La trasmissione dell’informazione - Generalità

L'infomazione codificata può subire "*manipolazioni*" o "*trattamenti*". L'***elaborazione*** matematica è una delle possibili manipolazioni dei numeri. L'informazione non numerica può essere ***ordinata*** od ***archiviata***, come avviene, per esempio, in un programma di "*database*". **L'informazione di ogni tipo può essere trasmessa**.

La **trasmissione dell'informazione** implica una ***sorgente*** che genera l'informazione da trasmettere.
Una parte autoconsistente d'informazione si definisce, ***messaggio***. Esso viene trasmesso attraverso un ***canale*** o ***mezzo*** che può essere un doppino telefonico, un cavo coassiale o una fibra ottica, oppure l'etere, se l'informazione è irradiata da una antenna verso un'altra.
Un ***codificatore*** trasforma, quando è necessario, il messaggio in sequenze di segnali adatti ad essere trasmessi lungo il mezzo scelto.
Un ***decodificatore*** trasforma i segnali ricevuti in informazione codificata nella forma originale od in una diversa per consentirne l'interpretazione.
Una unità di ***destinazione*** riceve ed interpreta i messaggi.

Intersezione tra Telecomunicazioni ed Informatica

La trasmissione dell'informazione e le tecniche necessarie per effettuarla non sono riservate alle **telecomunicazioni** a grande distanza.
Non a caso oggi si parla di ***Telematica*** come Scienza nata dal connubio tra **Telecomunicazioni** ed **Informatica**.
Quando bisogna trasferire in **tempi brevissimi grandi quantità d'informazione**, come avviene spesso tra due sottosistemi di un moderno calcolatore, possono essere necessarie tecniche di trasmissione sofisticatissime per coprire distanze dell'ordine di una decina di centimetri o anche meno.

Un semplice ***sistema di comunicazione*** è quello che collega la tastiera alla memoria di un computer. L'utente che batte il tasto è la ***sorgente***; la tastiera e la sua logica di controllo, che generano il codice ASCII corrispondente al tasto, formano il ***codificatore***. Il ***canale*** è costituito dal cavo di collegamento. Il ***decodificatore*** è un opportuno circuito integrato sulla scheda madre del computer; la ***destinazione*** è una prestabilita zona della memoria della macchina. Si tratta di un sistema di comunicazione tra i più lenti, visto che coinvolge le potenzialità fisiche di un essere umano.

Adattamento della codifica al canale

Il codificatore **trasforma l'informazione in sequenze di segnali adatti ad essere trasmessi attraverso il canale**.
Le **caratteristiche del canale** possono essere determinanti nella scelta della codifica del ***messaggio*** che, come ***informazione in movimento***, deve necessariamente essere **adattata al mezzo di comunicazione**.

In generale, quindi, l'informazione può essere disponibile in un ***codice sorgente*** che soddisfa le esigenze del sistema che la produce. Il codificatore la convertirà in un ***codice canale*** ed il decodificatore in un ***codice destinazione*** che potrebbe essere diverso dai primi due.

Un esempio perfettamente calzante è rappresentato dalle ormai storiche trasmissioni telegrafiche in **codice Morse** i cui 36 simboli (26 caratteri alfabetici e 10 cifre) sono riportate nella tabella seguente.

In realtà, oltre a quelli della tabella, esiste un ulteriore codice, quello di separazione tra i caratteri, associato ad una pausa un po più lunga di quelle tra i punti e le linee.
Si osservi che esistono codici da 2 elementi e codici da 5. I codici più lunghi sono assegnati ai simboli meno probabili.

Il codice MORSE

A	· —	J	· — — —	S	· · ·	1	· — — — —
B	— · · ·	K	— · —	T	—	2	· · — — —
C	— · — ·	L	· — · ·	U	· · —	3	· · · — —
D	— · ·	M	— —	V	· · · —	4	· · · · —
E	·	N	— ·	W	· — —	5	· · · · ·
F	· · — ·	O	— — —	X	— · · —	6	— · · · ·
G	— — ·	P	· — — ·	Y	— · — —	7	— — · · ·
H	· · · ·	Q	— — · —	Z	— — · ·	8	— — — · ·
I	· ·	R	· — ·	0	— — — — —	9	— — — — ·

Ridondanza dell’informazione

Per codificare ***n*** simboli, utilizzando tutte le possibili combinazioni dei bit sono sufficienti $\lfloor \log_2(n) \rfloor + 1$ **bit**.
E’ vero che se tutte le combinazioni dei due simboli binari sono assegnate ad un messaggio non c'e' ***ridondanza*?**
Se gli **n simboli o messaggi** sono **equiprobabili la risposta è: SI**.

Se esiste una apprezzabile **differenza di probabilità di occorrenza tra i messaggi** e si assegnano i **codici più corti a quelli che si presentano più frequentemente**, la lunghezza totale di un certo numero di messaggi può diminuire apprezzabilmente rispetto al caso di una codifica uniforme, dando luogo ad una **lunghezza media del messaggio inferiore** a quella della formula. **Vedi esempio →**

La ridondanza è importante perché **tutti i sistemi di comunicazione hanno un limite di flusso esprimibile in bit/sec**. Fissato il numero di bit trasferibile in un certo tempo, questo corrisponderà ad una **quantità d'informazione tanto maggiore quanto minore è la lunghezza media dei messaggi**. Considerazioni perfettamente simili possono essere fatte per **l'area di memoria necessaria per contenere l'informazione**.

Ridondanza dell’informazione: Esempio

Se una sorgente può generare **6 diversi messaggi** $m_1, \ldots\ldots m_6$

e le corrispondenti probabilità di generazione sono:

$p(m_1) = 15 / 32$; $p(m_2) = 9 / 32$; $p(m_3) = 3 / 32$;
$p(m_4) = 3 / 32$; $p(m_5) = 1 / 32$; $p(m_6) = 1 / 32$

e si assegnano i seguenti codici:

$m_1 = 0$; $m_2 = 10$; $m_3 = 110$; $m_4 = 1110$; $m_5 = 11110$; $m_6 = 11111$;

la lunghezza media di un messaggio risulta essere:

(15 x 1 + 9 x 2 + 3 x 3 + 3 x 4 + 1 x 5 + 1 x 5) / 32 = **2,0**

Il procedimento adottato è perfettamente trasparente; si sono assegnati **i codici con numero di bit più alto ai messaggi meno probabili**.
La **lunghezza media del messaggio, che con il codice uniforme avrebbe richiesto 3 bit** (8 combinazioni da 000 a 111), sia pure sotto particolari ipotesi, **è passata a 2 bit**.

Il *rumore* elettronico (*electronic noise*)

I sistemi di comunicazione a monte ed a valle del supporto fisico della comunicazione (cavo, doppino telefonico od altro) sono costituiti da **circuiti elettronici,** che sono per definizione soggetti al cosiddetto ***rumore bianco*** che è intimamente legato alle caratteristiche dei ***dispositivi attivi*** (***shot noise***) e ***passivi*** (***Johnson noise***) utilizzati.

Il rumore bianco può essere ridotto ma è **ineliminabile**.
Esso è presente **a tutte le frequenze** ed ha **ampiezza mediamente costante nel tempo** se non intervengono modificazioni dei parametri in gioco, per esempio la temperatura.

Esistono **altri tipi di rumore** (***deterministici*** o ***casuali***) tutti, in ultima analisi legati alla presenza di altre **apparecchiature elettriche** che li generano.

Probabilità di errore

La presenza del rumore comporta inevitabilmente la possibilità che un livello logico che codifica elettricamente uno "0" venga interpretato da uno dei circuiti elettronici come quello rappresentativo di un "1".
Questa probabilità è **tanto più grande** quanto **maggiore è il livello di rumore** (rapporto segnale/rumore) e (in primissima approssimazione) **tanto più piccola** quanto **maggiore è la differenza di tensione tra il livello "0" ed il livello "1"** (***immunità al rumore***) dei circuiti elettronici del sistema.

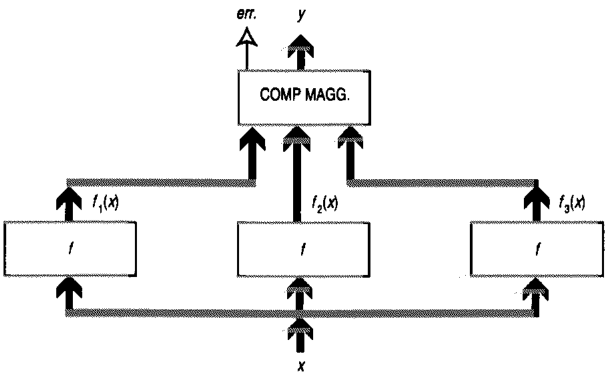
Per **ridurre la probabilità di errore** in un sistema di comunicazione si può cercare di **diminuire il rumore ed aumentare l'immunità al rumore** dei circuiti, ma si può anche fare in modo che **eventuali errori vengano rilevati** e, in qualche caso, addirittura **corretti,** in modo che essi non influenzino, se non in minima parte, le prestazioni del sistema.

Rilevazione e correzione degli errori
Il ruolo della *ridondanza*

Se c'è **ridondanza** nel codice, non tutte le combinazioni di bit sono associate ad un simbolo o carattere. L'inversione di un "1" con uno "0" può portare ad una **combinazione non utilizzata** il cui riconoscimento equivale ad una **rilevazione di errore**.
Alla categoria dei codici ridondanti appartiene, per esempio, un **particolare codice BCD** in cui i simboli del sistema decimale (0,...9) sono rappresentati con 5 bit, utilizzando esclusivamente le combinazioni di bit che hanno due soli "1" (o "0") su cinque; detto perciò codice **"*due su cinque*"**. E' ovvio che la presenza di una combinazione di bit con tre "1", invece che due, è un chiaro indice di errore.
La ridondanza che sembrava un male da estirpare con decisione viene introdotta di proposito, ma in maniera oculata.

La **ridondanza** può essere introdotta in un sistema anche a **livello strutturale** adoperando la ***logica maggioritaria***.

Logica maggioritaria



Il bit d'errore sarà **attivato (livello "1")** soltanto se tra i tre ingressi del comparatore **non ce ne sono due uguali:** $f_1(x) \neq f_2(x) \neq f_3(x)$.

In **tutti gli altri casi:** ($f_1(x) = f_2(x) \neq f_3(x)$; $f_1(x) \neq f_2(x) = f_3(x)$; $f_1(x) = f_3(x) \neq f_2(x)$), **il bit di errore rimarrà a "0"** e sarà sempre preso come risultato, **quello dato da almeno due delle unità di elaborazione**.

Controllo di parità

Il controllo di parità è una tecnica efficace per il rilevamento di un errore. E' molto meno costosa della logica maggioritaria, ma non consente la correzione degli errori.

Consiste nell'**associare ad ogni gruppo di bit un "1" se il numero di "1" presenti è dispari, o uno "0" se il numero di "1" è pari**. Il risultato è che il gruppo di bit di partenza più il bit aggiunto debbono avere sempre un numero di "1" pari (***parità pari***).

Se si fa la scelta duale si ha una ***parità dispari.***
Quest'ultima scelta è preferibile perché nel caso che il gruppo di bit sia di tutti "0" ci sarà **al valore "1"** almeno **il bit di parità**, evitando la trasmissione di una **combinazione di tutti "0"** che potrebbe essere facilmente simulata da un guasto.

In caso di **errore singolo** (un "1" che diventa "0" o viceversa) un semplice controllo consente di **rilevare l'errore**, che, però diventa **irriconoscibile** nel caso di **doppio errore**, indipendentemente dal tipo di errore.

Doppio controllo di parità

messaggio	controllo longitud.
00100000	0
10100111	0
01000001	1
10001010	0
00110010	0
01010101	1
00011011	1
01100101	1
10101010	

controllo trasversale

(a) testo originale

messaggio	controllo longitud.
00100000	0
10100111	0
01000001	1
10001010	0
001$\boxed{0}$0010	0
01010101	1
00011011	1
01100101	1
10101010	

controllo trasversale

(b) 1 errore rilevabile

messaggio	controllo longitud.
00100000	0
1010$\boxed{0}$011	$\boxed{1}$
01000001	1
10001010	0
00100010	1
01010101	1
00011011	1
01100101	1
10101$\boxed{1}$10	

controllo trasversale

(c) 3 errori non rilevabili

Codici ciclici

Solo rilevazione di errore senza correzione né localizzazione. Associano alla sequenza di bit da trasmettere **una sequenza di bit di controllo** ottenuta **dividendo il polinomio p(x)** per un **assegnato polinomio G(x)** (***polinomio generatore)***. La stringa di **bit di controllo** si ricava dai coefficienti del **polinomio r(x), resto della divisione**.
Si effettua una ***divisione binaria modulo 2*** che trasforma $\mathbf{10_2}$ in **0** e $\mathbf{11_2}$ in **1**.

- Nessun significato matematico.
- Il resto è univocamente determinato per una assegnata stringa ed un dato polinomio generatore
- Il resto ha un grado tanto minore del polinomio di partenza, quanto più elevato è il grado del polinomio generatore (pochi bit di controllo).

L'unità **ricevente** ripete l'operazione e **confronta i coefficienti del polinomio resto, r'(x) con la stringa di controllo**. Se **r'(x) risulta diverso da r(x)** viene segnalato **errore** e non c'è nient'altro da fare che **ripetere la trasmissione**. Se il controllo continua a fallire, il sistema si deve bloccare

Uso tipico: trasmissioni tra disco fisso e memoria dei sistemi di elaborazione

Esempio di codice ciclico

$$
\begin{array}{cccccccc|l}
x^7 & +x^6 & & +x^4 & & & +x & +1 & x^4+x^2+1 \\
\cline{9-9}
-x^7 & & -x^5 & & -x^3 & & & & x^3+x^2+x \\
\cline{1-8}
 & +x^6 & +x^5 & +x^4 & +x^3 & & +x & +1 & \\
 & -x^6 & & -x^4 & & -x^2 & & & \\
\cline{1-8}
 & & +x^5 & & +x^3 & +x^2 & +x & +1 & \\
 & & -x^5 & & -x^3 & & -x & & \\
\cline{1-8}
 & & & & & +x^2 & & +1 &
\end{array}
$$

- La stringa da trasmettere è **11010011**
- Il polinomio generatore è: **G(x) =** $\mathbf{x^4 + x^2 + 1}$.
- Il polinomio resto risulta: **r(x) =** $\mathbf{x^2+1}$
- La stringa di controllo da trasmettere è: **0101**.

Essendo il polinomio p(x) di ordine 7 e quello generatore di ordine 4, il resto è di ordine non superiore a 3 e:

la stringa di controllo contiene 4 bit.

Codici di Hamming

Usano la ridondanza per fare in modo che due qualsiasi parole distinte differiscono fra loro per almeno ***h*** bit, dove ***h*** è la ***cosiddetta distanza di Hamming***. I codici non ridondanti hanno $h = 1$, il codice con bit di parità $h = 2$; **il più semplice codice di Hamming è quello con $h = 3$.**

Il **numero dei bit di controllo** necessari per una parola di **n bit** si ricava dalla condizione:

$$2^k \geq n+k$$

Il **numero totale di bit della stringa** passa da **n** a **n +k**.

I k bit di controllo debbono essere posizionati in maniera opportuna negli n + k bit della nuova stringa da trasmettere.
Assegniamo a ciascuna posizione della stringa un indice ***i*** **(1, 2, .. n + k).**
I bit di controllo $p_0, p_1, p_2, \ldots p_{k-1}$ debbono essere associati agli indici nel cui codice binario c'è **un solo "1" di peso** rispettivamente, $2^0, 2^1, \ldots, 2^{k-1}$

Ciascuno dei bit di controllo garantisce l'integrità di se stesso e di tutti i bit della stringa il cui indice in binario ha un "1" di peso corrispondente alla potenza del 2 associata al pedice del bit stesso ($p_0 \rightarrow 2^0$, $p_1 \rightarrow 2^1$, ... $p_{k-1} \rightarrow 2^{k-1}$).

Codice di Hamming (h=3)per una stringa di 8 bit

Per una stringa di **8 bit,**

$b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7,$

occorrono **4 bit di controllo,**

$p_0, p_1, p_2, p_3.$

La successione da trasmettere é:

$p_0, p_1, b_0, p_2, b_1, b_2, b_3, p_3, b_4, b_5, b_6, b_7$

La specifica funzione dei bit di controllo, che entra in gioco in caso di errore, è che **ciascuno controlla la parità dei bit**:

p_0: $p_0, b_0, b_1, b_3, b_4, b_6$;
p_1: $p_1, b_0, b_2, b_3, b_5, b_6$;
p_2: p_2, b_1, b_2, b_3, b_7
p_3: p_3, b_4, b_5, b_6, b_7

	2^3	2^2	2^1	2^0
p_0	0	0	0	$\boxed{1}$
p_1	0	0	$\boxed{1}$	0
b_0	0	0	1	1
p_2	0	$\boxed{1}$	0	0
b_1	0	1	0	1
b_2	0	1	1	0
b_3	0	1	1	1
p_3	$\boxed{1}$	0	0	0
b_4	1	0	0	1
b_5	1	0	1	0
b_6	1	0	1	1
b_7	1	1	0	0

Esempio pratico

b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7
0	0	1	1	1	1	0	1

La successione da trasmettere è:
p_0, p_1, b_0, p_2, b_1, b_2, b_3, p_3, b_4, b_5, b_6, b_7

I bit di controllo sono:

p_0 = 1 perché tra i bit da esso controllati ci sono "1" in numero pari (2), b_3 e b_4;
p_1 = 0 perché gli "1" sono 3, b_2; b_3 e b_5;
p_2 = 0 perché gli "1" tra i suoi bit controllati sono 3, b_2, b_3, e b_7;
p_3 = 0 perché gli "1" tra i suoi bit controllati sono 3, b_4, b_5, e b_7.

La stringa da trasmettere sarà pertanto: **100001101101** ;

Se viene ricevuta la successione: **100101101101,** per effetto di un errore sul quarto bit che corrisponde a **p_2** , dopo il controllo di parità, si troverà **$p'_2 = 1 \neq p_2$** e, poiché tutti gli altri bit che contribuiscono alla parità, **b_1, b_2, b_3, b_7** coincidono, è evidente che l'errore è avvenuto su p_2 che sarà riportato a "0"