

## Rappresentazione e Codifica dell'Informazione

... problema della rappresentazione dei valori delle informazioni

... uso di sequenze di simboli

Esempi:

*categoria di un albergo* ... sequenza di 'stelle'

*numero matricola* ... sequenza di cifre rappresentanti uno studente

*codice articolo* ...sequenza di simboli rappresentanti un articolo di un negozio

*codice fiscale* ... sequenza di lettere e cifre rappresentanti una persona

*numeri Naturali* ... sequenze di cifre  $\{0, 1, 2, \dots, 9\}$  ....

*parole della lingua italiana* ... sequenze di lettere  $\{a, b, c, \dots, z\}$

## Rappresentazione e Codifica dell'Informazione

la rappresentazione è effettuata attraverso un insieme finito di simboli disponibili

il numero di simboli disponibili è in generale minore del numero di valori da rappresentare

Esempio:

lettere alfabeto italiano (21)  $\implies$  parole lingua italiane ( $\gg$  21)

la rappresentazione avviene associando a ciascun valore da rappresentare una sequenza di simboli

La *CODIFICA* è una tecnica con la quale un dato viene rappresentato mediante un definito insieme di simboli, o di dati, più elementari di qualsiasi natura (grafica, luminosa, acustica, ...)

Con tali simboli è possibile formare sequenze che possono essere messe in relazione biunivoca con gli elementi costituenti l'informazione

## Codifica dell'Informazione

Formalizzando:

l'informazione da rappresentare appartiene ad un tipo **T** a cardinalità  $N$

$$\mathbf{T}=(x_1, \dots, x_n) \quad x_i \text{ generico valore da rappresentare}$$

**T** è detto *Alfabeto Sorgente*

si vogliono rappresentare i valori  $x_i$  tramite gli elementi di un altro tipo **E** a cardinalità  $K \leq N$

$$\mathbf{E}=(a_1, \dots, a_k) \quad a_j \text{ generico simbolo}$$

**E** è detto *Alfabeto in Codice*

La *Codifica* è un'applicazione  $C$ , detta *tabella-codice*, che trasforma ciascun elemento  $x_i \in T$  in una sequenza di elementi  $a_j \in E$ , detta *parola-codice* (di lunghezza  $l_i$ )

## Codifica dell'Informazione

**Esempio:**

*Alfabeto Sorgente:* (a, b, c, ....., v,z)

*Alfabeto in codice:* (-, .)

*Codice*

|          |      |
|----------|------|
| <b>a</b> | .-   |
| <b>b</b> | -... |
| <b>c</b> | -..  |
| <b>o</b> | ---  |
| <b>s</b> | ...  |

## Codifica dell'Informazione

*Codifica a lunghezza fissa*

**la lunghezza  $li$  delle parole codice associate ai valori dell'alfabeto sorgente è costante**

*Codifica a lunghezza variabile*

**la lunghezza  $li$  delle parole codice associate ai valori dell'alfabeto sorgente è variabile**

- il codice può essere una associazione del tutto arbitraria di parole codice a valori
- oppure, può essere fondato su regole ben definite
- esempi: il codice fiscale, il codice esame, numero di matricola

## Codifica dell'Informazione

**Esempio:**

***Alfabeto Sorgente:* (picche, fiori, quadri, cuori)**

***Alfabeto in codice:* (\*, /)**

|               | <i>Codice</i> | <i>Altro Codice</i> |
|---------------|---------------|---------------------|
| <b>picche</b> | *             | **                  |
| <b>fiori</b>  | /             | //                  |
| <b>quadri</b> | **            | */                  |
| <b>cuori</b>  | //            | /*                  |

## Codifica dell'Informazione

*Codifica a lunghezza fissa*

$T=(x_1, \dots, x_n)$  Alfabeto Sorgente, cardinalità  $N$

$E=(a_1, \dots, a_k)$  Alfabeto in Codice, cardinalità  $K$

la parola-codice ha una lunghezza  $l_i = m = costante$  per tutti gli elementi di  $T$

Sono possibili  $k^m$  parole codice differenti (le disposizioni con ripetizione dei  $k$  simboli di  $E$  sugli  $m$  posti della sequenza)

Ad ognuno degli elementi  $x_i \in T$  si fa corrispondere una delle  $k^m$  parole codice e dovrà necessariamente aversi  $k^m \geq N$

(gli  $N$  elementi devono trovare almeno altrettante disposizioni che li rappresentino)

*Per codificare un elemento di un tipo a cardinalità  $N$  mediante un alfabeto in codice di  $K$  simboli è necessaria una sequenza di lunghezza minima  $m$ , con*

$$m = \lceil \log_k N \rceil$$

## Codifica dell'Informazione

**Esempio:**

$T=(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9)$

$E=(\alpha, \beta, \gamma)$

$$m = \lceil \log_3 9 \rceil = 2$$

$$x_1 = \alpha \beta$$

$$x_2 = \beta \alpha$$

$$x_3 = \alpha \gamma$$

$$x_4 = \gamma \alpha$$

$$x_5 = \beta \gamma$$

$$x_6 = \gamma \beta$$

$$x_7 = \alpha \alpha$$

$$x_8 = \beta \beta$$

$$x_9 = \gamma \gamma$$

Codice non ridondante

$$k^m = N$$

## Codifica dell'Informazione

Esempio:

$$T = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9)$$

$$E = (0, 1)$$

$$m = \lceil \log_2 9 \rceil = 4$$

$$x_1 = 0000$$

$$x_2 = 0001$$

$$x_3 = 0010$$

$$x_4 = 0011$$

$$x_5 = 0100$$

$$x_6 = 0101$$

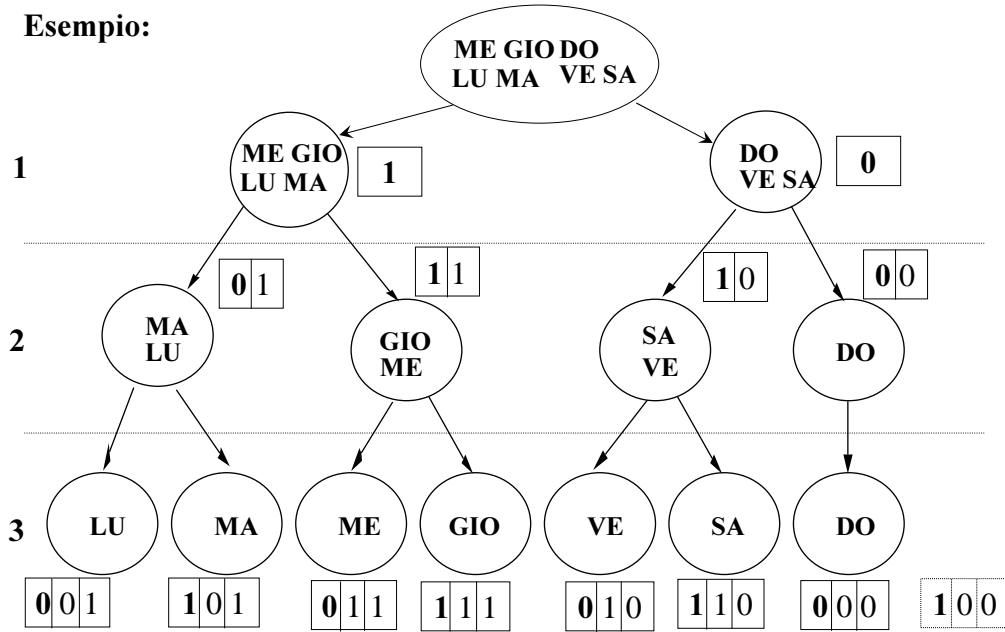
$$x_7 = 0110$$

$$x_8 = 0111$$

$$x_9 = 1000$$

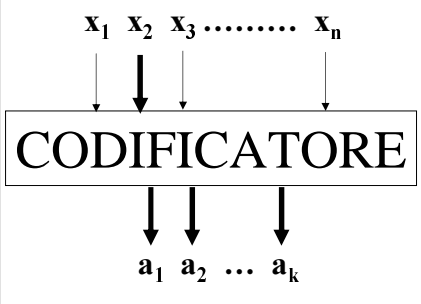
Codice ridondante  
 $k^m > N$

Esempio:



### Un Modello

$x_1$   $x_2$   $x_3$  .....  $x_n$



**CODIFICATORE**

$a_1$   $a_2$  ...  $a_k$

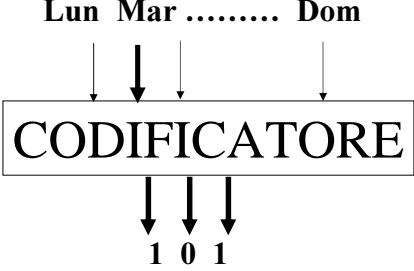
$n$  input, gli elementi dell'alfabeto sorgente, di cui  $1$  solo attivo

Un'applicazione che trasforma un elemento dell'alfabeto sorgente nella parola codice

$k$  output, i simboli dell'alfabeto codice, formanti la parola codice

**Esempio:**

Lun Mar ..... Dom



**CODIFICATORE**

1 0 1

---

Elementi di Informatica  
Prof. G. A. Di Lucca - Univ. del Sannio
11

### Informazione e Registri

Le informazioni (dati e/o istruzioni) trattate da una macchina sono memorizzate in elementi detti *registri*

Il registro può essere visto come un contenitore di valori di informazione; un registro è individuato da un nome

nome valore ← registro

Un registro contiene il valore di un'informazione di un determinato tipo, mentre il nome equivale all'attributo dell'informazione

Esempio:

cliente Rossi ← registro

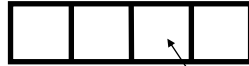
Il valore è rappresentato (codificato) mediante apposite sequenze di simboli di un alfabeto codice

---

Elementi di Informatica  
Prof. G. A. Di Lucca - Univ. del Sannio
12

## Informazione e Registri - 2

Un registro è composto da elementi più semplici (celle) ognuno dei quali può assumere un numero finito di stati



Registro con 4 celle

Elemento atomico del registro

Se il dato è codificato in un alfabeto codice a cardinalità  $k$  (ovvero con  $k$  simboli diversi) e la parola codice ha una lunghezza fissa  $l$ , il registro che lo memorizza è composto da una sequenza di almeno  $l$  celle  $k$ -stabili

*(ciascuna cella deve poter assumere uno dei  $k$  simboli del codice)*

Per la rappresentazione di un'informazione di un tipo a cardinalità  $N$  occorrono registri ad almeno  $N$  stati stabili, ciascuno dei quali individua uno dei possibili valori dell'informazione

## Informazione e Registri - 3

Registro: elemento fisico atto ad assumere  $N$  stati finiti ( $N$ -stabile) associabili a  $N$  valori distinti di un'informazione

Esempio:



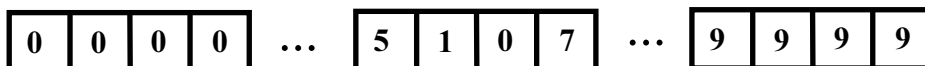
Registro con 4 celle

...supponiamo che:

... ogni cella può assumere 10 stati distinti (deca-stabile)

... ad esempio le cifre da 0 a 9

... il registro in totale può assumere  $10^4 = 10000$  stati distinti



## Informazione e Registri - 4

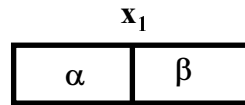
**Esempio:**

$$T=(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9)$$

$$E=(\alpha, \beta, \gamma)$$

$$m = \lceil \log_3 9 \rceil = 2$$

...un registro di 2 celle, ciascuna tri-stabile



$x_1 = \alpha \beta$   
 $x_2 = \beta \alpha$   
 $x_3 = \alpha \gamma$   
 $x_4 = \gamma \alpha$   
 $x_5 = \beta \gamma$   
 $x_6 = \gamma \beta$   
 $x_7 = \alpha \alpha$   
 $x_8 = \beta \beta$   
 $x_9 = \gamma \gamma$

- ciascuna delle celle del registro può assumere 1 dei tre valori ( $\alpha, \beta, \gamma$ )
- celle tri-stabili, registro a 9 stati totali

## Informazione e Registri - 5

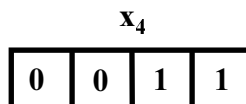
**Esempio:**

$$T=(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9)$$

$$E=(0, 1)$$

$$m = \lceil \log_2 9 \rceil = 4$$

...un registro di 4 celle, ciascuna bi-stabile



$x_1 = 0000$   
 $x_2 = 0001$   
 $x_3 = 0010$   
 $x_4 = 0011$   
 $x_5 = 0100$   
 $x_6 = 0101$   
 $x_7 = 0110$   
 $x_8 = 0111$   
 $x_9 = 1000$

- ciascuna delle 4 celle del registro può assumere 1 dei 2 valori (0,1)
- celle bi-stabili, registro a 16 stati totali

## Informazione e Registri - 6

**In un elaboratore tutte le informazioni sono codificate in *BIT* e la cella elementare dei registri è un elemento bistabile detto *flip-flop***

l'alfabeto codice utilizzato per codificare le informazioni e renderle comprensibili ad una macchina è quindi un alfabeto binario (cardinalità 2): {0, 1}, {spento, acceso}, {vero, falso}, ...

- .... esistenza di elementi fisici economici bistabili
- .... semplicità dei circuiti elettronici di elaborazione
- .... sicurezza del funzionamento, ovvero necessità di ridurre la possibilità di commettere un errore durante una *scelta* (discriminazione tra livelli)
- .... la dimensione (lunghezza) dei registri è 'finita'

## Codifica dell'Informazione

### Codifica Diretta ed Indiretta

**Codifica Diretta:** ad ogni elemento dell'Alfabeto Sorgente è associata una parola codice con i simboli dell'Alfabeto Codice

**Codifica Indiretta:** la parola codice è ottenuta usando un codice intermedio

$$T=(x_1, x_2, \dots, x_n) \quad J=(s_1, s_2, \dots, s_k) \quad E=(e_1, \dots, e_z)$$

Si vogliono codificare gli elementi di T secondo i simboli di E

- un codice intermedio associa ad ogni  $x_i$  una stringa di  $s_j$ 
  - Ciascun  $x_i$  è codificato da una stringa di simboli  $s_j$
- un altro codice associa ad ogni simbolo  $s_j$  una stringa di  $e_w$ 
  - Ciascun  $s_j$  è codificato da una stringa di simboli  $e_w$
- Si sostituiscono i simboli  $s_j$  di ciascuna parola codice corrispondente agli  $x_i$  con la corrispondente parola codice con i simboli  $e_w$ .....
  - Ciascun  $x_i$  è codificato da una stringa di simboli  $e_w$

### Codifica dell'Informazione

**Esempio:**  
**T= (x1,x2,.....x10)    J= (a,b,c)**

|                |            |  |
|----------------|------------|--|
| <b>codice:</b> |            |  |
| x1    aaa      | x6    abc  |  |
| x2    aab      | x7    aca  |  |
| x3    aac      | x8    acb  |  |
| x4    aba      | x9    acc  |  |
| x5    abb      | x10    baa |  |

**E(0,1) codice binario per J:**

a 00  
b 01  
c 11

**Codifica Binaria Indiretta**

X1  00 00 00  
X2  00 00 01  
X3  00 00 11  
X4  00 01 00  
X5  00 01 01  
X6  00 01 11  
X7  00 11 00  
X8  00 11 01  
X9  00 11 11  
X10 01 00 00

---

Elementi di Informatica  
Prof. G. A. Di Lucca - Univ. del Sannio 19

### Codifica dell'Informazione

Una codifica indiretta produce parole codice di lunghezza maggiore od uguale di quella delle parole codice prodotte da una codifica diretta

$$[\log_k N] * [\log_w K] \geq [\log_w N]$$

(... log : parte intera, maggiorata di uno se ...)

... nell'esempio precedente:

$$[\log_3 10] * [\log_2 3] = 6$$

$$[\log_2 10] = 4$$

... una parola codice 'più lunga' ...  
... codifica più 'comoda' ....  
... non devo ricordare il codice per ciascun elemento sorgente, ma quello di ciascun simbolo costituente l'elemento sorgente ...

---

Elementi di Informatica  
Prof. G. A. Di Lucca - Univ. del Sannio 20

## Codice ASCII

**Codice ASCII (American Standard Code for Information Interchange):  
usato per uniformare la codifica dei caratteri**

| bit  |       | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|------|-------|-----|-----|-----|-----|-----|-----|-----|-----|
|      | esad. | 0   | 1   | 2   | 3   | 4   | 5   | 6   | 7   |
| 0000 | 0     | NUL | DLE | spz | 0   | @   | P   | '   | p   |
| 0001 | 1     | SOH | DC1 | !   | 1   | A   | Q   | a   | q   |
| 0010 | 2     | STX | DC2 | "   | 2   | B   | R   | b   | r   |
| 0011 | 3     | ETX | DC3 | #   | 3   | C   | S   | c   | s   |
| 0100 | 4     | EOT | DC4 | \$  | 4   | D   | T   | d   | t   |
| 0101 | 5     | ENQ | NAK | %   | 5   | E   | U   | e   | u   |
| 0110 | 6     | ACK | SYN | &   | 6   | F   | V   | f   | v   |
| 0111 | 7     | BEL | ETB | '   | 7   | G   | W   | g   | w   |
| 1000 | 8     | BS  | CAN | (   | 8   | H   | X   | h   | x   |
| 1001 | 9     | HT  | EM  | )   | 9   | I   | Y   | i   | y   |
| 1010 | A     | LF  | SS  | *   | :   | J   | Z   | j   | z   |
| 1011 | B     | VT  | ESC | +   | ;   | K   | [   | k   | {   |
| 1100 | C     | FF  | FS  | ,   | <   | L   | \   | l   |     |
| 1101 | D     | CR  | GS  | -   | =   | M   | ]   | m   | }   |
| 1110 | E     | SO  | RS  | .   | >   | N   | ^   | n   | ~   |
| 1111 | F     | SI  | US  | /   | ?   | O   | _   | o   | DEL |

**Esempio: Lettera 'A' => 100 0001**

## Codifica dell'Informazione

### Codici ridondanti e controllo dell'errore

**Codici binari ridondanti sono utilizzati per la *individuazione di malfunzionamenti* (guasto dei circuiti, difetti di trasmissione) che conducano a parole errate**

**In un codice ridondante, solo  $n$  delle  $\kappa^m$  parole disponibili sono lecite (dove  $n \leq \kappa^{m-1}$ ) per cui**

- se  $\delta$  non è una parola lecita, allora è *certamente* errata
- se  $\delta$  è una parola lecita, allora è *probabilmente* corretta

**... maggiore è la ridondanza maggiore è la probabilità di scoprire un errore ... nessuna certezza che una parola codice lecita non sia errata ...**

## Codifica dell'Informazione

### Un esempio: Il controllo di parità

Utilizzato per il controllo dell'errore nella trasmissione di parole di codici binari completi ( $K = 2$ ,  $n = 2^m$ )

Dal codice completo si ottiene un codice ridondante aggiungendo un bit con la seguente regola:

- 1 se gli  $m$  bit hanno un numero dispari di 1
- 0 se gli  $m$  bit hanno un numero pari di 1

... le parole lecite del nuovo codice avranno un numero pari di bit 1 ...



... possibile scoprire un solo errore ...