Codici correttori di errori

Mauro Biliotti

La teoria dei codici correttori di errori nasce in risposta al problema pratico di realizzare la trasmissione di informazioni, codificate in forma digitale, in maniera tale che eventuali errori che dovessero alterare il messaggio durante la trasmissione possano essere scoperti e corretti in fase di ricezione.

In generale un messaggio puo' essere riguardato come una sequenza di blocchi di simboli. Ciascun blocco e' costituito da una sequenza finita di simboli 0 e 1. Una sequenza finita di simboli 0 e 1 e' detta *parola*. Per comodita', spesso si suppone che ogni parola abbia la stessa lunghezza n (riguardata come sequenza).

Un *codice* (di lunghezza n) e' un insieme di parole (di lunghezza n).

Si puo' pensare alle parole del codice come alle sequenze di simboli 0 e 1 che hanno un "significato" nel linguaggio considerato. Un messaggio e' una sequenza di parole del codice. Per poter assolvere al compito di correggere gli errori intervenuti nella trasmissione di alcune parole, il codice usato deve avere particolari proprieta', come vedremo nel seguito.

Iniziamo con un esempio. Supponiamo di usare parole di lunghezza $n = 4$. Con i simboli 0 e 1 si possono costruire $2^4 = 16$ sequenze distinte di 4 elementi, quindi 16 parole distinte. Un codice di lunghezza 4 che consista di tutte le 16 parole di lunghezza 4 non permette alcuna correzione di errori. Infatti qualsiasi sequenza di 4 simboli 0 e 1 ricevuta risultera' una parola del codice e sara' impossibile sapere se la parola originariamente trasmessa era quella ricevuta, oppure un'altra.

Supponiamo che 16 sia il numero di parole che occorrono per compilare i vari messaggi, ossia che il codice debba consistere di 16 parole, ma che si voglia cercare di eliminare

l'inconveniente appena riscontrato. L'idea e' di aggiungere una specifica coda, ad esempio di *3* digits, a ciascuna delle *16* parole di *4* digits, facendole cosi' diventare sequenze di lunghezza *7*. Ad esempio la parola:

1 0 0 1

potrebbe diventare

1 0 0 1 1 0 1

In tal modo, fra tutte le $2^7 = 128$ sequenze di lunghezza *7*, solo *16* saranno *parole del codice* e quindi riconoscibili fra le altre.

Un processo di comunicazione e' illustrato nella Tab. A.

Si potrebbe dire che, in linea teorica, il principale problema della teoria dei codici consista nello studiare i metodi per aggiungere code il piu' corte possibile, le quali consentano di correggere piu' errori possibile. In pratica, ragioni di semplicita' di codifica e di decodifica possono essere prevalenti nella scelta di un determinato codice.

L'idea di definire una distanza fra le parole.

Siano

$$\underline{u} = (a_1,\ldots,a_n) \qquad a_i \in \{0,1\}$$

$$\underline{v} = (b_1,\ldots,b_n) \qquad b_i \in \{0,1\} \ .$$

Definiamo la *distanza di Hamming* $d(\underline{u},\underline{v})$ fra $\underline{u}$ e $\underline{v}$ nel modo seguente:

$d(\underline{u},\underline{v})$ *e' uguale al numero delle posizioni* i *nelle quali* $a_i \neq b_i$.

E' facile verificare che:

(1) $d(\underline{u},\underline{u}) = 0$

(2) $d(\underline{u},\underline{v}) = d(\underline{v},\underline{u})$

(3) $d(\underline{u},\underline{v}) + d(\underline{v},\underline{z}) \geq d(\underline{u},\underline{z})$.

Nell'insieme di tutte le parole di lunghezza *n* possiamo definire delle "sfere" di dato centro $\underline{u}$ e dato raggio *r* sfruttando la distanza introdotta. Precisamente:

$$S_r(\underline{u}) = \{ \underline{v} : d(\underline{u},\underline{v}) \leq r \}.$$

Supponiamo di scegliere un insieme C di parole di lunghezza n tali che le sfere di centro una parola in C e raggio r siano a due a due disgiunte, come nella Tab. B. Se nel trasmettere una parola $\underline{u}$ di C si commette un numero di errori $\leq r$, si riceve una parola $\underline{v}$ che cade dentro una sfera (e una sola), precisamente quella di centro $\underline{u}$. Decodificando $\underline{v}$ come la parola $\underline{u}$ centro di tale sfera si ritrova quindi la parola originariamente trasmessa. *E' stato cosi' corretto automaticamente l'errore di trasmissione.*

Per capire quale sia il massimo numero di errori che un dato codice C puo' correggere, occorre determinare il massimo fra i numeri r, tali che le sfere con centro una parola di C e raggio r siano a due a due disgiunte. Si indichi tale massimo con t. Si procede nel modo seguente:

- per ogni coppia di parole distinte di C se ne calcola la distanza e fra tutte le distanze cosi' ottenute si prende la minore (vi possono essere coppie di parole che hanno la stessa distanza, ovviamente). Tale distanza si chiama *minima distanza del codice* C e si indica generalmente con d.

E' un facile esercizio provare che $t = [(d-1)/2]$ ove con $[(d-1)/2]$ si denota il massimo intero $\leq (d-1)/2$ ($(d-1)/2$ se d e' dispari e $(d-2)/2$ se d e' pari).

L'idea di utilizzare i gruppi (o gli spazi vettoriali).

Indichiamo con V l'insieme di tutte le parole di lunghezza n. In V posso introdurre l'operazione di somma fra parole. Se

$$\underline{u} = (a_1,\ldots,a_i,\ldots,a_n) \qquad a_i \in \{0,1\}$$

$$\underline{v} = (b_1,\ldots,b_i,\ldots,b_n) \qquad b_i \in \{0,1\}$$

si definisce

$$\underline{u} + \underline{v} = (a_1 + b_1,\ldots,a_i + b_i,\ldots,a_n + b_n)$$

ove la somma $a_i + b_i$, $i = 1,\ldots,n$, e' fatta modulo 2.

Ad esempio, se $\underline{u} = (1,0,0,1)$ e $\underline{v} = (1,1,0,1)$ e'

$$\underline{u} + \underline{v} = (0,1,0,0)$$

Invece di prendere come codice un qualsiasi sottoinsieme di V e' piu' conveniente (ne vedremo il motivo) prendere un sottoinsieme C di V che rispetto alla operazione sopra definita risulti *un gruppo* o, come si usa dire usualmente, un *codice lineare*.

Data una parola $\underline{u}$ di V definiamo *peso* di $\underline{u} = (a_1,\ldots,a_i,\ldots,a_n)$ (il peso di $\underline{u}$ si denota con $wt(\underline{u})$, ove wt e' abbreviazione della parola inglese weight, peso) il numero delle posizioni i tali che $a_i = 1$. Dato un codice C, sottoinsieme di V, definiamo il *minimo peso* di C come il minimo dei pesi delle parole non nulle di C. Ebbene, *se* C *e' un codice lineare allora minimo peso e minima distanza in* C *coincidono.*

Vedere cio' e' molto semplice. Infatti date due qualunque parole $\underline{u}$ e $\underline{v}$ in V e' $d(\underline{u},\underline{v}) = wt(\underline{u}-\underline{v})$ (la dimostrazione e' un esercizio). Se il codice C e' lineare, cioe' un sottogruppo di C, quando $\underline{u}$ e $\underline{v}$ sono in C, anche $\underline{u}-\underline{v}$ e' in C e quindi l'insieme delle distanze fra tutte le possibili coppie di parole di C coincide con l'insieme dei pesi di tutte le parole di C. Ovviamente anche i minimi dei due insiemi coincidono e quindi minimo peso e minima distanza di C coincidono.

Quale e' il vantaggio?. Supponiamo ad esempio che il codice C abbia 2^{10} parole. Tutte le possibili coppie di parole di C sono allora 2^{20} e quindi per calcolare la minima distanza di C occorre fare 2^{20} prove, ma per calcolare il minimo peso di C bastano 2^{10} prove!.

Come si puo' costruire un codice lineare?. L'idea piu' semplice consiste nel fissare in V un certo insieme B di parole a due a due distinte e tali che nessuna parola di B possa ottenersi come somma di altre parole di B. Le parole di C sono la parola nulla, tutte le parole di B e tutte le possibili somme di due o piu' parole di B. E' facile verificare che il codice C cosi'

ottenuto e' lineare. Non e' neppure difficile provare che se B e' costituito da k parole, C e' costituito da 2^k parole.

Le parole di B, elencate in colonna, costituiscono una matrice, denotata spesso in letteratura con la lettera G, la quale prende in nome di *matrice dei generatori* del codice. Un esempio, che riporta un codice famoso, e' nella Tab. C (si noti che *7* indica la lunghezza delle parole, mentre *4* indica il numero delle parole che generano il codice, cioe' la cardinalita' di B).

Un altro vantaggio dell'uso dei codici lineari si ha in fase di decodifica. Ne accenneremo brevemente.

Supponiamo di costruire una tabella nel modo seguente:

- Nella prima riga poniamo in sequenza le parole del codice con la sola condizione che al primo posto vi sia la parola nulla ($\underline{0} = (0,\ldots,0)$).
- Fra le parole di V non in C scegliamone una di peso minimo, sia essa $\underline{e}_1$, e creiamo la seconda riga della tabella sommando $\underline{e}_1$ a ciascuna delle parole della prima riga. Per intendersi, se in una certa posizione della prima riga si trova la parola $\underline{u}$, nella seconda riga nella stessa posizione di $\underline{u}$ (cioe' sotto $\underline{u}$) scriveremo $\underline{u} + \underline{e}_1$.
- Fra le parole di V che non compaiono nelle prime due righe scegliamone una di peso minimo, sia essa $\underline{e}_2$, e creiamo la terza riga della tabella sommando $\underline{e}_2$ a ciascuna delle parole della prima riga.
- Iteriamo il procedimento fino ad aver esaurito tutte le parole di V.

Un esempio e' fornito nella Tab. D.

La tabella appena costruita prende il nome di *tabella standard* del codice.

Nella tabella standard ogni parola di V compare una ed una sola volta. Infatti, per chi ha un minimo di confidenza con la teoria dei gruppi, le righe della tabella standard altro non sono che i laterali di C in V.

Gli elementi $\underline{e}_i$ della prima colonna della tabella standard

prendono il nome di *leaders di laterale*. Si noti che il leader di laterale $\underline{e}_i$ altro non e' che l'errore, ossia la differenza, fra la parola che si trova in una data posizione nella riga i-esima e la parola del codice che sta ad essa in testa nella prima riga.

Si puo' provare che se un codice lineare C e' capace di correggere t errori una riga che contenga una parola di V di peso s con $s \leq t$ non ne contiene altre di peso $\leq t$ e quindi tale parola e' leader di laterale in quella riga.

Quando si riceve una parola $\underline{v}$, tale parola puo' essere univocamente individuata nella tabella standard. Se, per caso, nella trasmissione sono stati incorporati piu' di t errori, ossia si e' ecceduta la capacita' di correzione del codice, non e' possibile ricostruire la parola originariamente trasmessa, ma se non sono intervenuti piu' di t errori la parola $\underline{v}$ cadra' necessariamente in una riga il cui leader $\underline{e}_i$ ha peso $\leq t$ e bastera' allora sottrarre $\underline{e}_i$ a $\underline{v}$ per ottenere la parola $\underline{u}$ originariamente trasmessa. In effetti, per come e' costruita la tabella standard, $\underline{u}$ e' la parola che sta in testa alla colonna in cui si trova $\underline{v}$. Si comprende come cio' sia utile per una decodifica automatica mediante un apposito apparato.

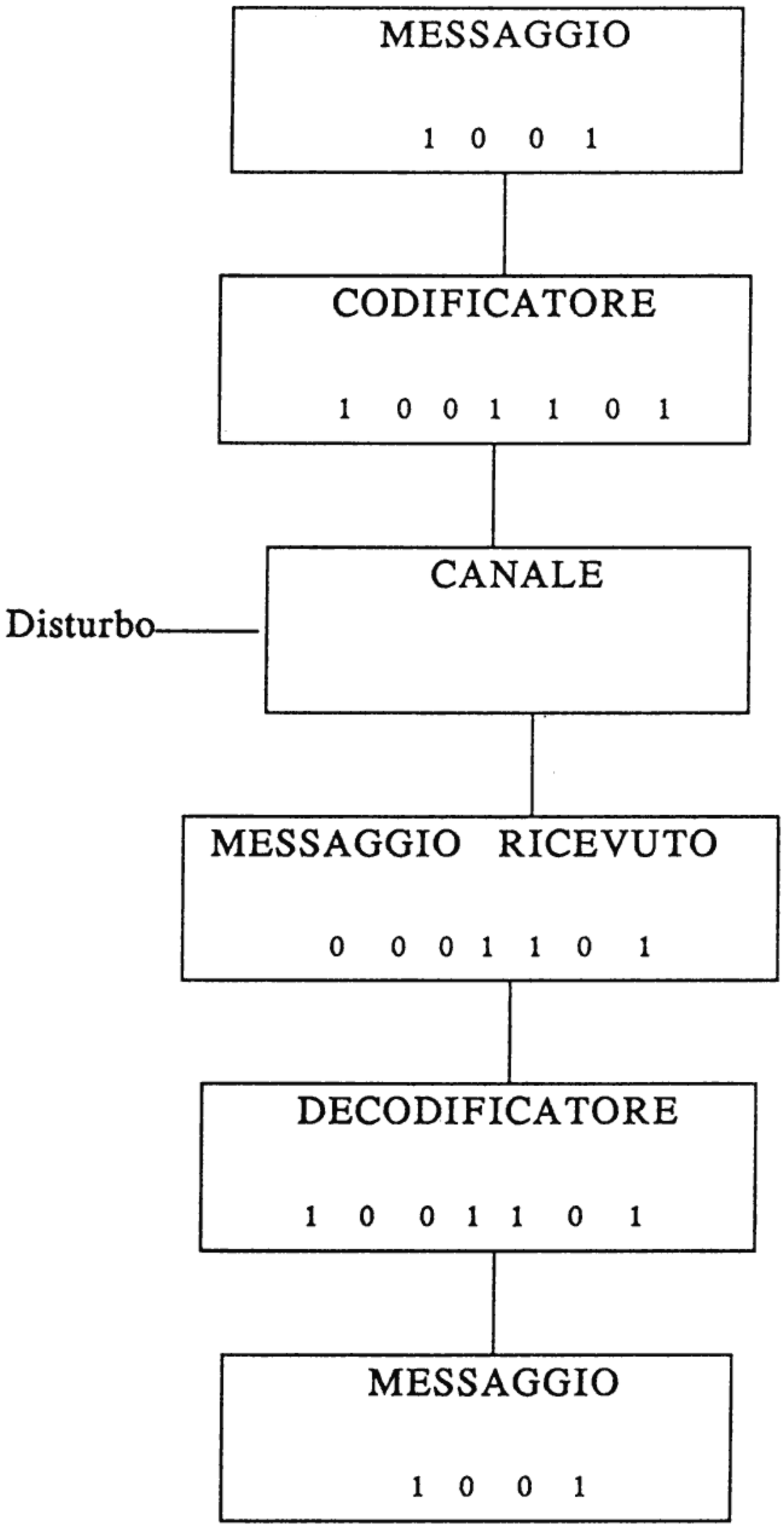
Il metodo di decodifica mediante tabella standard ha il grosso difetto di dover mettere tutte le parole di V nella memoria della macchina. In effetti sono stati elaborati sistemi di decodifica molto piu' economici e veloci, la cui trattazione esula dai limiti di questa elementare introduzione.

Un'osservazione.

Non sempre i codici sono utilizzati per correggere gli errori, ma sovente sono utilizzati solo per individuare gli errori, pur non possedendo le capacita' per correggerli. L'esempio piu' classico consiste nell'aggiunta ad ogni sequenza di 0 e 1 che si vuole trasmettere un bit dato dalla somma modulo 2 degli elementi della sequenza. Tale bit (0 o 1) viene

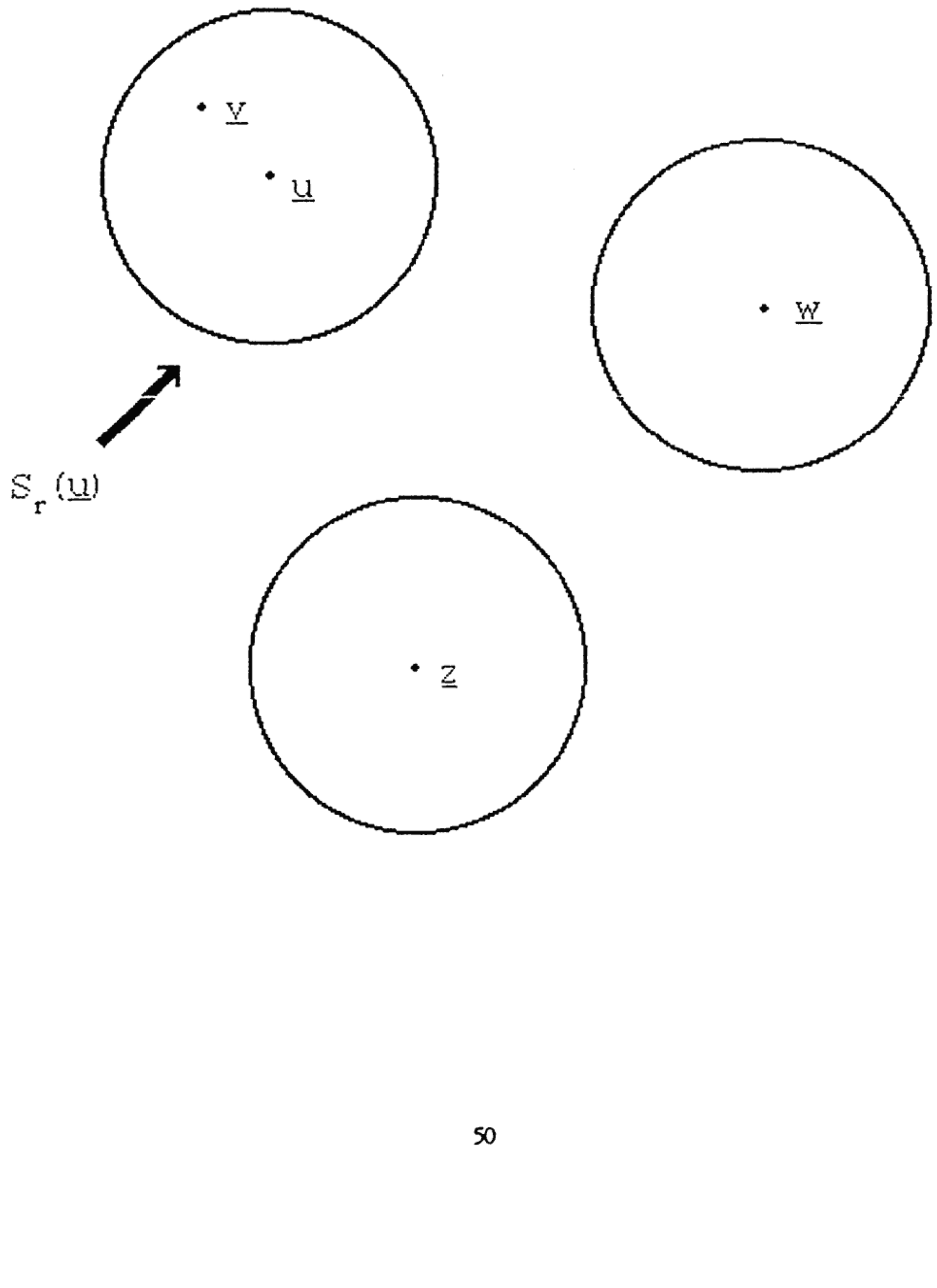
usualmente chiamato *bit di controllo di parita'*. Se nella sequenza si verifica un singolo errore il bit di controllo di parita' permette di accertarsi che si e' verificato l'errore, ma non di correggerlo. Due errori non vengono neppure individuati. In compenso la "coda" e' corta, un solo bit aggiuntivo!. L'aggiunta del bit di controllo di parita' e' ad esempio usata in certe fasi di elaborazione nei computers.

Diagramma di un sistema di comunicazione **Tab. A**



Tab. B

Sfere di centro una parola del codice e raggio r



Tab. C

Matrice dei generatori del (7,4) - codice di Hamming

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Tab. D

Decodifica mediante la tabella standard

Matrice che definisce il codice (un (5,2)-codice)

$$G = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Tabella standard

	Leaders			
Parole del codice	0 0 0 0 0	1 0 1 0 1	0 1 0 1 1	1 1 1 1 0
Laterali	1 0 0 0 0	0 0 1 0 1	1 1 0 1 1	0 1 1 1 0
	0 1 0 0 0	1 1 1 0 1	0 0 0 1 1	1 0 1 1 0
	0 0 1 0 0	1 0 0 0 1	0 1 1 1 1	1 1 0 1 0
	0 0 0 1 0	1 0 1 1 1	0 1 0 0 1	1 1 1 0 0
	0 0 0 0 1	1 0 1 0 0	0 1 0 1 0	1 1 1 1 1
	1 1 0 0 0	0 1 1 0 1	1 0 0 1 1	0 0 1 1 0
	1 0 0 1 0	0 0 1 1 1	1 1 0 0 1	0 1 1 0 0