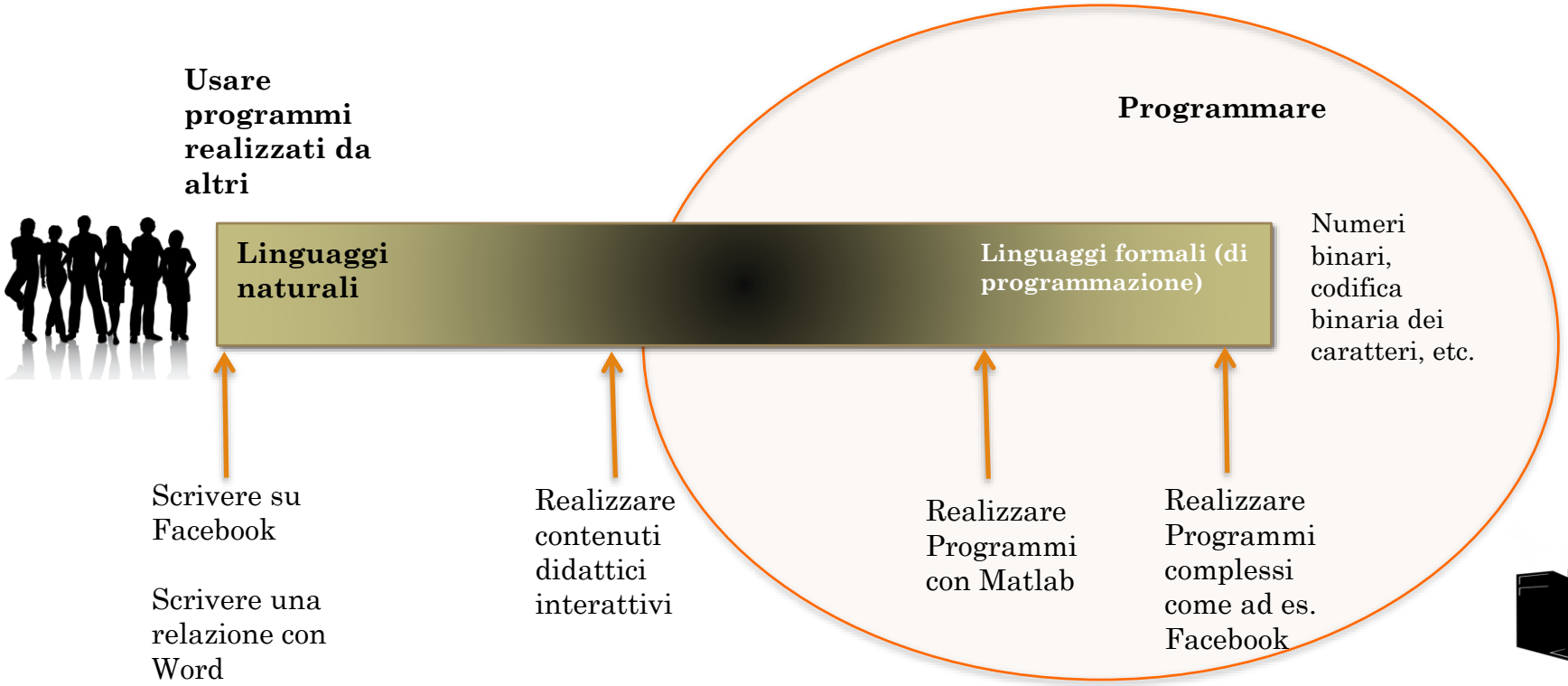
- Per comunicare con un elaboratore, l'**ambiguità** dei **linguaggi naturali** rappresenta un grosso problema

I Linguaggi Naturali nella Comunicazione con i Calcolatori

- Risulta quindi necessaria la definizione di un **Linguaggio** più **Formale**, che permetta di
  - Individuare un **alfabeto**, ovvero un elenco finito di simboli
  - Definire un insieme di **regole** sintattiche, che specificano come i simboli dell'alfabeto possono essere combinati tra loro per creare frasi ben formate all'interno del linguaggio stesso (**grammatica**)
  - Attribuire un significato non ambiguo alle frasi del linguaggio (**semantica**)

Usare e Programmare il computer

- I Programmi (o software) risolvono problemi specifici con approccio basato sulle informazioni e vengono eseguiti dai computer

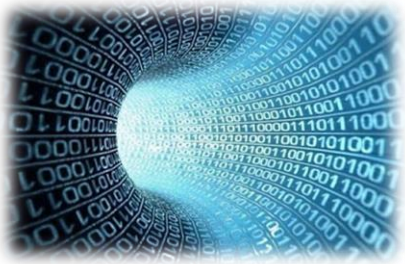


Rappresentazione dell'Informazione: Accordo sui Simboli

- L'informazione è rappresentata dai **dati**, che a loro volta sono **espressi in** forma di **simboli**
- La **stessa informazione** può essere **codificata con simboli e modalità diverse**
- **Esempio:** Stessa informazione rappresentata in più modi:
  - 1963 -> simboli "0", "1", "2", …
  - MCMLXIII -> simboli della codifica romana
  - Millenovecentosessantatre -> rappresentazione testuale
  - …

Rappresentazione dell'informazione nei calcolatori – 1/2

- Consideriamo un alfabeto ridotto che contiene solo i simboli "0" e "1"
- Un **bit** (contrazione di **bi**nary digi**t**) è un simbolo scelto sull'alfabeto composto dai simboli
  - 0
  - 1
- Nei calcolatori **ogni elemento** (numeri, testo, audio, video, istruzioni, etc) viene **rappresentato** esclusivamente con **sequenze di bit**
- I **dati** e le **istruzioni** vengono **codificati** con **sequenze di bit**



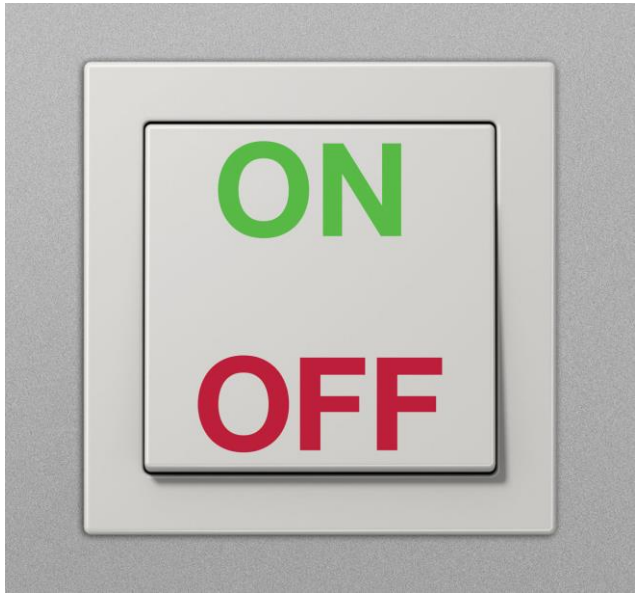
Codifica Binaria

- **Alfabeto binario**
  - Usiamo solo due simboli 0, 1 (bit)

- **Problema**
  - Assegnare una sequenza di bit **univoca a tutti gli oggetti** in un insieme predefinito

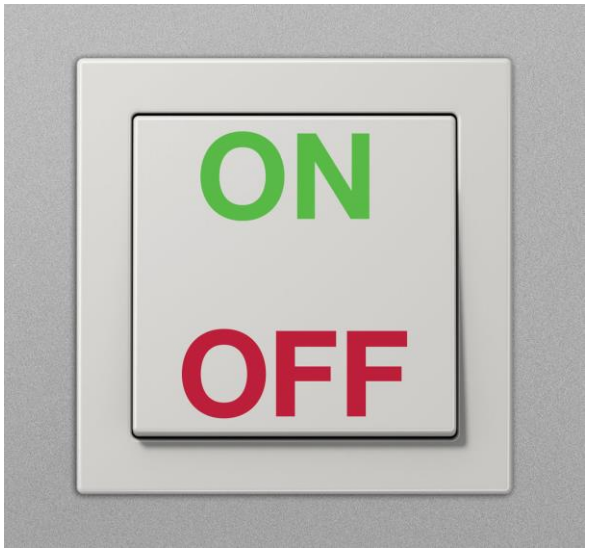
Esempio semplice – codifica binaria: l'*interruttore*

- Un interruttore ha due sole possibilità:
  - Acceso (ON)
  - Spento (OFF)
- L'informazione sullo stato dell'interruttore corrisponde alla scelta fra due sole alternative



Esempio semplice: l'*interruttore*

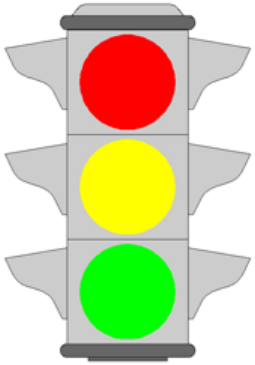
- **1** bit basta per rappresentare lo stato dell'interruttore
  - Interruttore Acceso (ON) => **1**
  - Interruttore Spento (OFF) => **0**



Esempio semplice: l'*interruttore*

Esempio: il *semaforo*

Il semaforo ha **tre** possibilità



Esempio: il *semaforo*

E’ possibile utilizzare una sequenza di **tre** bit per rappresentare ciascuna alternativa

0

0

1

001

0

1

0

010

1

0

0

100

Esempio: il *semaforo*

E’ possibile ottimizzare l’uso del numero di bit ed usare una sequenza di soli **due** bit per rappresentare ciascuna alternativa

11

10

00

Esempio: il *semaforo*

E’ possibile ottimizzare l’uso del numero di bit ed usare una sequenza di soli **due** bit per rappresentare ciascuna alternativa

11

10

00

01 -> non usato

Esempio: il *semaforo*

- Rappresentazione degli stati di un semaforo mediante bit

3 bit

Stato	Codifica
ROSSO	1 0 0
VERDE	0 0 1
GIALLO	0 1 0

2 bit

Stato	Codifica
ROSSO	0 0
VERDE	0 1
GIALLO	1 0

Combinazioni di Bit

Bit a disposizione	Le combinazioni	Il numero di combinazioni
1	0, 1	$2 = 2^1$
2	00, 01, 10, 11	$4 = 2^2$
3	000, 001, 010, 011, 100, 101, 110, 111	$8 = 2^3$
4	0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111,1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111	$16 = 2^4$
5	00000, 00001, 00010, …	$32 = 2^5$
…	…	…

Codifica Binaria

- **Quanti oggetti posso rappresentare con *k* bit?**
  - *k* bit → 2^k *oggetti*

- **Quanti bit mi servono per codificare *N* oggetti?**
  - $\lceil log_2 N \rceil$ bit
    - (intero superiore)

- **Esempio**
  - Quanti bit servono per rappresentare 5 oggetti?
    - $\lceil log_2 5 \rceil = 3$ bit (**NOTA:** $\log_2 5 = 2.3219$)

Codifica di un'Informazione: Giorni della Settimana (*Esercizio per casa*)

- Assegnare un codice binario univoco a tutti i giorni della settimana

Codifica dei Caratteri – 1/5

- **Problema:** è possibile applicare queste idee alla rappresentazione di informazione più complessa, ad esempio di un testo?
  - Un testo è rappresentato attraverso una successione di caratteri
  - Ogni carattere viene scelto all'interno di un insieme finito di simboli (alfabeto)

Codifica dei Caratteri – 2/5

- Con 8 bit, è possibile rappresentare la scelta fra 256 alternative diverse (2^8=256)
  - Da 00000000… a 11111111
  - Passando per tutte le combinazioni intermedie (00000001, 00000010, …)
- Nel caso del testo, possiamo far corrispondere diverse combinazioni di 8 bit a caratteri diversi

Ogni singolo CARATTERE viene codificato con una combinazione di 8 bit

Codifica dei Caratteri – 3/5

- Ad esempio:
  - 00000000 -> A
  - 00000001 -> B
  - 00000010 -> C
  - 00000011 -> D
  - 00000100 -> E
- …. e così via

Codifica dei Caratteri – 4/5

A**merican **S**tandard **C**ode for **I**nformation **I**nterchange – **ASCII (Codice Standard Americano per lo Scambio di Informazioni) è un codice standard per la codifica dei caratteri

ASCII Code: Character to Binary

0	0011 0000	O	0100 1111	m	0110 1101
1	0011 0001	P	0101 0000	n	0110 1110
2	0011 0010	Q	0101 0001	o	0110 1111
3	0011 0011	R	0101 0010	p	0111 0000
4	0011 0100	S	0101 0011	q	0111 0001
5	0011 0101	T	0101 0100	r	0111 0010
6	0011 0110	U	0101 0101	s	0111 0011
7	0011 0111	V	0101 0110	t	0111 0100
8	0011 1000	W	0101 0111	u	0111 0101
9	0011 1001	X	0101 1000	v	0111 0110
A	0100 0001	Y	0101 1001	w	0111 0111
B	0100 0010	Z	0101 1010	x	0111 1000
C	0100 0011	a	0110 0001	y	0111 1001
D	0100 0100	b	0110 0010	z	0111 1010
E	0100 0101	c	0110 0011	.	0010 1110
F	0100 0110	d	0110 0100	,	0010 0111
G	0100 0111	e	0110 0101	:	0011 1010
H	0100 1000	f	0110 0110	;	0011 1011
I	0100 1001	g	0110 0111	?	0011 1111
J	0100 1010	h	0110 1000	!	0010 0001
K	0100 1011	I	0110 1001	'	0010 1100
L	0100 1100	j	0110 1010	"	0010 0010
M	0100 1101	k	0110 1011	(	0010 1000
N	0100 1110	l	0110 1100	)	0010 1001
				space	0010 0000

Codifica dei Caratteri - 5/5

- **Soluzione:** una parola (o più parole) sarà rappresentata dal computer come una successione di gruppi di 8 bit

O	G	G	I		P	I	O	V	E
01001111	01000111	01000111	01001001	00100000	01010000	01001001	01001111	01010110	01000101

La Codifica dei Numeri

- **Obiettivo**
  - Codifica in binario dei numeri per favorire l'elaborazione da parte dei calcolatori
- **Vincoli**
  - Codifica e decodifica devono essere definite in maniera tale da poter essere compiute in maniera automatica
- **Problema**
  - Deve essere possibile codificare tutti i numeri
    - 0, 1, 2, 3, …
    - -1, -2, -3, …
    - -12.4, -2.004, 0.56, 134.89, …
  - …in sequenze binarie
    - 0000000, 000001, 000010, …

Sistemi di Numerazione Posizionale – 1/4

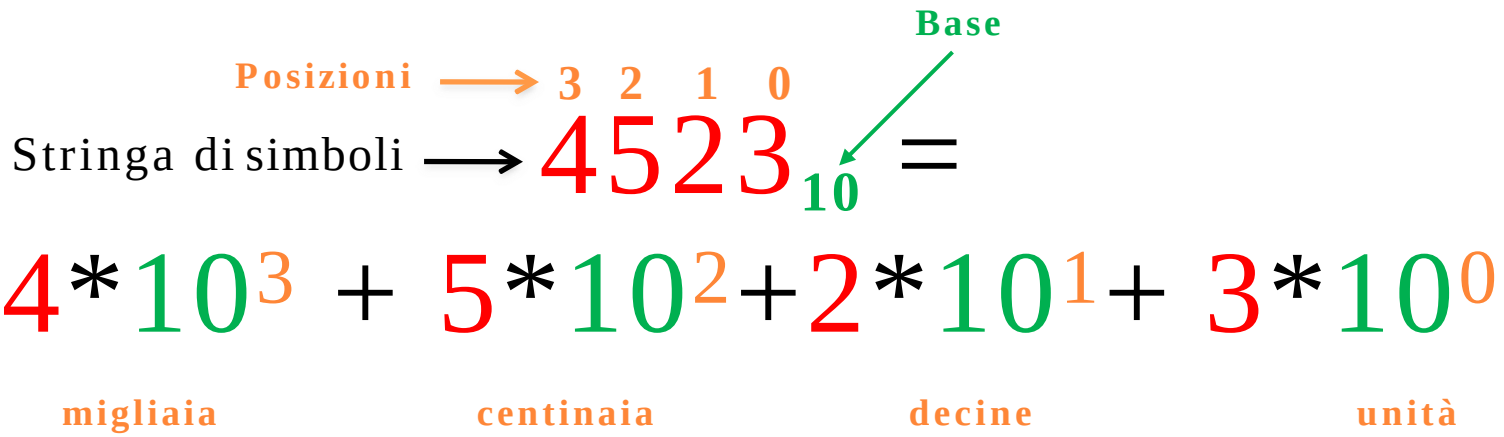
- Il nostro sistema di **numerazione**
  - Utilizza una notazione **posizionale** ed è in **base 10**
  - L'alfabeto utilizzato è l'insieme dei simboli {0, 1, 2, …, 9}
  - Non è l'unico sistema possibile

Sistemi di Numerazione Posizionale – 2/4

- **3251**
  - **3** unità di migliaia
  - **2** centinaia
  - **5** decine
  - **1** unità
- **814763**
  - **3** unità
  - **6** decine
  - **7** centinaia
  - **4** unità di migliaia
  - **1** decina di migliaia
  - **8** centinaia di migliaia

Sistemi di Numerazione Posizionale – 3/4

- Essendo posizionale, il valore di una "sequenza" di simboli viene calcolata assegnando dei "pesi" ad ogni simbolo a seconda della sua posizione



Sistemi di Numerazione Posizionale – 4/4

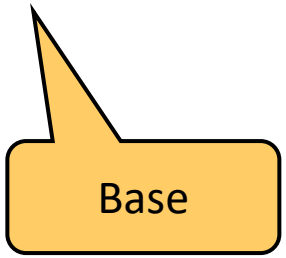
- Concetto di **base di rappresentazione B**
- Rappresentazione del **numero** come **sequenza di simboli**, detti **cifre**
  - Appartenenti ad un alfabeto composto da **B** simboli distinti
    - Ogni simbolo rappresenta un valore fra 0 e **B**-1
- Il valore di un numero *v* espresso in questa notazione è ricavabile
  - A partire dal valore rappresentato da ogni simbolo
  - Pesato in base alla posizione che occupa nella sequenza

Sistemi di Numerazione più Diffusi

Sistema	Base	Simboli	Usato dagli umani?	Usato dai computer?
Decimale	10	0, 1, … 9	Si	No
Binario	2	0, 1	No	**Si**
Ottale	8	0, 1, … 7	No	No
Esadecimale	16	0, 1, … 9, A, B, … F	No	No

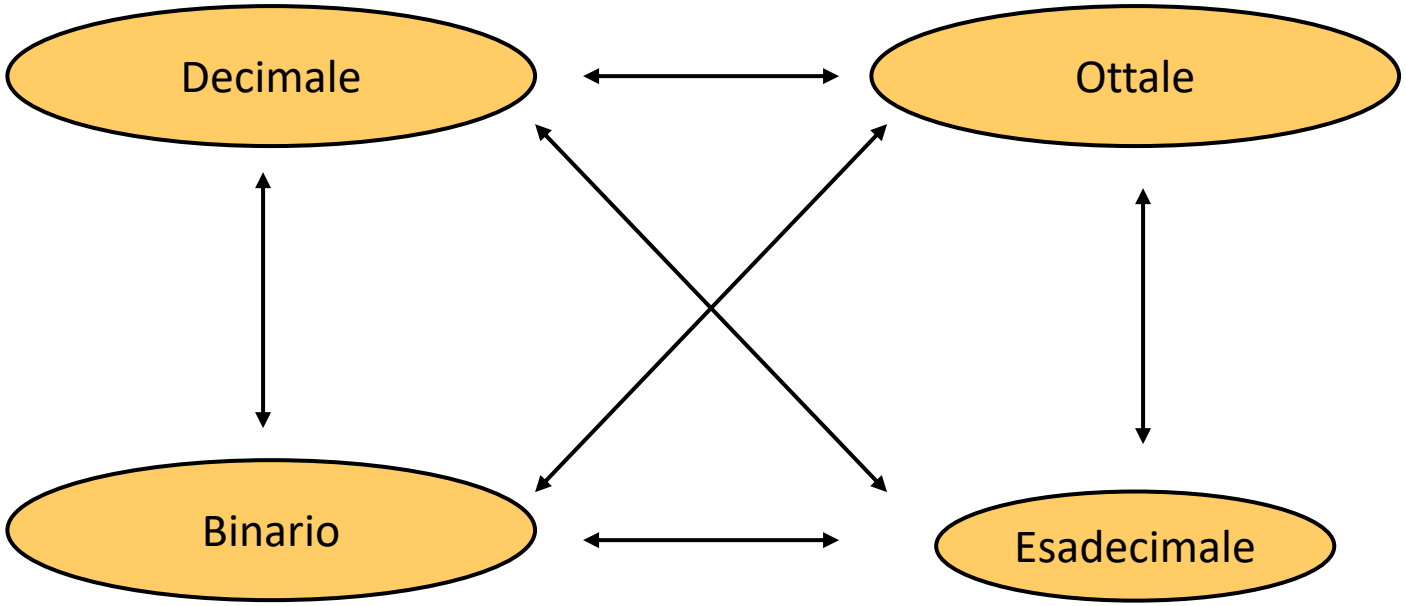
Esempio

$$25_{10} = 11001_{2} = 31_{8} = 19_{16}$$

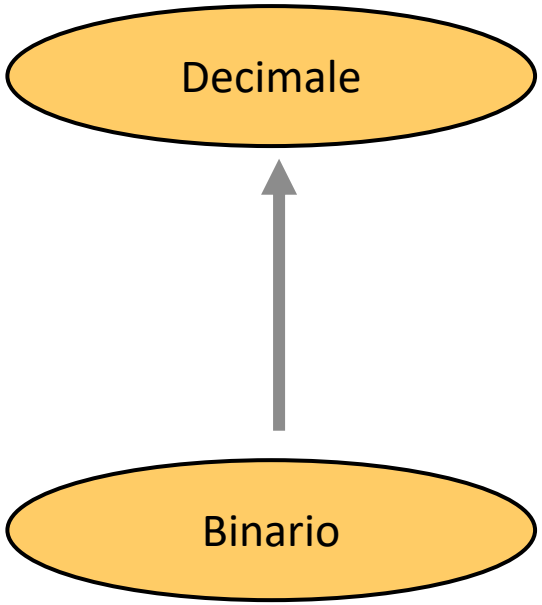


Conversioni tra Basi (più Diffuse)

- Le possibilità



Da Binario a Decimale



Da Binario a Decimale: Tecnica

- Data una sequenza di bit
- Moltiplica ciascun bit per 2^n, dove *n* è il "peso" del bit
  - Il peso è dato dalla posizione del bit, **a partire da 0 sulla destra**
- Somma i risultati

Da Binario a Decimale: Tecnica

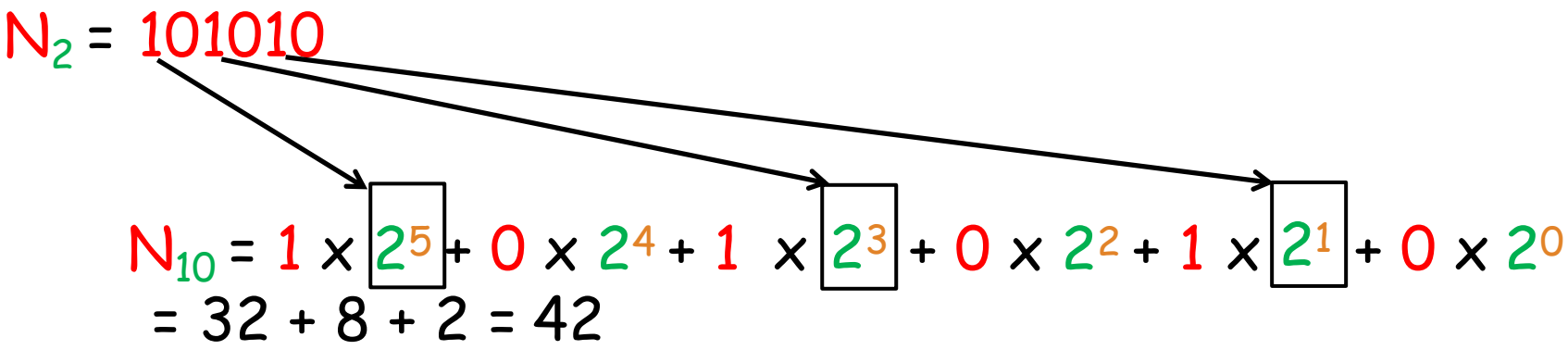
- Moltiplica ciascun bit per 2^n, dove *n* è il "peso" del bit
  - Il peso è dato dalla posizione del bit, a partire da 0 sulla destra
- Somma i risultati

101011_2 =>

Bit in posizione "0"

$$
\begin{aligned}
1 \times 2^0 &= 1 \; + \\
1 \times 2^1 &= 2 \; + \\
0 \times 2^2 &= 0 \; + \\
1 \times 2^3 &= 8 \; + \\
0 \times 2^4 &= 0 \; + \\
1 \times 2^5 &= 32 \\
\hline
&= 43_{10}
\end{aligned}
$$

Da Binario a Decimale: Esempio

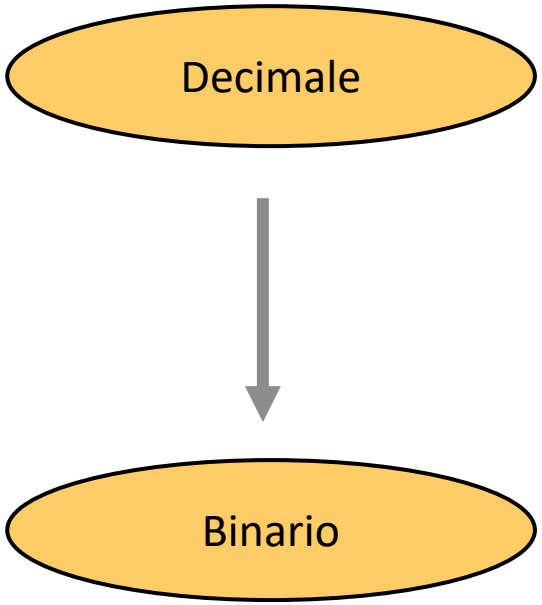


Da Binario a Decimale: Altri Esempi*

- $10011010 = 1*2^7 + 0*2^6 + 0*2^5 + 1*2^4 + 1*2^3 + 0*2^2 + 1*2^1 + 0*2^0$
  $= 2^7 + 2^4 + 2^3 + 2^1$
  $= 128 + 16 + 8 + 2$
  $= 154$

- $00101001 = 0*2^7 + 0*2^6 + 1*2^5 + 0*2^4 + 1*2^3 + 0*2^2 + 0*2^1 + 1*2^0$
  $= 2^5 + 2^3 + 2^0$
  $= 32 + 8 + 1$
  $= 41$

Da Decimale a Binario



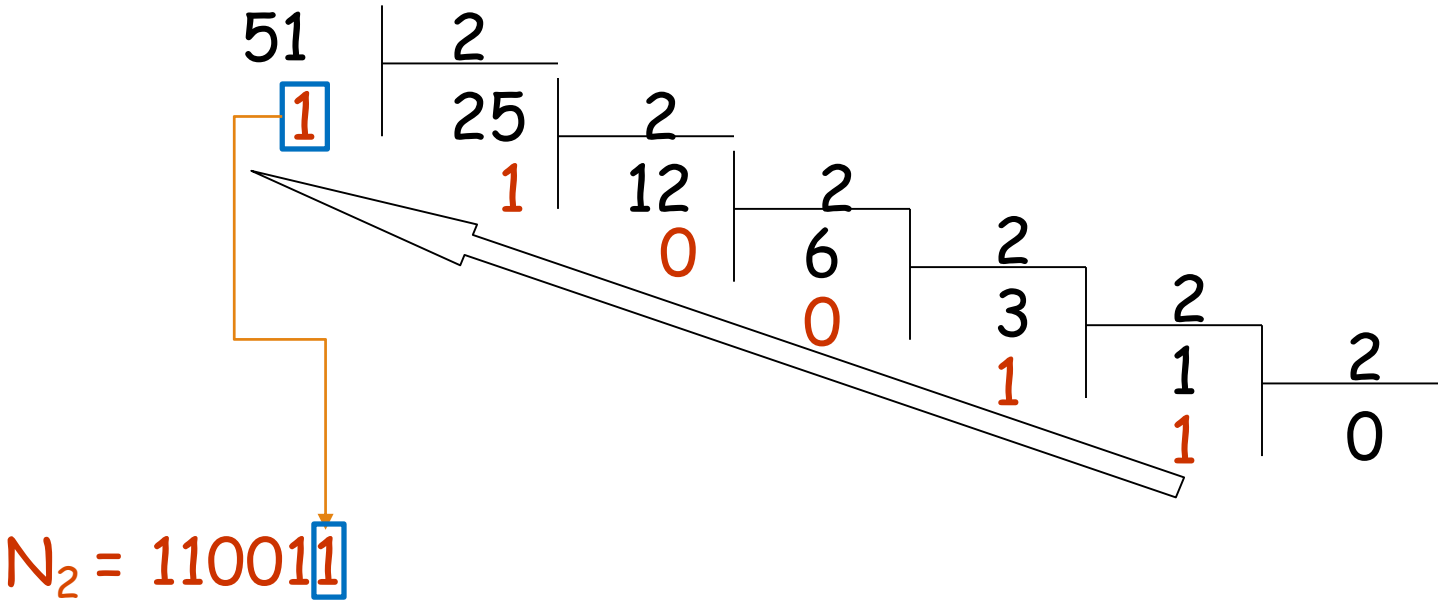
Da Decimale a Binario: Tecnica

- Dividi per due e tieni traccia del resto (**divisione euclidea** o **con resto**)
- Il **primo resto** è il **bit in posizione 0** (LSB, least-significant bit)
- Il **secondo resto** è il **bit in posizione 1**
- E così via…

Da Decimale a Binario: Esempio

(Da decimale a binario)

$N_{10} = 51$ $N_2 = ???$



Rappresentazione degli Interi: "Modulo e Segno"

- N = 0, +1, -1, +2, -2, +3, -3,…
- Come possiamo rappresentare il segno di un numero?
- **Idea:** Aggiungiamo un ulteriore bit che poniamo a
  - **1** se il numero è negativo
  - **0** altrimenti

Esempio

$N_{10} = +14 \quad N_2 = 01110$

$N_{10} = -14 \quad N_2 = 11110$

Rappresentazione degli Interi: “Modulo e Segno”

- **Alfabeto binario**
  - Anche il segno è rappresentato da 0 o 1
  - Indispensabile indicare il numero k di bit utilizzati
- **Modulo e segno** (rappresentazione con **k** bit)
  - **1 bit** di **segno** (**0 positivo**, **1 negativo**)
  - **k-1 bit** di **modulo**

Rappresentazione degli Interi: “Modulo e Segno”

- **Modulo e segno** (rappresentazione con **k** bit)
  - **1 bit** di **segno** (**0 positivo**, **1 negativo**)
  - **k-1 bit** di **modulo**

Esempio

- **k = 4**

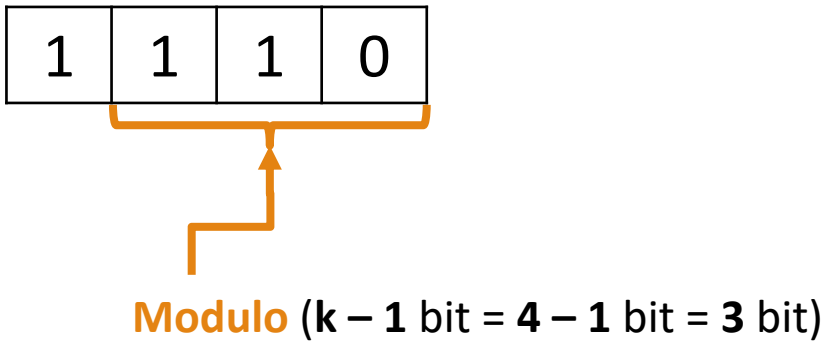
1	1	1	0

Rappresentazione degli Interi: "Modulo e Segno"

- **Modulo e segno** (rappresentazione con **k** bit)
  - **1 bit** di **segno** (**0 positivo**, **1 negativo**)
  - **k-1 bit** di **modulo**

Esempio

- **k = 4**

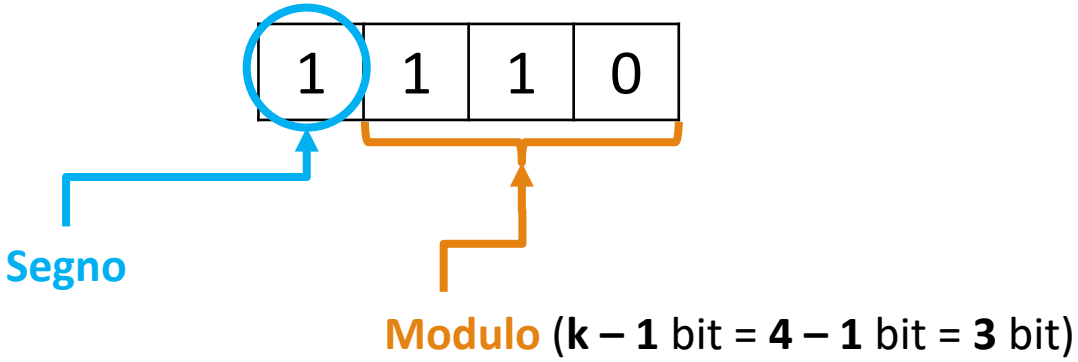


Rappresentazione degli Interi: "Modulo e Segno"

- **Modulo e segno** (rappresentazione con **k** bit)
  - **1 bit** di **segno** (**0 positivo**, **1 negativo**)
  - **k-1 bit** di **modulo**

Esempio

- **k = 4**

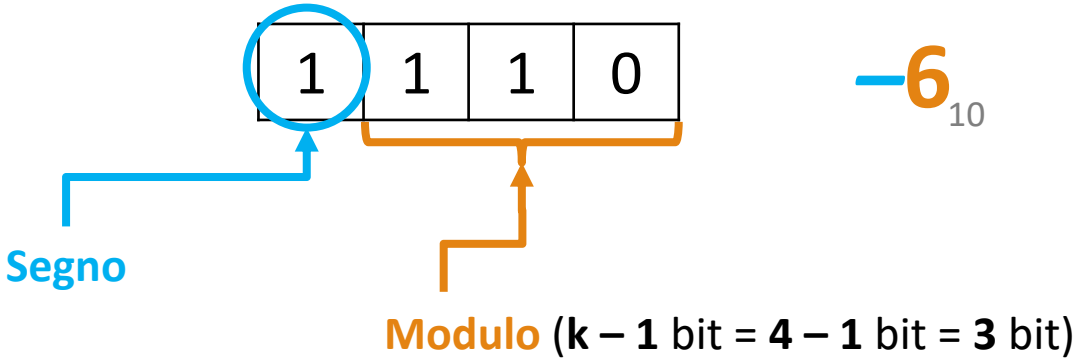


Rappresentazione degli Interi: "Modulo e Segno"

- **Modulo e segno** (rappresentazione con **k** bit)
  - **1 bit** di **segno** (**0 positivo**, **1 negativo**)
  - **k-1 bit** di **modulo**

Esempio

- **k = 4**



-6_{10}

Rappresentazione degli Interi: "Modulo e Segno"

- Con **k bit** è possibile rappresentare valori **$-2^{k-1}+1$ a $2^{k-1}-1$**
  - *Esempi*
    - **4 bit** → valori che vanno da **-7 a +7**
    - **8 bit** → valori che vanno da **-127 a + 127**
  - **Osservazione**
    - due rappresentazioni dello 0
      - Con **4 bit** sono **+**0_{10} = **0**000_{ms} **-**0_{10} = **1**000_{ms}

Rappresentazione degli Interi: “Modulo e Segno”: Limiti sui Numeri Rappresentabili

- **4 bit** a disposizione
  - Possiamo rappresentare da 0000 a 0111 e da 1000 a 1111, in decimale da 0 a 7 e da 0 a -7
- **5 bit** a disposizione
  - Possiamo rappresentare da 00000 a 01111 e da 10000 a 11111, in decimale da 0 a 15 e da 0 a -15
- …
- Con **k bit** a disposizione possiamo rappresentare **numeri da 0 a $2^{k-1} - 1$** e **da $-(2^{k-1} - 1)$ a 0**

Numeri Interi in Complemento a Due

- **Alfabeto binario**
  - Anche il segno è rappresentato da 0 o 1
  - Indispensabile indicare il numero k di bit utilizzati
- **Complemento a due**
  - **X** corrisponde al **binario naturale** di $\mathbf{2^k + X}$
    - $+6_{10} \rightarrow 2^4 + 6 = 22 \rightarrow [1]0110 \rightarrow 0110_{C2}$
    - $-6_{10} \rightarrow 2^4 - 6 = 10 \rightarrow [0]1010 \rightarrow 1010_{C2}$
  - Si rappresentano i **valori da -2^{k-1} a $2^{k-1}-1$**

Numeri Interi in Complemento a Due

- **Esempi**
  - Con **4 bit** i valori vanno da **-8** a **+7**
  - Con **8 bit** i valori vanno da **-128** a **+127**
  - Con **32 bit** i valori vanno da **-2'147'483'648** a **+2'147'483'647**

- **Osservazione:** una sola rappresentazione dello 0
  - Con 4 bit è $+0_{10} = 0000_{C2}$ mentre $1000_{C2} = -8_{10}$

Complemento a Due
Un metodo Alternativo

- **Metodo Alternativo**
- Effettuare il **complemento** di ogni bit di **X**, poi aggiungere 1
  1. Codifica binaria di $+6_{10}$ => 0110_2 (N.B. ci vogliono 4 bit)
  2. Complemento di tutti i bit => 1001_{C2} (corrisponderebbe a -7_{10})
  3. Aggiungere 1 => 1010_{C2} (che corrisponde a -6_{10})

Complemento a Due: Notazione Posizionale

- Se si considera la notazione posizionale, si può notare che nella rappresentazione binaria in complemento a due **il valore si può ottenere** anche **associando alla cifra più significativa** un **peso negativo**, mentre **tutte le altre cifre mantengono** il **peso** originario **positivo**
- Il valore di un numero $c_{n-1}c_{n-2}\ldots c_1c_0$ può essere espresso in complemento a due come
  - $-c_{n-1}\times 2^{n-1} + c_{n-2}\times 2^{n-2} + \ldots + c_1\times 2^1 + c_0\times 2^0$
- **Esempio**
  - $0101_{C2} = -0\times 2^3+1\times 2^2+0\times 2^1+1\times 2^0 = 5_{10}$
  - $1011_{C2} = -1\times 2^3+0\times 2^2+1\times 2^1+1\times 2^0 = -5_{10}$

Complemento a Due: Estensione del Segno

- **Estensione del "segno"**
  - I valori positivi iniziano con 0, quelli negativi con 1
  - Data la rappresentazione di un numero su k bit, la rappresentazione dello stesso numero su k+1 bit si ottiene aggiungendo (a sinistra) un bit uguale al primo
- **Esempi**
  - Rappresentazione di -6 su 4 bit = 1010
  - Rappresentazione di -6 su 5 bit = 11010
  - Rappresentazione di -6 su 8 bit = 11111010

Esercizi

- Scrivere in binario semplice i seguenti numeri in base 10
  - 5310
  - 21110
- Scrivere in binario semplice su 7 bit il numero 13_{10}
- Scrivere in modulo e segno su 7 bit il numero 13_{10}
- Scrivere in modulo e segno su 7 bit il numero -13_{10}
- Scrivere in modulo e segno su 5 bit il numero 17_{10}

Soluzione Esercizi

- Scrivere in binario semplice su 7 bit il numero 13_{10}
  - 0001101
- Scrivere in modulo e segno su 7 bit il numero 13_{10}
  - 0001101
- Scrivere in modulo e segno su 7 bit il numero -13_{10}
  - 1001101
- Scrivere in modulo e segno su 5 bit il numero 17_{10}
  - 10001, In modulo e segno è -1_{10}
  - RISPOSTA: Non è possibile. Ho bisogno di almeno 6 bit (010001)

Esercizi svolti

- $53 = 32 + 16 + 4 + 1$
  $= 2^5 + 2^4 + 2^2 + 2^0$
  $= 1*2^5 + 1*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 1*2^0$
  = 110101 in binario

- $211 = 128 + 64 + 16 + 2 + 1$
  $= 2^7 + 2^6 + 2^4 + 2^1 + 2^0$
  $= 1*2^7 + 1*2^6 + 0*2^5 + 1*2^4 + 0*2^3 + 0*2^2 +$
  $1*2^1 + 1*2^0$
  = 11010011 in binario

Esercizi per casa

- Scrivere in binario semplice su 7 bit il numero 11_{10}
- Scrivere in modulo e segno su 8 bit il numero 25_{10}
- Scrivere in modulo e segno su 7 bit il numero -12_{10}
- Scrivere in modulo e segno su 5 bit il numero 20_{10}
- Scrivere in complemento a due su 7 bit il numero 15_{10}
- Scrivere in binario semplice i seguenti numeri in base 10
  - 2752
  - 5184

Riferimenti

- **Libro di testo**
  - Capitolo 2
    - Paragrafi 2, 2.1 [**NO approfondimento 2.4**], 2.2, 2.3, 2.4, 2.5 [**NO approfondimento 2.5**]