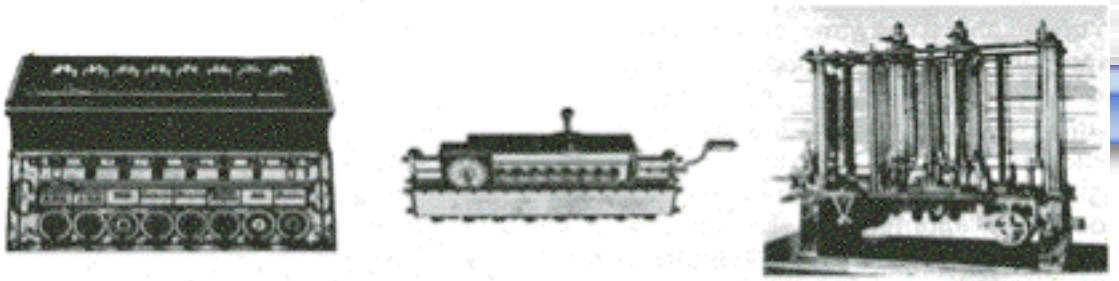
La lezione introduttiva sui dati e le informazioni

Crescenzio Gallo

Introduzione



La pascaline | La macchina di Liebniz | Il calcolatore meccanico di Babbage

Il moderno elaboratore nasce come *estensione della capacità di calcolo* umana, per l'accelerazione di calcoli lunghi e complessi (si pensi ad es. alla determinazione della *n*-ma cifra di π o al calcolo di $n!$).

Nella maggior parte delle moderne applicazioni prevale invece la sua capacità di memorizzare enormi quantità di *dati* ed *informazioni* (vedremo la differenza), che sono una forma di rappresentazione astratta di una parte del mondo reale.

Introduzione

L'informazione resa disponibile riguarda un insieme selezionato di dati relativi al mondo reale e rilevante per il problema da modellare e quindi risolvere.

L'*astrazione* consiste nell'ignorare certe proprietà e caratteristiche degli oggetti reali perché marginali e irrilevanti per il problema in esame: ciò semplifica l'attività di *problem-solving*.

Ad es. nella gestione dei dati anagrafici degli studenti per un'università è irrilevante memorizzare dati del tipo "colore degli occhi", "peso", "altezza", etc.

Introduzione

Una volta scelti i dati "rilevanti", altrettanto importante è la scelta della loro *rappresentazione*, spesso difficile e non univoca e tipicamente dipendente dalle operazioni che su di essi si andranno a svolgere.

Ad es. nel caso di dati numerici (essi stessi astrazione, in senso platoniano, delle proprietà degli oggetti che caratterizzano) può essere adottata la rappresentazione *additiva* (ad es. il sistema di numerazione romano) o *posizionale* (quello arabico da noi utilizzato).

Introduzione

Un ulteriore aspetto è relativo al *tipo* del dato, cioè all'insieme dei suoi valori (cardinalità) e delle operazioni ammissibili.

Vi sono alcuni tipi *base* presenti in tutti i linguaggi:

- numeri interi (e tipi scalari derivati)
- numeri reali(*)
- caratteri alfanumerici
- valori logici (vero, falso)

(*) con la precisione offerta dallo strumento utilizzato

Introduzione

Dai tipi base mediante l'uso di costruttori si possono ottengono i *tipi strutturati* (cioè composti da altri tipi, base o strutturati, ricorsivamente). Ad es.(*):

```
<numero>::=<numero intero>|<numero reale>
<numero intero>::=<cifra>{<cifra>}
<cifra>::=0|1|2|3|4|5|6|7|8|9
<numero reale>::= . . .
```

(*) *viene utilizzata una notazione intuitiva detta "metasintassi BNF"*

Introduzione

I costruttori fondamentali (metodi di strutturazione sufficienti ad esprimere la maggior parte delle situazioni reali da modellare e dei problemi da risolvere) sono:

- Array (vettori, in matematica)
- Record (agglomerati di varia natura)
- Insiemi (nel senso matematico)
- Sequenze (o file)
- Liste (concatenazioni di record)
- Alberi (strutture gerarchiche) o grafi finiti

Codice

La matematica, la logica, l'informatica, … si servono di rappresentazioni formali, cioè di "insiemi di segni allineati" ed organizzati, per esprimere i loro concetti.

Partiamo da un *alfabeto* = insieme finito di segni grafici (detti *caratteri*): al minimo ne occorreranno 2 (ad es. alfabeto binario = {0,1}); con i 10 segni {0,1,2,3,4,5,6,7,8,9} possiamo costruire tutti i numeri in rappresentazione decimale.

L'**alfabeto latino** è il sistema di scrittura alfabetica più diffuso nel mondo. È l'alfabeto adottato dalla grande maggioranza delle lingue dell'Europa centrale e occidentale, nonché delle aree del mondo colonizzate dagli europei.

Durante il XX secolo è stato anche adottato da alcune lingue non europee.

Le lettere maiuscole													
A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Lettere dell'Alfabeto Greco					
Α Alfa	Β Beta	Γ Gamma	Δ Delta	Ε Epsilon	Ϝ Digamma*
Ζ Zeta	Η Eta	Θ Theta	Ι Iota	Κ Kappa	Λ Lambda
Μ Mu	Ν Nu	Ξ Xi	Ο Omicron	Π Pi	Ϻ San*
Ϙ Qoppa*	Ρ Rho	Σ Sigma	ς Sigma finale	Τ Tau	Υ Upsilon
Φ Phi	Χ Chi	Ψ Psi	Ω Omega	Ϡ Sampi*	

* lettere arcaiche

Anche quello latino e greco sono *alfabeti* in tale accezione.

Codice

Si dice *parola* una sequenza di segni (caratteri) presi da un alfabeto A.

Ad es. 0100101 è una "parola" dell'alfabeto binario; il n.ro di caratteri si dice *lunghezza* della parola.

Se card(A)=n allora il numero totale di "parole" di lunghezza m" è pari a n^m (disposizioni con ripetizione di n caratteri su m posti).

Se ad es. A={0,1} e m=8, abbiamo 2^8=256 possibili "parole" binarie (o meglio, bytes, come vedremo).

Codice

Sia ancora $A=\{0,1\}$. Dato un insieme finito X (di altri simboli, ad es. {a,…,z}), consideriamo la più piccola potenza m di 2 ≥ $(X)\#$; cioè, sia m tale che:

$$2^{m-1} < (X)\# \leq 2^m$$

Allora è possibile stabilire $f:X \rightarrow Y$ iniettiva(*) (dove $Y=\{$parole di lunghezza m di $A\}$) detta *codice*.

Ad es. nel codice ASCII (in cui $m=8$) il carattere @ corrisponde alla "parola binaria" 01000000.

(*) *f dev'essere iniettiva perché a elementi diversi di X devono corrispondere codici diversi; difficilmente è suriettiva… Perché?*

Codice

In molte attività umane si utilizzano i codici; anche la natura ne fa largo uso, vedi ad es. il *codice genetico* (che ha come alfabeto {A,G,T,C}, lettere iniziali delle famose componenti base del DNA).

Fissato un alfabeto (ad es. ancora $A=\{0,1\}$), il n.ro m tale che $2^{m-1}<(X)\#\leq 2^m$ rappresenta quindi la *lunghezza minima di un messaggio* che permette di individuare un elemento di X, cioè, la *quantità di informazione*(*) necessaria.

(*) *il che rende l'informazione un concetto meno "nebuloso" e più misurabile, legato al concetto di entropia.*

Informazione

INFORMATICA = informazione automatica = *disciplina che include problematiche, teorie, metodi, tecniche e tecnologie del trattamento (rappresentazione, elaborazione, conservazione, trasmissione, etc.) automatico delle informazioni.*

informazione = "materia prima" della convivenza civile, avente **forma** (numerica, alfanumerica, grafica) e **contenuto** (quali/quantitativo)

=> *esigenza di utilizzare metodologie e dispositivi atti a risolvere i molteplici aspetti che coinvolgono il dominio dell'informazione.*

Dato e informazione

DATO = rappresentazione *simbolica* ed *astratta* di entità (concrete o ideali).

Il dato "grezzo", come ad esempio: **27**, **0881675421**, **FG*510234**, **LEONE** non ha di per sé alcun significato. Ma:

Informazione = dato + significato

- **27:** può essere il giorno di riscossione dello stipendio, oppure l'età di una persona, o la lunghezza in cm. di un oggetto, ...
- **0881675421:** può essere un numero di telefono, o il codice di un articolo nel magazzino 0881, ...
- **FG*510234:** può essere una targa automobilistica, l'identificativo di una patente, ...
- **LEONE:** il re della foresta, o l'ex Presidente della Repubblica, ...

Dato e informazione

La targa dell'auto di Gianni è	**FG*510234**
descrittore	*dato*

Attenzione:

- **Il calcolatore elabora DATI;**
- **l'uomo è in grado di usare INFORMAZIONI.**

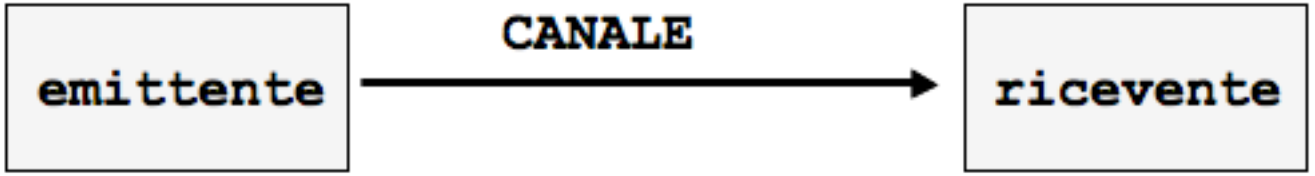
Elaborazione dell'informazione = trasformazione della stessa nella forma e/o nel contenuto => uso di un sistema (cioé i dispositivi hardware) e delle regole che ne definiscano il funzionamento (software) con finalità che qualificano il sistema stesso.

Dato e informazione

Informazione = ***entità che riduce lo stato di incertezza*** (*entropia*)

Aspetti fondamentali:

- **utilità**
- **emittente/ricevente**
- **linguaggio**
- **canale** (mezzo che offre il supporto fisico alla trasmissione) ==> *capacità* (ampiezza di banda), *rumore* (qualità trasmissiva)
- **supporto** (di memorizzazione)



Rappresentazione dei dati

DATI = *astrazioni con cui rappresentiamo gli oggetti della realtà*

Possono essere:

- **numerici** (virgola fissa o mobile) o **non-numerici** (testo, immagini, grafici, ...);
- **semplici** o **strutturati** (composti).

Rappresentazione dei dati

Dati semplici:

- 39
- "Maria"
- Martedì
- 2006

Dati strutturati:

- **data** = (giorno, mese, anno)
- **telefono** = (prefisso, numero)
- **domicilio** = (via, numero, cap, comune, provincia)

Rappresentazione dei dati

scheda anagrafica =

- nome
- cognome
- data di nascita
- residenza
- cittadinanza
- domicilio(…) ← *dato strutturato su più livelli*
- stato civile
- professione

Rappresentazione dei dati

In un dato è possibile distinguere:

- **nome** (data di nascita)
- **valore** (4/3/1943)
- **formato** di rappresentazione (*giorno*/*mese*/*anno* piuttosto che *mese*/*giorno*/*anno*)

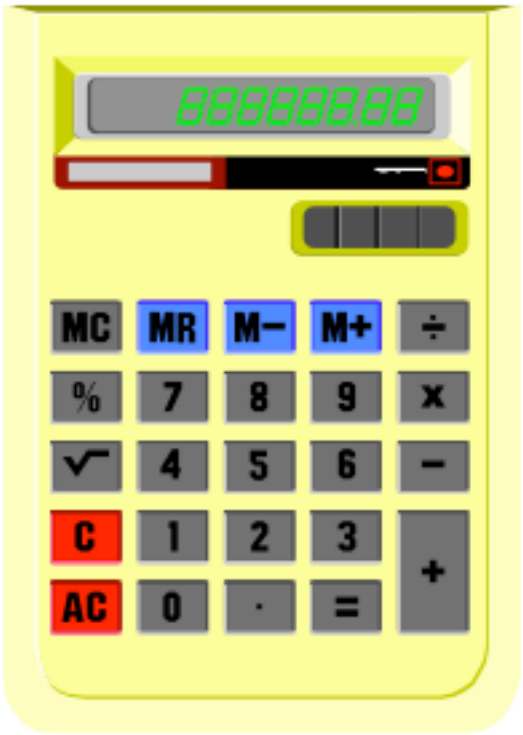
Rappresentazione dei dati

Possiamo distinguere tra:

dati primitivi (ad es. la data di nascita)

Rappresentazione dei dati

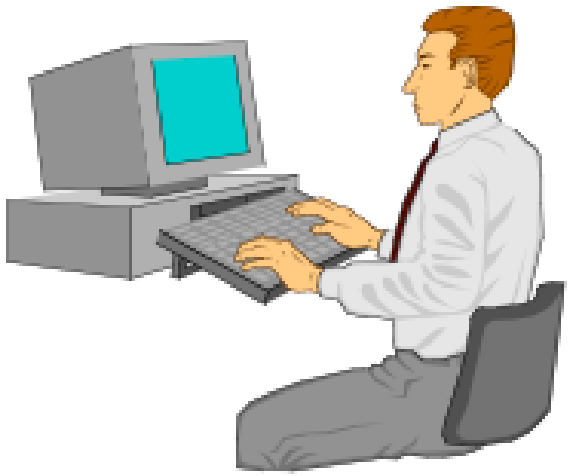
dati calcolati (cioé ottenuti in funzione di altri, come ad es. l'età), utilizzati per ragioni di efficienza e da evitare poiché fonte di inutili ridondanze e potenziali incongruenze.



Rappresentazione dei dati

Rispetto ad un elaboratore, i dati possono essere suddivisi in:

- **dati di entrata** (*input*)



Rappresentazione dei dati

Rispetto ad un elaboratore, i dati possono essere suddivisi in:

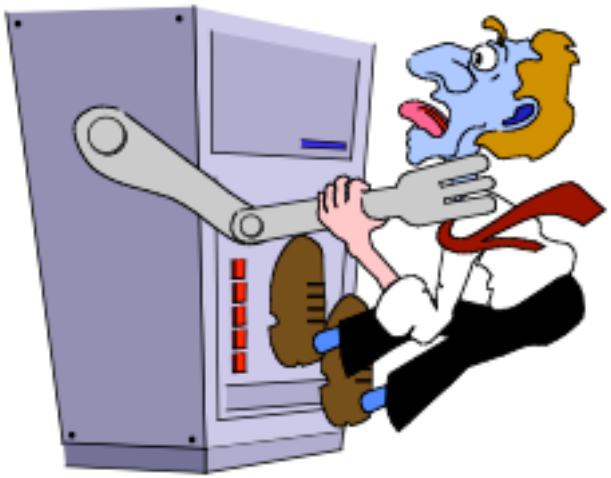
- **dati intermedi** (*locali*)

Rappresentazione dei dati

Rispetto ad un elaboratore, i dati possono essere suddivisi in:

- **dati in uscita** (*output*)

*L'**uomo**, attribuendo un **significato** ai **dati** in uscita (risultati), riceve delle **informazioni**.*



Rappresentazione dei dati

Codifica Binaria dei Dati

Elaboratori → dati in forma ***binaria***
simboli **0** e **1** = ***bit*** (per semplicità operativa ed economicità costruttiva)

CODICE

corrispondenza fra le informazioni utilizzate dall'uomo e i dati binari trattati dall'elaboratore

Rappresentazione dei dati

[A]=01000001 [B]=01000010 [1]= 00110001 …

Il Codice ASCII

Codice ASCII base

- **7 bit** per simbolo (2^7=**128 caratteri**)

Codice ASCII esteso

- **8 bit** per simbolo (2^8=**256 caratteri**)

8 bit = *byte* = carattere (lettera, cifra, simbolo speciale)

Rappresentazione dei dati

La *rappresentazione binaria* è utilizzata per codificare:

- **numeri interi**;
- **numeri decimali** (*fixed* e *floating-point*);
- **caratteri** e stringhe di caratteri;
- **istruzioni** nel linguaggio macchina;
- **insiemi di simboli**;
- **simboli grafici.**

Codifica binaria

Unità minima di rappresentazione:

bit (*bi*nary digi*t* = cifra binaria = 0,1)

Multipli:

byte (8 bit); ad es. 01000001

parola (2, 4, 8 byte - in relazione alla dimensione dei registri della CPU)

KB=1024 (2^{10}) byte (*)

MB=1024xKB (2^{20}) byte

GB=1024xMB (2^{30}) byte

(*) Attenzione: 1024 e non 1000!!!

Codifica binaria

Dati alfanumerici: codice ASCII o EBCDIC (1 carattere = 1 byte), Unicode (1 car. = 2 byte).

Ad es. in ASCII la parola CASA è rappresentata come:

C	A	S	A
01000011	01000001	01010011	01000001

Codifica binaria

Dati numerici in virgola fissa: sono di fatto equivalenti a numeri interi, con un fattore di scala fisso; rappresentati per cifre (BCD) o con conversione in base 2.

Ad es. il n.ro 14 in BCD è [00010100], mentre in base 2 diventa [00001110]

=> *l'interpretazione dipende dalle convenzioni stabilite!*

Codifica binaria

Dati numerici in virgola mobile (*notazione scientifica*). Ad es.: 3e-4 = $3 \cdot 10^{-4}$ = 0,0003

Si preferisce la cosiddetta *forma normalizzata* $0{,}3 \cdot 10^{-3}$ (mantissa<1 + esponente).

Overflow = impossibilità a rappresentare l'ordine di grandezza del numero.

Codifica binaria

Attenzione! I codici delle lettere maiuscole e minuscole sono diversi ed ovviamente progressivi, per rispettare l'ordinamento alfabetico; la codifica di una cifra come numero è diversa da quella del carattere.

Il *numero* **213** in binario è **11010101** e occupa 8 bit = 1 byte; la *sequenza* 213 (da interpretare come successione dei tre caratteri 2, 1 e 3) è rappresentata con [00110010] [00110001] [00110011] ed occupa 24 bit = 3 byte.

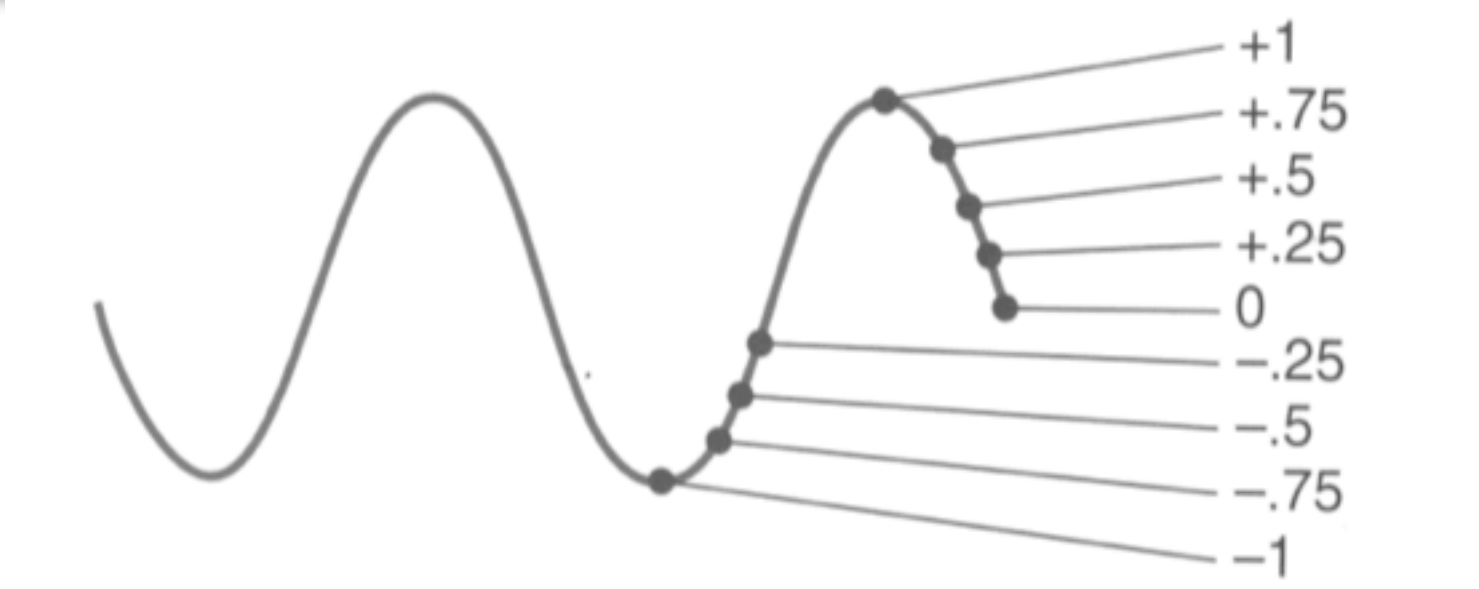
Informazione analogica e digitale

L'informazione può essere rappresentata in due modi:

- **analogica** (continua) - può assumere qualsiasi valore in un intervallo continuo (infinito);
- **digitale** (discreta) - può assumere solo un numero finito di valori.

Come già osservato, gli elaboratori utilizzano informazioni digitali con due soli valori permessi (0, 1).

Analogico / digitale



Segnali analogici

Segnali digitali

Vantaggi dell'informazione binaria

Semplicità: sono permessi due soli valori che possono essere interpretati come *zero/uno*, *spento/acceso*, etc.

Espandibilità: situazioni più complesse si possono ottenere combinando più valori binari.

Chiarezza: si riducono le possibilità di errore perchè occorre scegliere fra due soli valori.

Velocità: le elaborazioni (ed i circuiti) si semplificano se vi sono due soli valori (stati).

Il sistema binario

E' costituito da due sole cifre, 0 e 1 (unici resti possibili nella divisione di un numero naturale per 2).

Le posizioni delle cifre costituenti un numero binario corrispondono ad opportune potenze della base 2 (*forma polinomiale* del numero):

$$(1101)_2 = \mathbf{1}\cdot 2^3 + \mathbf{1}\cdot 2^2 + \mathbf{0}\cdot 2^1 + \mathbf{1}\cdot 2^0 = (13)_{10}$$

---

essendo $2^0=1$, $2^1=2$, $2^3=8$, $2^4=16$, $2^5=32$, $2^6=64$, $2^7=128$, …

Il sistema binario

Sistema decimale

$$(127)_{10} = 1x10^2+2x10^1+7x10^0 = 100+20+7$$

Sistema binario

$$(10100101)_2 = 1x2^7+1x2^5+1x2^2+1x2^0$$
$$= 128+32+4+1 = (165)_{10}$$

binario	0	1	10	11	100	101	110	111	1000
decimale	0	1	2	3	4	5	6	7	8

Il sistema binario

Binario → Decimale

quoziente	*resto*
106	0
53	1
26	0
13	1
6	0
3	1
1	1

$1101010 = 1x2^6 + 1x2^5 + 1x2^3 + 1x2^1 = 64 + 32 + 8 + 2 = 106$

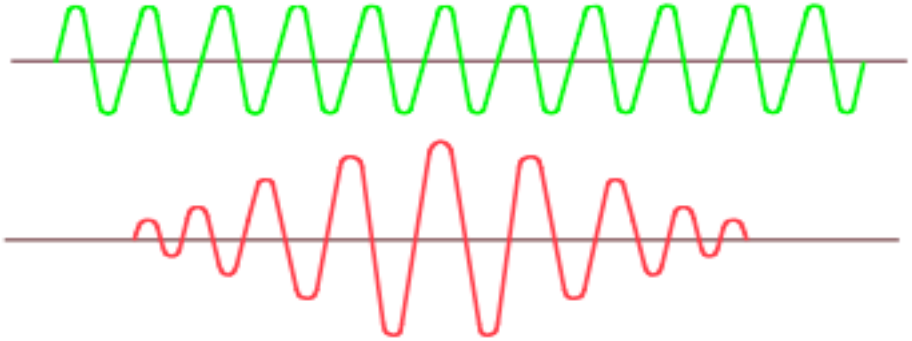
Il sistema ottale ed esadecimale

$$(55153)_8 = 5x8^4 + 5x8^3 + 1x8^2 + 5x8^1 + 3x8^0 = (23147)_{10}$$

$$(5A6B)_{16} = 5x16^3 + 10x16^2 + 6x16^1 + 11x16^0 = (23147)_{10}$$

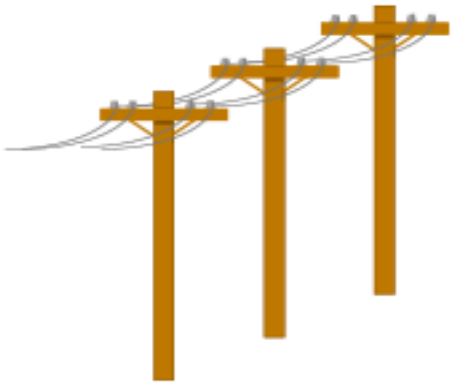
binario	[1 0 1]	[1 0 1]	[0 0 1]	[1 0 1]	[0 1 1]
ottale	**5**	**5**	**1**	**5**	**3**
binario	[1 0 1]	[1 0 1 0]	[0 1 1 0]	[1 0 1 1]	
esadecimale	**5**	**A**	**6**	**B**	

Trasmissione delle informazioni



SEGNALI

EMITTENTE



mezzo trasmissivo (canale)

RICEVENTE

Errori di trasmissione

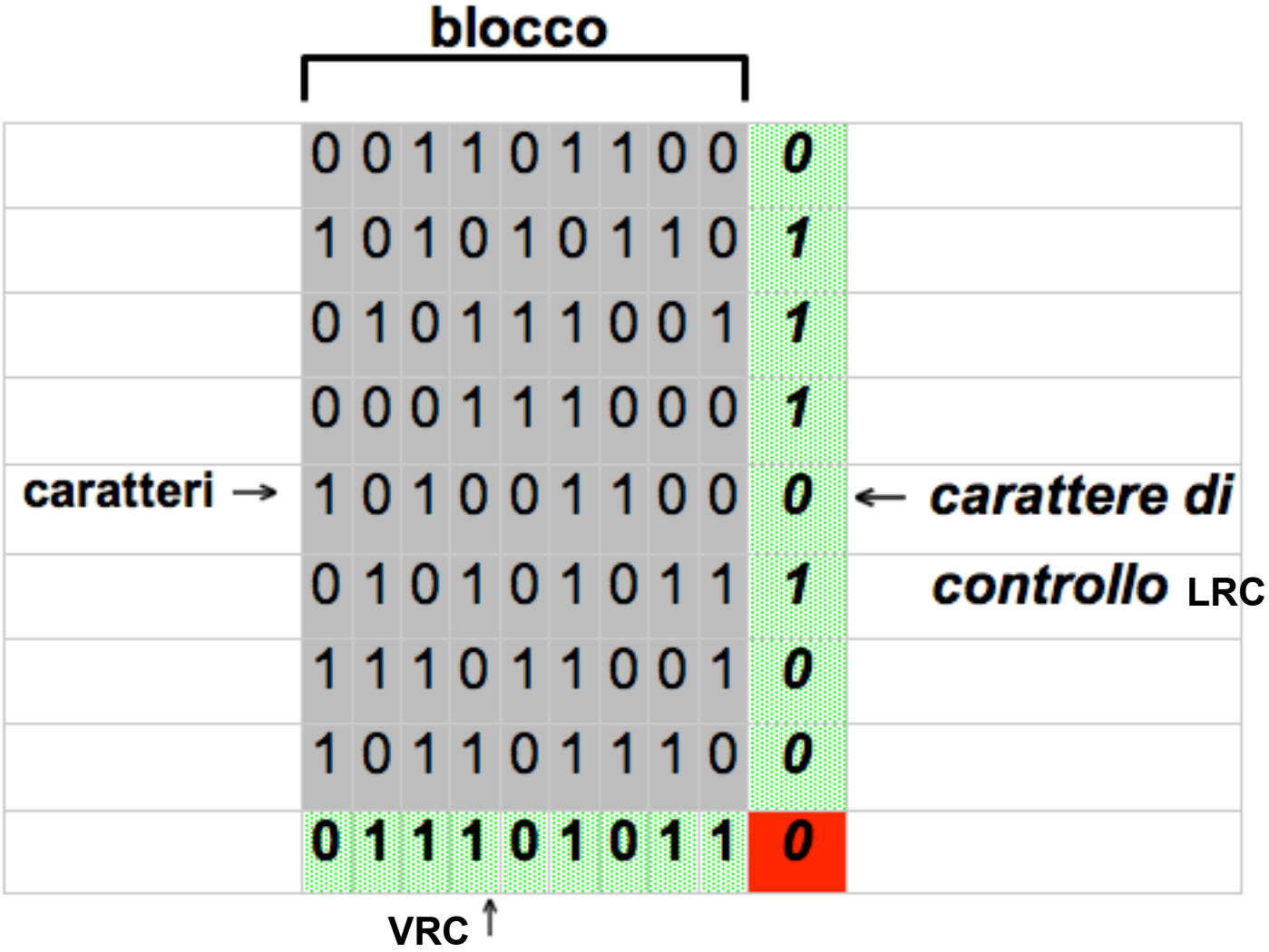
DISTURBI: occasionali alterazioni dei messaggi → **ERRORI** di trasmissione

TASSI DI ERRORE: su linee telefoniche ~ uno ogni cento milioni di bit trasmessi

CONTROLLO DEGLI ERRORI DI TRASMISSIONE

- **controllo di parità verticale (VRC)**
- **controllo di ridondanza longitudinale (LRC)**
- **controllo polinomiale**

Errori di trasmissione



Errori di trasmissione

⇒ **ritrasmissione**

- *controllo semplice*
- *costo di trasmissione maggiore*

⇒ **correzione** (ricostruzione del messaggio a partire dai bit ricevuti e dagli errori riscontrati):

- *più complesso e costoso*
- *applicabile anche per trasmissioni monodirezionali*
- *utile se l'indice di affidabilità è basso (ricorrere sempre alla ritrasmissione del messaggio può voler dire di fatto rallentare notevolmente la velocità del canale)*

Errori di trasmissione