Fondamenti Teorici

Antonio Pescapè e Marcello Esposito

Parte Terza

v1.0

Agenda

- Concetto di Informazione
- Codifica
- Rappresentazione dei numeri

Informazione

Il numero di telefono di casa di Anna è 0817651831

Informazione

Il numero del telefono cellulare di Anna è 3337654321

Informazione

Il numero di serie del contratto di lavoro di Anna è 3337654321

Informazione

- Attributo (significato):

  il numero di telefono di casa di Anna

- Valore:

  0817654321

- Elemento spesso implicito: l'insieme a cui appartiene il valore (Tipo)

  sequenza di cifre

- Convenzione per la manipolazione del valore: codifica

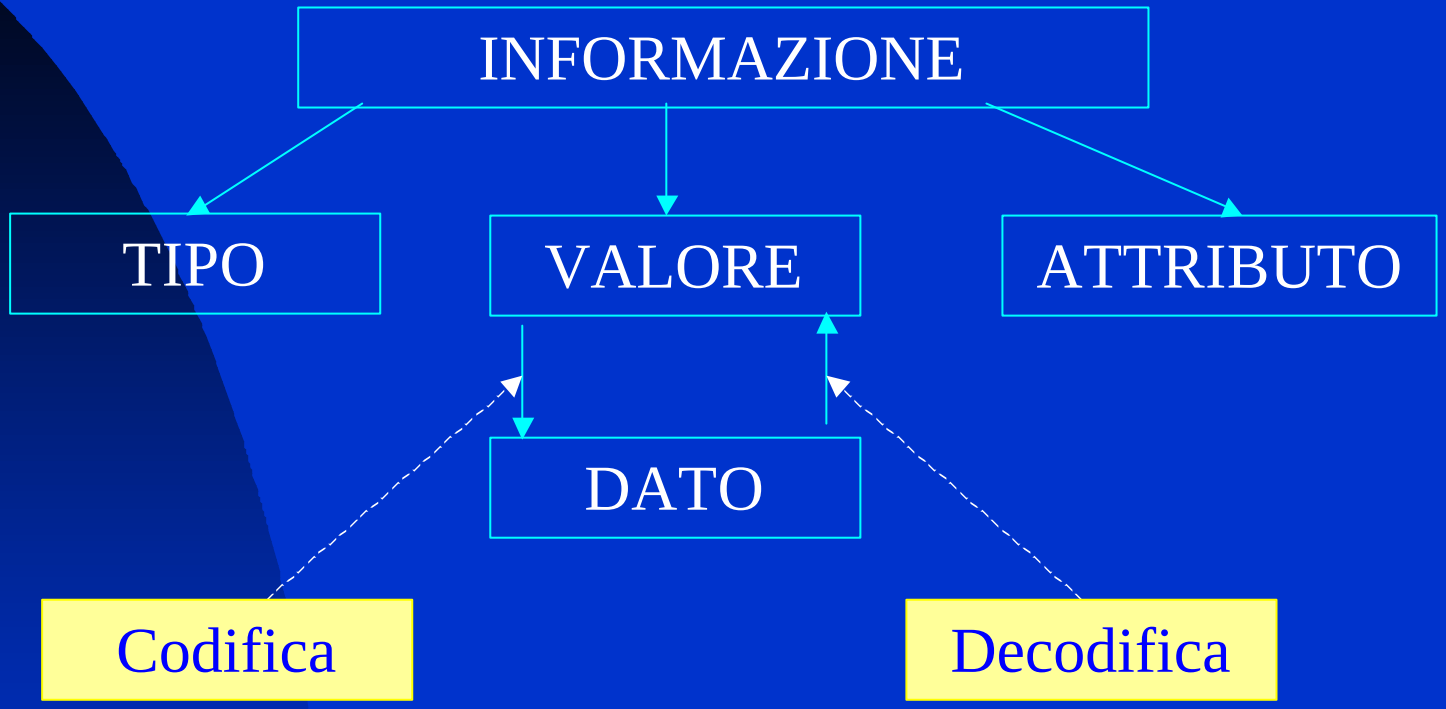
Informazione

L'INFORMAZIONE è quindi caratterizzata dalla tripla:

{TIPO, ATTRIBUTO, VALORE}

- La soluzione dell'equazione di I grado è 4,5
- Il numero di clienti è 1000
- Il prezzo delle scarpe è 80.000 lire
- La data di nascita di Marco è 01.01.01

Informazione, dato e codifica



Codifica delle informazioni

- *Rappresentazione* di informazioni appartenenti a un insieme finito *D*
- Funzione *iniettiva* dall'insieme *D* (*dominio*) a un insieme direttamente manipolabile *R* (*codominio*)

$$c : D \rightarrow R$$

$$|R| \geq |D|$$

- La funzione *c* è detta *codifica* o *rappresentazione* delle informazioni appartenenti a *D*

Sul piano formale

- Alfabeto origine T : il dato da rappresentare appartiene ad un tipo $T = (x_1,\ldots..,x_n)$
- Alfabeto in codice E : il dato è rappresentato mediante dati di tipo $E = (a_1,\ldots..,a_k)$
- Tabella Codice C : la codifica di T mediante E è un'applicazione C, detta *tabella codice*, che trasforma ciascun elemento $x_i \in T$ in una stringa di lunghezza l_i di elementi $a_i \in E$, detta *parola codice*

Codifica a lunghezza fissa

Se si pone $l_i = m$ = costante per tutti gli elementi di T, si ottiene una codifica a lunghezza fissa. In tal caso il codice è fissato facendo corrispondere a ciascun elemento $x_i \in T$ una delle k^m disposizioni con ripetizione dei k simboli di tipo E sugli m posti della stringa e quindi:

$$K^m >= N$$

da cui

$$m = \{[\log_k N]\}$$

Per codificare un dato di cardinalità N mediante un alfabeto di k simboli è necessaria una stringa di lunghezza minima m

l >= m **l lunghezza del codice**

Codifica a lunghezza fissa

Es. 1 : $T = (x_1,\ldots..,x_{20})$; $E = (0,1)$; $N = 20$; $k = 2$; $m = 5$;

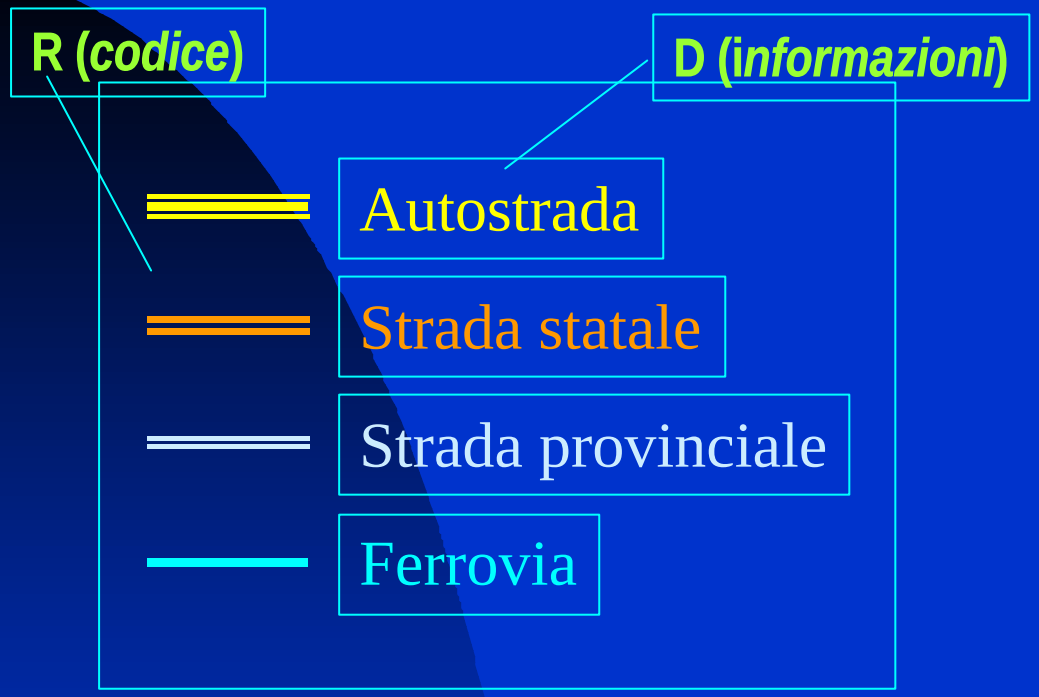
X1	00000	X5	10010
X2	01001	X6	00001
X3	01011	X7	01000
X4	01010	X8	01100
……	……	X20	…..

Codifica a lunghezza fissa

Es. 2 : **T = (a,b,c); E = (0,1); N = 3; k = 2; m = 2**

a	00
b	01
c	11

Esempi Comuni



Agrigento	AG
Alessandria	AL
Ancona	AN
Aosta	AO
……	…
Viterbo	VT

$26^2 \geq 103$

Come rappresentazione si può usare una sequenza (*stringa*) di *m* valori appartenenti a un insieme *K*:

$$|K^m| \geq |D|$$

Misura della quantità di Informazione

La quantità $m = \log_2 N$ **rappresenta la misura in bit della quantità di informazione in quanto, qualora il dato di cardinalità N fosse codificato in bit, occorrerebbero almeno** $\{[m]\}$ **bit**

Codifica a lunghezza fissa

- **CODICE INCOMPLETO** : se l = m ma N non è potenza di k, il codice viene detto incompleto e la differenza k^m-N fornisce il numero di parole codice non assegnate, cioè non associate ad alcun elemento dell'alfabeto origine. (vedi es. 1, 2)
- **CODICE RIDONDANTE** : si ottiene per l > m adoperando più caratteri dell'alfabeto in codice di quanto strettamente necessari.

  Questi codici vengono utilizzati per rilevare ed eventualmente correggere errori dovuti ad alterazioni del dato.

CODICE RIDONDANTE → **CODICE INCOMPLETO**

Codifica a lunghezza variabile

La lunghezza l_i del codice è funzione di ciascun elemento x_i :

$$l_i = f(x_i)$$

Proprietà fondamentale è quella che la (o le) parola più corta è individuata da un particolare codice che non si ritrova come sequenza iniziale in quelle più lunghe: è la lunghezza stessa che deve essere riconosciuta nel contesto della parola-codice.

L'uso di questo tipo di codifica è giustificato quando gli elementi del tipo T (alfabeto origine) non hanno tutti la stessa probabilità di occorrenza.

Codifica a lunghezza variabile

Dato l'insieme

$$T = (x_1,\ldots..,x_n)$$

Dette

$$p_1,\ldots..,p_n$$

le probabilità di occorrenza (frequenza) dei rispettivi elementi di T

la lunghezza l_i viene scelta in modo da minimizzare la lunghezza media L_m del codice

$$L_m = S_{i=1,n} (p_i * l_i)$$

Effettuata la ricerca della n-pla di valori l_i che rende minima L_m si ottiene un codice a lunghezza variabile a minima ridondanza. Le medesime informazioni codificate con un codice a lunghezza fissa avrebbero richiesto un codice di lunghezza maggiore di L_m

Codifica a lunghezza variabile

- Data la rappresentazione $c : D \rightarrow R$, l'insieme R può essere costituito da stringhe di lunghezza differente
- Esempio (stringhe di cifre da 0 a 3):

Informazione	*codice*	*Informazione*	*codice*	*Informazione*	*codice*
Casa	0	Banca	32	Andrea (C)	3310
Genitori	1	Paolo	3300	Andreani	3311
Segretaria	2	Anna	3301	Marocco	3312
Direttore	30	Mario	3302	Daita	3313
Taxi	31	Mario (C)	3303	Luchini	3320

Codifica a lunghezza variabile

- La corrispondenza viene decisa tenendo conto della frequenza con cui vengono usati i valori in *D*
- Vantaggi:
  - Risparmio di spazio nella memorizzazione
  - Risparmio di tempo nella trasmissione

Binary Digit (bit)

- $R \equiv \{ 0 , 1 \}$
- Può rappresentare qualunque informazione a due valori ($|D| = 2$)
- Una stringa di m bit può assumere 2^m valori diversi
- Esempio:

♥	00
♦	01
♣	10
♠	11

$|D| = 4$

$m = 2$

Codifica con stringhe di bit

- Per un qualunque insieme D finito:

$$c : D \rightarrow \underbrace{\{0,1\} \times \ldots \times \{0,1\}}_{\lceil \log_2 |D| \rceil \text{ volte}}$$

- Esempio:

Lunedì	000	111
Martedì	001	001
Mercoledì	010	110
Giovedì	011	000
Venerdì	100	101
Sabato	101	100
Domenica	110	010

$$|D| = 7$$

$$\lceil \log_2 |D| \rceil = 3$$

$$2^3 = 8$$

codifica che non usa 111

codifica che non usa 011

Significato di un codice

- Un codice non ha significato di per sé
- Il significato è attribuito dalla codifica (cioè dalla funzione *c*)
- L'associazione stringa-codifica è data dall'operatore umano
- Ad esempio, la stringa 1000 0101 rappresenta:
  - il numero naturale 133 in binario naturale
  - il numero naturale -123 in complemento a 2
  - il carattere à in codice ASCII esteso

Il tipo Atomico Assoluto

Dato un tipo di cardinalità K, nell'implementazione della macchina che deve trattare tale dato si deve progettare la circuiteria per la:
- Memorizzazione : si deve disporre di elementi k-stabili
- Trasmissione : si deve disporre di segnali a k valori
- Elaborazione : si deve disporre di circuiti che trattano k valori discreti

Pertanto la scelta del valore k è dettata da considerazioni pratiche e non teoriche:
- esistenza di elementi economici a k stati stabili
- semplificazione nella discriminazione dei k livelli
- semplicità dei circuiti per le elaborazioni

Ecco da dove viene fuori k = 2.

Codifica indiretta

Codifica diretta

$T=(x_1,x_2,\ldots,x_{20})$

$E=(0,1)$

$m=\{[\log_k N]\}$

Codifica indiretta

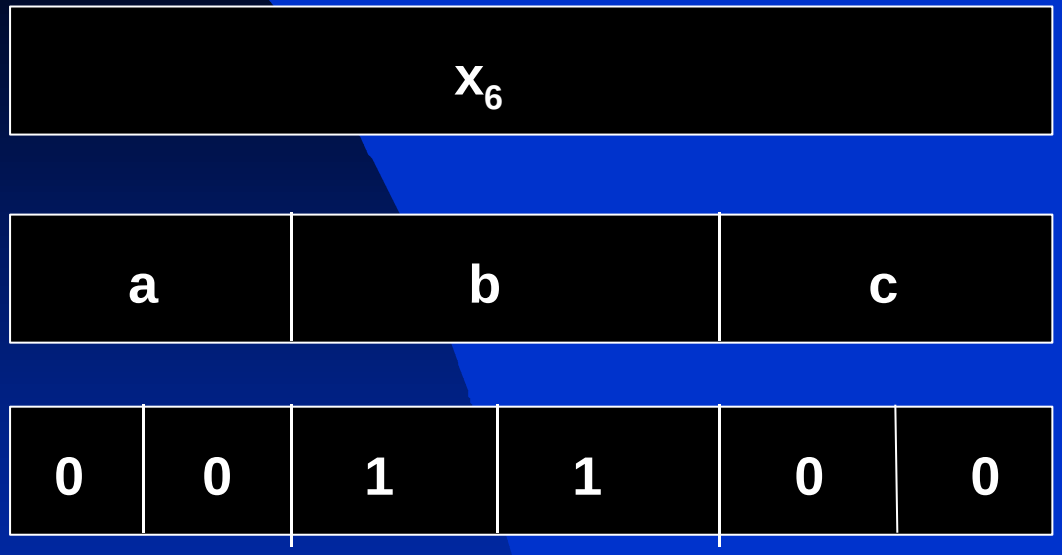
$T=(x_1,x_2,\ldots,x_{20})$ alfabeto origine

$J=(a,b,c)$ alfabeto intermedio, di cardinalità k intermedia $2<k<20$

$B=(0,1)$ alfabeto destinazione

X16 → bac → 011100

Registri e Codifica Indiretta

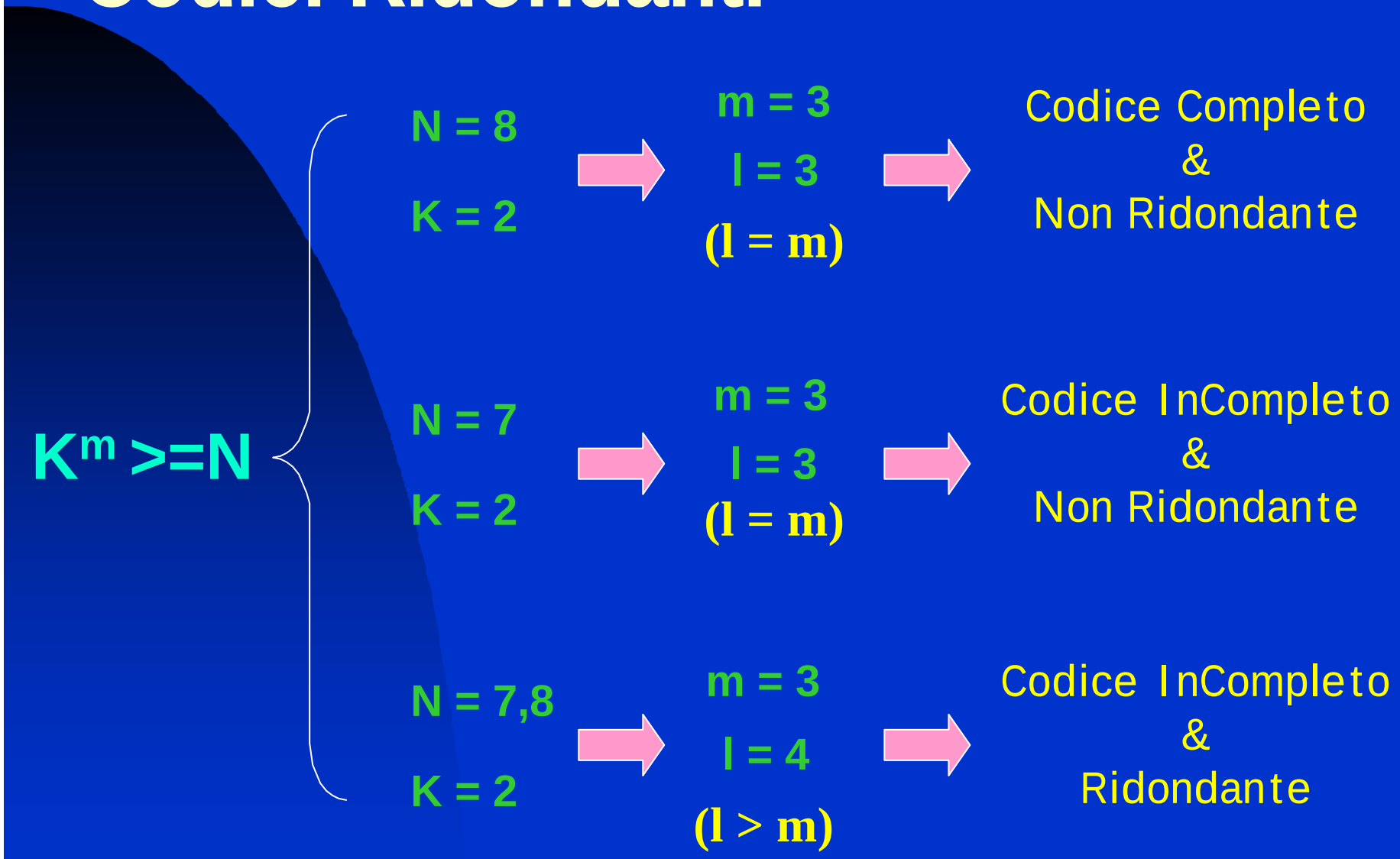


$T=(x_1,x_2,\ldots,x_{20})$ alfabeto origine

J=(a,b,c) alfabeto intermedio

B=(0,1) alfabeto destinazione

Codici Ridondanti



Codici Ridondanti

Un codice a lunghezza fissa l per la codifica di un dato di cardinalità N con un alfabeto codice di cardinalità k si dice ridondante se:

$$K^l > N$$

(l > m, con m : $K^m = N$)

Una misura della ridondanza è:

$$r = 1 - [(\log_k N)/l] \in [0,1]$$

r è uguale a 0 per $\log_k N = l$ mentre tende ad 1 per l che tende ad infinito

Codici Ridondanti

La ridondanza di un codice può essere adoperata per identificare eventuali malfunzionamenti che per una causa qualsiasi conducano a parole-codice errate.
Infatti solo N delle k^l parole-codice sono lecite, le altre rappresentano condizioni anomale. Data una stringa S di l simboli, rappresentante una parola codice:

- se S non è una stringa lecita, S è certamente errata
- se S è una stringa lecita, S è probabilmente corretta

Come fare per capire se una stringa è lecita ?

Codici Ridondanti

L'apparecchiatura che riceve un codice, lo analizza per controllare se esso è lecito:

- Se non lo è, lo dichiara errato.
- Se è lecito, lo accetta anche se potrebbe essere errato.

Aumentando la ridondanza si può soltanto aumentare la probabilità che una stringa lecita sia corretta, mai trasformarla in certezza.

Il confronto per stabilire se un codice è lecito, potrebbe avvenire confrontando il codice corrente con tutte le parole lecite. In genere però si utilizzano particolari proprietà delle parole lecite.

Codici Ridondanti: il bit di parità

Tipico controllo effettuato è il *controllo di parità* : le parole codice lecite hanno tutte un numero pari di bit “1”, quelle illecite un numero dispari.

Per realizzare il controllo di parità basta aggiungere un bit in più rispetto a quelli strettamente necessari, detto appunto *bit di parità*.

Dato un codice a k bit non ridondante, si contano i bit uguali ad “1” e si aggiunge un bit di parità con valore 1 se i k bit sono in numero dispari, 0 se sono in numero pari.

E’ chiaro che la parità dei k+1 bit non indica certezza assoluta di assenza di errore.

Rilevazione e correzione degli errori

- Natura degli errori
- Codici ridondanti
- Rilevazione e correzione degli errori
- Bit di parità: si aggiunge un bit in modo che il numero di 1 sia pari:

  00101101 $\rightarrow$ 00101101 0

  01101000 $\rightarrow$ 01101000 1
- L'errore su un bit viene rilevato (non corretto)

  011010001 $\rightarrow$ 011000001

Rilevazione e tolleranza degli errori

- Rilevazione (mediante bit di parità) e correzione mediante aggiunta di informazioni ridondanti (bit di parità + checksum)

10110010	0
11100100	0
01010110	0
00000111	1
10000110	1
10000001	0

Check Sum

Bit di parità

101	0010	0
111	0100	0
	0	
000	0111	1
100	0110	1
100	0001	0

Rappresentazione posizionale dei numeri naturali

- *Numeri* e *rappresentazione* dei numeri:

  quindici

  15 XV 1111_2

- Rappresentazione posizionale: base di rappresentazione (es. 10), si usano 10 simboli (*cifre*) che rappresentano i numeri da 0 a 9

  $15 = 1 \times 10^1 + 5 \times 10^0$

- Il numero viene rappresentato dalla lista di cifre

Rappresentazione posizionale dei numeri naturali

- *Numerazione Araba: sistema posizionale ordinato secondo le potenze di una base intera. Tale base b è pari a 10.*

- $15903.477 = 1 \times 10^4 + 5 \times 10^3 + 9 \times 10^2 + 0 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 7 \times 10^{-2} + 7 \times 10^{-3}$

- Tale modo di procedere vale per qualsiasi base b

Rappresentazione posizionale dei numeri naturali

- Base di rappresentazione (*b*)
- Un numero x si rappresenta mediante una stringa di cifre

$$X_{n-1}\, X_{n-2} \ldots X_1\, X_0,\, X_{-1} \ldots X_{-m}$$

In questo modo

$$\sum_{i=-m,n-1} X_i \times b^i$$

- In queste ipotesi la rappresentazione è unica

Rappresentazione posizionale dei numeri naturali

$$\sum_{i=-m,n-1} X_i \times b^i$$

- Dove:
  - Le n cifre $X_{n-1}\ X_{n-2}\ldots X_1\ X_0$ prima del "punto frazionario" rappresentano la parte intera
  - Le m cifre $X_{-1}\ \ldots X_{-m}$ dopo il "punto frazionario" rappresentano la "parte frazionaria"
  - Ciascuna cifra X_i è compresa tra 0 e b – 1 ($0 \leq X_i < b$)

Rappresentazione posizionale dei numeri

- Esempio: NUMERAZIONE BINARIA

$b = 2$, cifre = { 0 , 1 }

rappresentazione di *quindici*:

$$15 = 8 + 4 + 2 + 1 = 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 1111$$

rappresentazione di *duecentododici*:

$$212 = 128 + 64 + 16 + 4 = 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 11010100$$

Rappresentazione posizionale dei numeri

- Esempio: NUMERAZIONE ESADECIMALE

$b = 16$,

cifre = { 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F }

rappresentazione di *quindici*:

F

rappresentazione di *duecentododici* :

D4

$212 = 208 + 4 = 13 \times 16^1 + 4 \times 16^0 = D \times 16^1 + 4 \times 16^0 = D4$

$(208 = 13 \times 16)$

Conversione da rappresentazione a numero

- Da base 2 a base 10:

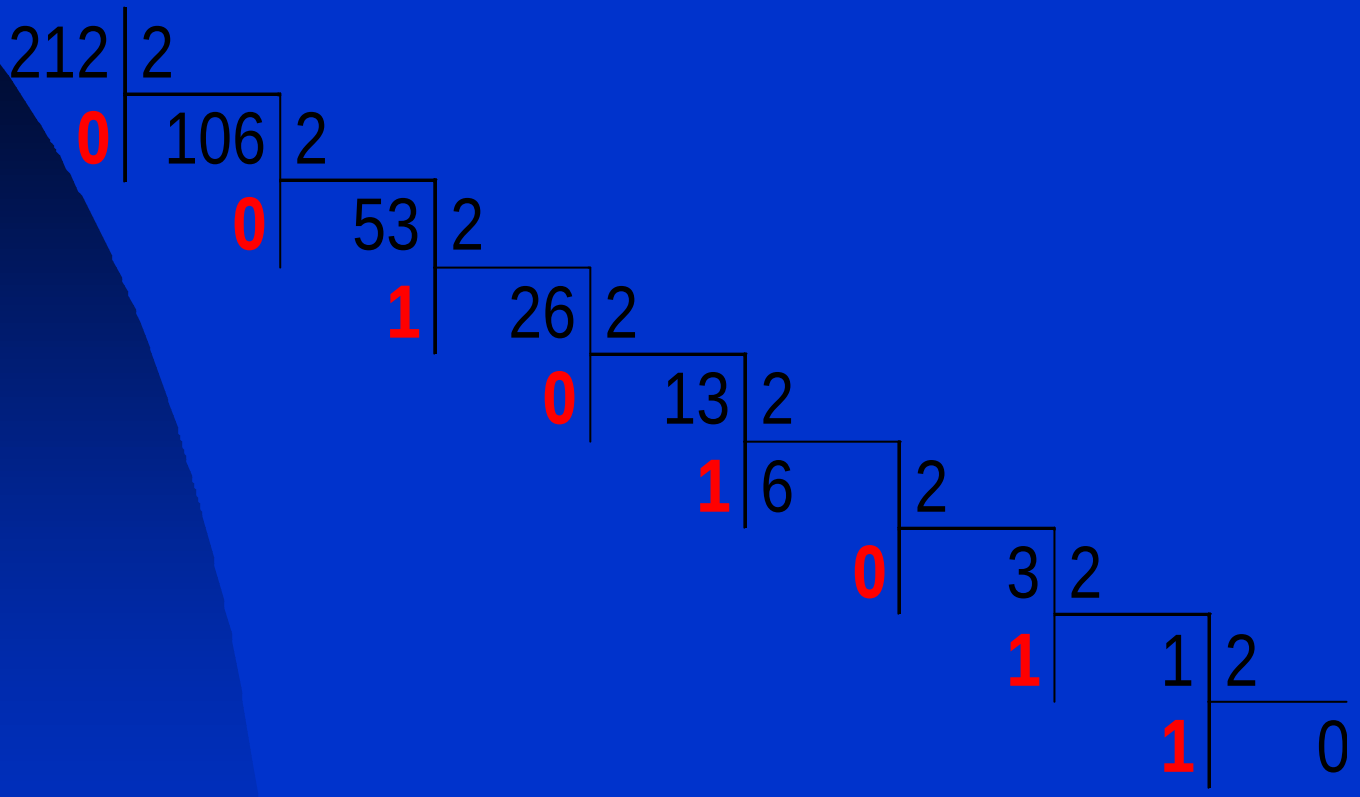
$$11010100_2 \rightarrow 0 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 0 \times 2^3 + 1 \times 2^4 + 0 \times 2^5 + 1 \times 2^6 + 1 \times 2^7 = 212_{10}$$

- Metodo alternativo:

$$(((((((1) \times 2 + 1) \times 2 + 0) \times 2 + 1) \times 2 + 0) \times 2 + 1) \times 2 + 0) \times 2 + 0 =$$
$$1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 212_{10}$$

Conversioni da numero a rappresentazione

- In base 2:



divisioni successive per 2 fino a un quoto uguale a 0
i resti (dall'ultimo al primo) danno la sequenza di cifre

Base esadecimale

- b = 16, cifre = { 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F }
- 212 : 16 = 13 col resto di 4

$$212_{10} = D4_{16}$$

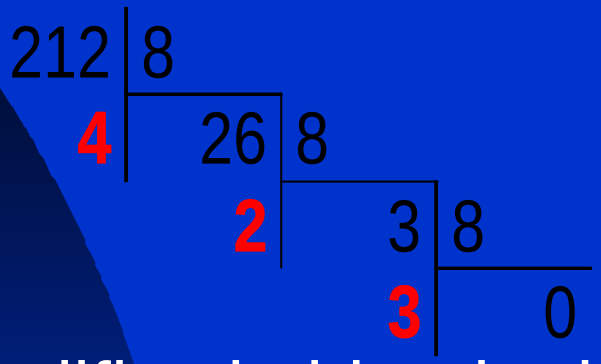
notazioni pratiche: D4H, 0xD4

- La codifica in binario delle 16 cifre esadecimali richiede 4 bit
- Conversione esadecimale-binario:

$$11010100_2 \leftrightarrow \underbrace{1101}_{\text{rappresentazione in base 2 di 13 su 4 bit}}\ \underbrace{0100}_{\text{rappresentazione in base 2 di 4 su 4 bit}} \leftrightarrow D4_{16}$$

Base ottale

- b = 8, cifre = { 0,1,2,3,4,5,6,7}



$212_{10} = 324_8$

- La codifica in binario delle 8 cifre ottali richiede 3 bit
- Conversione ottale-binario:

rappresentazione in base 2 di 4 su 3 bit

$$11010100_2 \leftrightarrow 011\ 010\ 100 \leftrightarrow 324_8$$

rappresentazione in base 2 di 3 su 3 bit

rappresentazione in base 2 di 2 su 3 bit

Rappresentazione dei caratteri

- Codice ASCII (American Standard Code for Information Interchange)
- Rappresentazione su 7 bit: 128 combinazioni
  - da 0 a 31: “caratteri” di controllo
  - da 32 a 47: interpunzione e caratteri speciali
  - da 48 a 57: cifre decimali
  - da 58 a 64: interpunzione e caratteri speciali
  - da 65 a 90: lettere maiuscole dell’alfabeto inglese
  - da 91 a 96: interpunzione e caratteri speciali
  - da 97 a 122: lettere minuscole dell’alfabeto inglese
  - da 123 a 127: caratteri speciali

Rappresentazione dei caratteri

- Relazioni tra caratteri e numeri
  - le stringhe di bit non hanno significato di per se: a ogni carattere corrisponde un numero da 0 a 255
  - il valore numerico di una cifra si ottiene sottraendo al numero corrispondente alla cifra quello corrispondente a 0
- Ordinamento dei caratteri:
  - rispettato l'ordinamento relativo tra: cifre, maiuscole, minuscole
  - spazio < cifre < maiuscole < minuscole

Estensioni del codice ASCII

- Codice ASCII esteso
  - i caratteri da 128 a 255 rappresentano vari caratteri speciali, simboli matematici e lettere non appartenenti all'alfabeto inglese
- Lo standard UNICODE
  - rappresentazione su 16 bit compatibile con il codice ASCII
  - meccanismi di estensione per superare il limite di 65536
- Codici per la rappresentazione di cifre decimali
  - codice EBCD