

1.2. La codifica delle informazioni

1.2.1. Natura dei dati e codifica

I dati da elaborare ed elaborati sono di diversa natura (numeri, caratteri, immagini, suoni, etc.) ma devono assumere una forma adatta ad essere trattata dal calcolatore. Questo, dal suo canto, tratta soltanto dati rappresentati come sequenze di bit, perché i circuiti della macchina trattano segnali a soli due valori (vedi RICHIAMO).

Il processo che trasforma un dato in una sequenza di bit è detto di *codifica*, quello che viceversa, data la sequenza di bit, ripropone il dato nella sua forma originale è detto di *decodifica*.

Con un solo bit è possibile distinguere fra due oggetti diversi, uno associato allo "0" e l'altro all' "1" e sarebbe quindi possibile codificare due "oggetti"; con 2 bit è possibile distinguere e quindi codificare 4 ($=2^2$) oggetti; con 3 bit è possibile codificare 8 ($=2^3$) oggetti e così via.

Quindi, con n bit è possibile codificare 2^n oggetti distinti: ogni combinazione di 0 ed 1 è associata ad un distinto oggetto in modo convenzionale (ad esempio attraverso una tabella) o con qualche legge aritmetica.

Il processo di codifica è diverso a seconda della natura del dato (testo, numero, immagine, filmato), come verrà sviluppato in seguito.

RICHIAMO 1.1.11 Circuiti e bit

1.2.1.1. Codifica in bit (Approfondimento)

Con un solo bit si possono distinguere due oggetti o due attributi, l'uno convenzionalmente associato allo 0, l'altro all'1 (vedi figura): se i due attributi sono ad esempio "bianco" e "nero", 0 potrebbe significare "bianco", 1 potrebbe significare "nero".

Con 2 bit è possibile realizzare 4 "parole-codice" (00,01,10,11), ottenute effettuando quello che in matematica si dice "disposizione con ripetizioni dei 2 oggetti 0 ed 1 sui 2 posti, primo e secondo": ciascuna parola-codice è associata a uno dei 4 oggetti da codificare; supposto che questi ultimi siano "Napoli", "Roma", "Milano", "Torino" si potrebbe avere ad esempio 00=Napoli, 01=Roma, 10=Milano, 11=Torino.

Con 3 bit si potrebbe avere analogamente 000=A, 001=B, 010=C, 011=D, 100=E, 101=F, 110=G, 111=H.

Con un procedimento che in matematica si dice di "induzione" si può affermare che con n bit si realizzano 2^n parole-codice e quindi la possibilità di codificare altrettanti "oggetti" o "attributi", nel significato più generale.

A esempio, con 7 bit si codificano $2^7=128$ caratteri distinti della tastiera, con 8 bit $2^8=256$ distinti "livelli di grigio" nella rappresentazione codificata di un disegno in bianco e nero, con 16 bit $2^{16}=65.536$ distinti numeri interi.

codifica ad 1 bit

| |
|---------------|
| 0 oggetto n.1 |
| 1 oggetto n.2 |

codifica a 2 bit

| |
|----------------|
| 00 oggetto n.1 |
| 01 oggetto n.2 |
| 10 oggetto n.3 |
| 11 oggetto n.4 |

codifica a 3 bit

| |
|-----------------|
| 000 oggetto n.1 |
| 001 oggetto n.2 |
| 010 oggetto n.3 |
| 011 oggetto n.4 |

codifica a n bit

| |
|--------------------|
| 0...00 oggetto n.1 |
| 0...01 oggetto n.2 |
| |
| |

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

1.2.3. Codifica dei dati numerici

Il sistema di numerazione usato dalla nostra civiltà è un sistema decimale: un numero viene scomposto in unità, decine, centinaia, e viene rappresentato da una stringa di cifre decimali (da 0 a 9): ad esempio, il numero 327 indica 7 unità, 2 decine e 3 centinaia.

Le caratteristiche degli elaboratori si adattano più agevolmente ad utilizzare un sistema di numerazione binario e pertanto l'aritmetica binaria è la più diffusa per la codifica dei numeri nei sistemi digitali: il numero è scomposto in unità, coppie, quadruple, ... e viene rappresentato da una stringa di cifre binarie (il bit, appunto, da 0 a 1): ad esempio, il numero 110 indica 0 unità, 1 coppia e 1 quadrupla e corrisponde dunque al numero che in decimale si scriverebbe 6.

Per fissare le idee, si mostra in figura la sequenza dei primi 16 numeri naturali (da 0 a 15) espressi in binario.

Il problema della rappresentazione dei numeri è più complesso di quanto finora abbiamo detto: oltre ai numeri interi e positivi, esistono altri tipi di numeri (negativi, con virgola, etc.). Tuttavia, non è qui il caso di approfondire ulteriormente l'argomento che riguarda l'esperto piuttosto che l'utente del computer.

| binario | decimale | binario | decimale | binario | decimale | binario | decimale |
|---------|----------|---------|----------|---------|----------|---------|----------|
| 0 | 0 | 100 | 4 | 1000 | 8 | 1100 | 12 |
| 1 | 1 | 101 | 5 | 1001 | 9 | 1101 | 13 |
| 10 | 2 | 110 | 6 | 1010 | 10 | 1110 | 14 |
| 11 | 3 | 111 | 7 | 1011 | 11 | 1111 | 15 |

1.2.4. Codifica di segnali analogici (e del suono)

Alcuni dati sono in origine rappresentati da segnali analogici; tipico è il caso del suono raccolto da un microfono, che riproduce la variazione nel tempo della pressione che le onde sonore esercitano sul microfono stesso. Per trattare tali dati in un computer occorre trasformarli in segnali digitali, cioè in sequenze di bit.

La caratteristica di un segnale analogico è di essere continuo nel tempo e di poter assumere, in teoria, infiniti valori in ogni istante: un segnale può essere trattato come una "funzione" o una curva che evolve nel tempo (vedi figura). Il problema della sua conversione in un segnale digitale si risolve con un *campionamento* ed una *quantizzazione*.

Il campionamento consiste nel "leggere" il valore analogico in un numero finito e discreto di istanti, rappresentando quindi la grandezza analogica in un numero finito di valori del tempo. È come tracciare sulla curva del valore analogico un insieme di segmenti verticali, ciascuno dei quali identifica un istante di campionamento (vedi figura).

La quantizzazione consiste nel rappresentare ogni campione con un valore numerico codificato in bit. Fissare il numero di bit per tale codifica, equivale a fissare un insieme di segmenti orizzontali sulla curva analogica e nel rappresentare ciascun campione con il numero posto sul segmento più vicino al valore della curva (vedi figura) e quindi un numero finito e discreto di valori.

In definitiva, sulla curva analogica è come se venisse tracciata una griglia e la grandezza viene rappresentata da una sequenza di numeri codificati in bit, uno per campione; esistono appositi teoremi e tecniche che definiscono la griglia ottimale per tale rappresentazione (un accenno vi è nell'approfondimento). È infatti intuitivo che una griglia più fitta (campioni più frequenti e più bit per rappresentare i valori) migliora l'approssimazione del sistema ma richiede maggiori risorse per rappresentare una curva (più memoria per memorizzarla e più tempo per trasmetterla). Va inoltre considerato che spingere l'approssimazione oltre certi limiti è inutile, in quanto il problema reale trattato non apprezza in generale approssimazioni troppo fini.

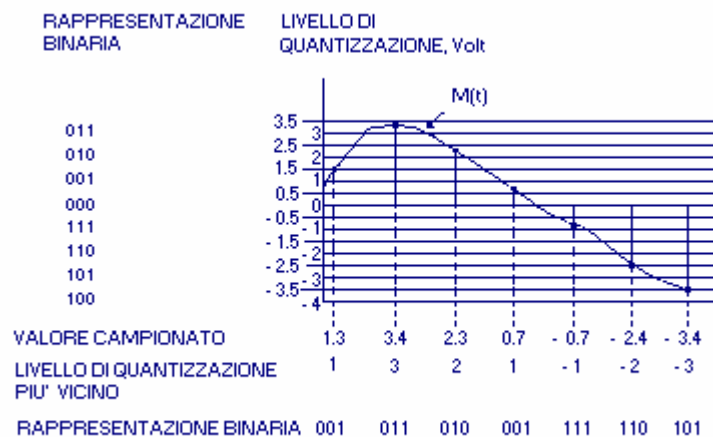


Figura: Campionamento, Quantizzazione e Codifica

1.2.4.1. Il teorema del campionamento (Approfondimento)

Ogni segnale può essere studiato dal punto di vista della frequenza (intuitivamente, ci sono segnali che variano molto spesso nel tempo, altri che variano lentamente) scomponendolo nella sovrapposizione di segnali sinusoidali a frequenza diversa (*Trasformata di Fourier*) e si osserva che tutti i segnali reali sono limitati in banda, cioè non posseggono componenti significative in potenza oltre una certa frequenza B.

Un teorema, detto *teorema del campionamento*, regola la frequenza necessaria per campionare un segnale senza perderne in contenuto informativo. Esso afferma che un segnale è completamente descritto dalla sequenza dei suoi campioni senza perdita di informazioni se questi vengono

prelevati ad una frequenza f_c maggiore o uguale al doppio della massima frequenza del segnale stesso: $f_c \geq 2B$.

Un campionamento siffatto è dunque anche reversibile, nel senso che dati i campioni si può ricostruire la curva originale senza perdere in approssimazione.

Al contrario del campionamento, la fase di quantizzazione non è reversibile e, l'approssimazione introdotta corrisponde ad una perdita di informazione che, in fase di riproduzione del segnale, si traduce in una perdita di qualità, almeno sul piano teorico. Per rimediare, è possibile aumentare il numero di bit, ma a scapito dell'occupazione di memoria; tali informazioni, poi, devono essere trasmesse e, quindi, occorre tenerne conto anche per valutare un'adeguata velocità di flusso dei dati detta *bitrate* ed espressa in bit al secondo.

Ciò, peraltro, non deve indurre a ritenere il sistema analogico più preciso di quello digitale: c'è differenza infatti fra curva analogica teorica e curva analogica reale, in quanto quest'ultima è anch'essa approssimata dall'intervento degli apparecchi che devono misurare i valori che, per quanto analogici, sono essi stessi approssimati.

1.2.5. Codifica delle immagini

Per rappresentare in bit una immagine (una scena, una fotografia, etc.) le si sovrappone idealmente un reticolo di $n \times m$ maglie; ciascuna maglia individua un *pixel* (*picture element*) e costituisce un "elemento" della stessa; l'immagine è rappresentata con l'insieme dei suoi pixel, secondo uno standard predefinito. Il numero di pixel di un'immagine è anche detto "risoluzione" di un'immagine.

Un pixel è rappresentato da un numero (espresso da più bit) che ne indica il "livello di grigio" se l'immagine è in bianco e nero, la "profondità di colore" se è a colori. Una rappresentazione di immagini in questa forma è detta in tecnologia *Raster* ed è quella usata per rappresentare una generica scena, prelevata da uno scanner, da una telecamera, oppure "catturata" dallo stesso monitor del sistema.

Lo standard *Bitmap* della Microsoft è una particolare realizzazione della Raster.

Diversa è la tecnica di rappresentazione di disegni geometrici, detta anche *vettoriale*: l'immagine è ottenuta componendo figure geometriche elementari fondamentali di differenti tipi (segmenti, poligoni, archi di circonferenze, etc.) dette primitive, ciascuna delle quali è individuata mediante appositi parametri.

In ogni caso, la rappresentazione di una immagine richiede molti bit e quindi occupa parecchio spazio in memoria. Per fissare le idee vedere l'approfondimento.

Le caratteristiche tecniche di un video sono anch'esse caratterizzate dalla risoluzione e dalla profondità di colore.

1.2.5.1. Rappresentazione delle immagini (approfondimento)

Una semplice rappresentazione in bianco e nero può prevedere 8 bit per pixel; si possono così rappresentare per ogni pixel dell'immagine $2^8 = 256$ livelli di grigio.

Lo schermo di un PC è caratterizzato da una propria *risoluzione*, cioè dal numero di pixel su di esso. Supposto di usare un reticolo di 1024 colonne e 768 righe, la rappresentazione a tutto schermo di una immagine in bianco e nero con 256 livelli di grigio richiede allora $8 \times 1024 \times 768 = 6.291.456$ bit.

Se consideriamo immagini a colori, la situazione si complica: ogni colore si ottiene combinando i tre colori primari (Rosso, Verde, Blu), ciascuno dei quali con un dato "livello di

grigio"; quindi per ogni pixel si utilizzano $3 \times 8 = 24$ bit. In questo modo si possono rappresentare $2^{24} = 16$ milioni di colori diversi per ogni pixel. Il numero di bit impiegato si dice *profondità del colore*.

Una immagine con buona risoluzione ed elevata profondità di colore può essere troppo ingombrante (richiede troppi bit) ed allora bisogna accontentarsi di una risoluzione oppure di una profondità di colore più contenute. Sono anche possibili tecniche di compressione, che però in genere ne diminuiscono le caratteristiche di qualità.

I dati *Raster* possono essere memorizzati attraverso tipologie di file che sfruttando algoritmi di compressione diversi, gravano in modo differente sul supporto di memorizzazione. Tipici formati Raster sono ad esempio, oltre al già citato Bitmap (Bmp), "Joint Photographic Exports Group (Jpeg)", "Graphic Interchange Format (Gif)", "Tagged Image File Format (Tiff)" e così via, derivanti da standard o da proposte di gruppi di lavoro internazionali.

1.2.6. Codifica di filmati

Le immagini in movimento sono trattate come una veloce successione di immagini fisse (basta guardare una vecchia pellicola di un film per rendersene conto). Si sfrutta, infatti, il fenomeno della *persistenza* delle immagini sulla retina che è in grado di ricostruire il movimento se la sequenza è sufficientemente rapida.

Per codificare immagini in movimento si usa, quindi, una codifica delle immagini fisse in "quadri" (*frame*, in inglese) e si riproducono questi ultimi l'uno dopo l'altro con sufficiente velocità, altrimenti il film procede "a scatti".

La velocità è peraltro condizionata dal numero di bit da trasmettere per ciascun frame ed una semplificazione che diminuisce il numero di bit facendo raggiungere la velocità necessaria è quella di rappresentare soltanto le variazioni fra un *frame* ed il successivo. Su questi concetti si basa il diffuso standard *MPEG* (*Moving Picture Experts Group*).

1.2.7. La compressione dei dati

Effettuare una *compressione* dei dati significa rappresentarli con un numero di bit inferiore a quello derivante dalla loro normale codifica. Talora si ricorre a questa tecnica per risparmiare spazio sui supporti di memoria (lo spazio dipende dal numero di bit da memorizzare) oppure per risparmiare tempo se i dati debbono essere trasferiti via Internet (se occorre trasmettere meno bit, la rete impiega meno tempo). In gergo, un dato compresso si dice anche *zippato*.

Un dato zippato, peraltro, non può essere usato o interpretato in tale forma, ma deve essere preventivamente *decompresso* (si dice anche *estratto*), cioè ricondotto alla sua forma originale..

Per comprimere un insieme di dati, si adoperano appositi programmi che usano algoritmi di compressione non distruttivi, consentendo cioè l'applicazione del processo inverso (in contesti diversi si applicano anche algoritmi di compressione *distruttivi*, ad esempio ad una immagini si riducono i livelli di grigio). Per utilizzare i dati compressi si adoperano allora programmi che usano l'algoritmo inverso riconducendoli alla loro codifica e dimensione originale.