Teoria dell'Informazione/RTI **Anno Accademico 2022–2023**

Lezione 18

Ugo Vaccaro

Nella lezione scorsa abbiamo affrontato il seguente problema che può essere, informalmente, così descritto: supponiamo di avere una sequenza binaria $\mathbf{x} = x_1 x_2 \dots x_n \in \{0,1\}^n$ che viene trasmessa su di un canale di trasmissione non affidabile, per cui in fase di ricezione si può ricevere una sequenza $\mathbf{y} = y_1 y_2 \dots y_n \in \{0,1\}^n$, tale che $x_i \neq y_i$, per un certo numero di indici $i \in \{1, 2, \dots, n\}$. Analogamente, potremmo aver memorizzato $\mathbf{x}$ su di un supporto soggetto a malfunzionamenti, per cui in fase di lettura ci ritroviamo una sequenza $\mathbf{y} = y_1 y_2 \dots y_n \in \{0,1\}^n$, tale che $x_i \neq y_i$, per un dato numero di indici $i \in \{1, 2, \dots, n\}$. *Come possiamo risalire alla* $\mathbf{x}$*, conoscendo solo la* $\mathbf{y}$*?*

La soluzione che abbiamo derivato non suggerisce, però, nessun metodo efficace per risolvere, in pratica, il problema. L'obiettivo di questa lezione è di presentare un algoritmo per la correzione di errori che occorrono durante la trasmission (o memorizzazione) dati.

Abbiamo innanzitutto bisogno di ricordare alcune proprietà di base dei campi finiti. Ricordiamo che una struttura algebrica $(\mathbb{F}, \times, +)$ dotata di operazione di addizione e moltiplicazione, è un campo finito se, denotando con 0 l'elemento neutro rispetto all'addizione, vale che:

1. $\forall a, b \in \mathbb{F}$ vale che $a \times b \in \mathbb{F}$;
2. $(\mathbb{F} \setminus \{0\}, \times)$ è un gruppo commutativo;
3. $(\mathbb{F}, +)$ è un gruppo commutativo;
4. $\forall a, b, c \in \mathbb{F}, \quad (a + b) \times c = a \times c + b \times c$.

Un classico esempio di campo finito è il campo $F_p = \{0, 1, \dots, p-1\}$, dove p è un numero primo, e le operazioni di addizione e moltiplicazione sono le solite operazioni di addizione e moltiplicazioni modulo p. É possibile costruire campi finiti con p^m elementi, per ogni primo p e numero naturale $m \geq 1$. Diamo un'idea su come ciò sia possibile.

Denotiamo con F_2 il campo con elementi $\{0, 1\}$, munito dell'operazione di addizione e moltiplicazione modulo 2. Con F_{2^m} denoteremo il campo composto da tutti i 2^m vettori binari di lunghezza m. L'operazione di addizione in F_{2^m} è la solita operazione di addizione vettoriale, ovverosia se $\mathbf{a}, \mathbf{b} \in F_{2^m}$, allora $\mathbf{a} + \mathbf{b} = \mathbf{c} \in F_{2^m}$, ed il vettore $\mathbf{c}$ ha come i-esima componente la somma (modulo 2) della i-esima componente di $\mathbf{a}$ e $\mathbf{b}$.

Per definire un'opportuna operazione di moltiplicazione in F_{2^m}, è utile vedere gli elementi (vettori) di F_{2^m} come polinomi di grado al più $m-1$. Ovvero, dato il vettore $\mathbf{a} = (a_0, a_1 \dots, a_{m-1})$, esso lo vedremo come il polinomio $a(x) = a_0 + a_1 x + \dots + a_{m-1} x^{m-1}$. Un polinomio $p(x)$ di grado > 1 a coefficienti in F_2 è detto *polinomio irriducibile* su F_2 se esso non è divisibile per nessun altro polinomio di grado > 1 a coefficienti in F_2. Si può vedere che un polinomio irriducibile su F_2 necessariamente divide il polinomio $x^{2^m-1} + 1$. Un polinomio $p(x)$ a coefficienti in F_2 è detto *polinomio primitivo* se esso è un polinomio irriducibile (quindi divide $x^{2^m-1} + 1$), e *non* divide $x^n + 1$ per ogni $n < 2^m - 1$. Polinomi primitivi esistono e sono noti per ogni m. Ad esempio, i seguenti polinomi sono primitivi su F_2: $m = 3, x^3 + x + 1$ $m = 4, x^4 + x + 1$ $m = 5, x^5 + x^2 + 1$. A questo punto, definiamo la seguente regola di moltiplicazione tra vettori in F_{2^m}. Fissato un polinomio primitivo $p(x) = p_m x^m + \dots + p_0$, per ogni coppia di dati $\mathbf{a}, \mathbf{b} \in F_{2^m}$, porremo

$$\mathbf{a} \times \mathbf{b} = \mathbf{c} \text{ se e solo se } c(x) = a(x)b(x) \bmod p(x). \tag{1}$$

Si può vedere facilmente che in questo modo l'insieme $F_{2^m} - \{\mathbf{0}\}$ è un gruppo commutativo. Ne segue che F_{2^m} munito dell'operazione di addizione prima definita e della operazione di moltiplicazione (1) è un campo finito.

Sia ora $p(x) = p_0 + p_1 x + \ldots + p_m x^m$ un polinomio primitivo di grado m su F_2, e sia α una radice di $p(x)$. Ogni radice di $p(x)$ viene detta *radice primitiva*. Poichè per definizione $p(x)$ divide $x^{2^m-1}+1$, otteniamo che $x^{2^m-1}+1 = q(x)p(x)$, per qualche polinomio $q(x)$, da cui $\alpha^{2^m-1}+1 = q(\alpha)p(\alpha) = 0$, e quindi $\alpha^{2^m-1} = 1$. Ciò che è interessante, è che è possibile provare che le successive potenze $1 = \alpha^0, \ldots, \alpha^{2^m-2}$ sono *tutti* gli elementi di $V_m(F_2)$, tranne $\mathbf{0}$, in altri termini α è un generatore del gruppo (moltiplicativo) $F_{2^m} - \{\mathbf{0}\}$.

Vediamo un esempio. Sia $m = 3$, sia dato il polinomio primitivo $x^3 + x + 1$ e sia α radice primitiva, per cui $\alpha^3 + \alpha + 1 = 0$, il che è equivalente a dire che $\alpha^3 = \alpha + 1$. Il campo F_{2^3} ha ovviamente 8 elementi, che possono essere rappresentati come 8 potenze di α, oppure come 8 polinomi su F_2 di grado al più 2, oppure come vettori binari a tre componenti, come mostrato nella tabella di seguito:

α^i		polinomio (grado ≤ 2)	vettore (a_2, a_1, a_0)
$\alpha^0 = 1$	α^0	1	001
α^1	α	x	010
α^2	α^2	x^2	100
α^3	$\alpha + 1$	$x + 1$	011
α^4	$\alpha(\alpha + 1) = \alpha^2 + \alpha$	$x^2 + x$	110
α^5	$\alpha(\alpha^2 + \alpha) = \alpha^2 + \alpha + 1$	$x^2 + x + 1$	111
α^6	$\alpha(\alpha^2 + \alpha + 1) = \alpha^2 + 1$	$x^2 + 1$	101
α^7	$\alpha(\alpha^2 + 1) = 1$		

É inoltre semplice calcolare l'inverso di un elemento attraverso la identità $\alpha^{-i} = \alpha^{n-i}$, dove $n = 2^m - 1$. Utilizzando una rappresentazione siffatta, possiamo effettuare in maniera semplice le operazioni di addizione e moltiplicazione nel campo F_{2^3}. Ad esempio, se volessimo effettuare la moltiplicazione dell'elemento 111 per l'elemento 101, possiamo notare che 111 corrisponde a α^5 mentre 101 corrisponde ad α^6, per cui

$$111 \times 101 = \alpha^5 \times \alpha^6 = \alpha^{11} = \alpha^7 \times \alpha^4 = \alpha^4 = 011.$$

Analogamente, se volessimo "dividere" il vettore 111 per il vettore 101 avremmo

$$\frac{111}{101} = \frac{\alpha^6}{\alpha^5} = \alpha^6 \times \alpha^{-5} = \alpha^6 \times \alpha^{7-5} = \alpha^6 \times \alpha^2 = \alpha^8 = \alpha = 010.$$

Viceversa, se dovessimo addizionare α^5 con α^6, possiamo passare alla loro rappresentazione vettoriale nell'ultima colonna a destra, e agevolmente dedurre che $\alpha^5 + \alpha^6 = \alpha$.

Allo stesso identico modo, se fossimo partiti da un campo F_p arbitrario, avremmo potuto costruire un campo con p^m elementi. Il fatto interessante è che *ogni* campo finito $\mathbb{F}_q$ con q elementi è di questo tipo, ovvero è isomorfo ad un campo di polinomi con p^m elementi, per opportuni p primo e m numero naturale.

Ritorniamo al problema di correggere eventuali errori occorsi durante la trasmissione (o memorizzazione) di dati. In linea generale, ciò che occorre fare è di costruire una regola di codifica $c : \{0,1\}^k \mapsto \{0,1\}^n$, $n > k$, in modo che gli $(n-k)$ extra-bits che abbiamo aggiunto alle generica sequenza $\mathbf{x} \in \{0,1\}^k$ ci aiutino nella correzione. Un esempio chiarirà la questione.

Supponiamo di voler trasmettere una arbitraria sequenza $\mathbf{x} = x_1x_2x_3x_4 \in \{0,1\}^4$. Ci calcoliamo altri 3 bit x_5, x_6, x_7, ottenuti nel modo seguente. Dati x_1, x_2, x_3, x_4, imponiamo che x_5, x_6, x_7 soddisfino le seguenti equazioni:

$$\begin{aligned} x_1 + x_2 + x_4 + x_5 &= 0 \\ x_1 + x_4 + x_3 + x_6 &= 0 \\ x_2 + x_4 + x_3 + x_7 &= 0 \end{aligned} \tag{2}$$

Quindi la nostra codifica $c : \{0,1\}^4 \mapsto \{0,1\}^7$ sarà tale che $c(x_1x_2x_3x_4) = x_1x_2x_3x_4x_5x_6x_7$, dove $x_5x_6x_7$ sono calcolati a partire da $x_1x_2x_3x_4$ usando le equazioni della (2). Sotto l'ipotesi che sia occorso *al più un solo errore durante la trasmissione*, si vede immediatamente che ogni diverso errore nella trasmissione dei bit $x_1x_2x_3x_4x_5x_6x_7$ farebbe fallire un diverso sottoinsieme delle equazioni sopra riportate. Ad esempio, un errore su x_1 farebbe fallire la prima e la seconda equazione, mentre un errore su x_2 farebbe fallire la prima e la terza equazione, e così via.... Pertanto, verificando quali equazioni falliscono, si riesce a risalire *univocamente* alla sequenza trasmessa.

Le codifiche che studieremo, dette codifiche di Reed-Solomon, dal nome degli scopritori, sono definite come la valutazione di polinomi di grado limitato su di un campo finito. Sia $\mathbb{F}$ un campo finito dato (può essere utile immaginare che esso sia il campo F_{2^m}, ma non è strettamente necessario). I messaggi sorgente che vogliamo codificare, ad esempio sequenze binarie $\mathbf{a} \in \{0,1\}^\ell$, li possiamo interpretare come coefficienti di un polinomio di un dato grado fissato $k-1$. Ad esempio, possiamo scrivere $\mathbf{a} = \mathbf{a}_0\mathbf{a}_1 \ldots, \mathbf{a}_{k-1}$, dove ogni sottosequenza $\mathbf{a}_i \in \{0,1\}^m$, $i = 0, \ldots, k-1$, ed interpretare ciascuna $\mathbf{a}_i$ come un elemento del campo F_{2^m} prima visto. In questo modo, ad ogni sequenza $\mathbf{a} = \mathbf{a}_0\mathbf{a}_1 \ldots, \mathbf{a}_{k-1} \in \{0,1\}^\ell$ possiamo unicamente associare un polinomio $f(x) = f_0 + f_1x + f_2x^2 + \ldots + f_{k-1}x^{k-1}$ di grado al più $k-1$, dove $f_0 = \mathbf{a}_0, f_1 = \mathbf{a}_1, \ldots, f_{k-1} = \mathbf{a}_{k-1}$. In questo modo, le codifiche di Reed-Solomon sono ottenuti specificando il campo $\mathbb{F}$, l'intero k, ed $n < |\mathbb{F}|$ punti $\alpha_1, \ldots, \alpha_n \in \mathbb{F}$, e le sue parole sono

$$\mathcal{C} = \{(f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_n)) \ : \ f(x) = \sum_{i=0}^{k-1} f_i x^i, \ f_0, \ldots, f_{k-1} \in \mathbb{F}\}.$$

Denotiamo questa famiglia di codici, in generale, come $\mathrm{RS}_{\mathbb{F}}(n,k)$. Un caso particolare molto importante si ottiene quando $n = |\mathbb{F}|$ e valutiamo quindi i polinomi $f(x)$ su tutti gli elementi del campo $\mathbb{F}$. Possiamo notare una similarità con lo schema di condivisione di chiavi segrete di Shamir. In tal caso, la chiave segreta veniva assegnata al coefficiente di grado zero del polinomio $f(x)$. Qui, invece, la sequenza sorgente $(f_0f_1 \ldots f_{k-1})$ viene innanzitutto codificata nel polinomio $f(x) = f_0 + f_1x + \ldots + f_{k-1}x^{k-1}$, indi codificata nella sequenza $\mathbf{c} = (f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_n))$ di tutte le valutazioni $f(x)$. Di conseguenza, ogni componente/simbolo della parola codice $\mathbf{c}$ è un elemento del campo F_{2^m} e quindi, dal punto di vista pratico, è da considerarsi come una sequenza binaria lunga m.

Nel caso *non* binario, ovvero nel caso in cui il campo in questione è $\mathbb{F} = F_{p^m}$, per dati interi p (primo) e $m \geq 1$, ogni componente/simbolo di una parola codice di $\mathrm{RS}_{\mathbb{F}}(n,k)$ è un elemento del campo F_{p^m} e quindi deve essere considerato esso stesso come una sequenza lunga m sull'alfabeto $\{0, 1, \ldots, p-1\}$.

In virtù di risultati noti (che non vedremo), si sà che il codice $\mathrm{RS}_{\mathbb{F}}(n,k)$ permette, in linea di principio, la correzione di e errori, dove $e \leq \lfloor (n-k)/2 \rfloor$. Vediamo ora come progettare un algoritmo efficiente per la correzione di eventuali $e \leq \lfloor (n-k)/2 \rfloor$ errori occorsi. Sia $(f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_n))$ il vettore trasmesso, dove $f(x) = \sum_{i=0}^{r} f_i x^i$ un polinomio (a noi incognito) di grado al più $k-1$. Riceviamo un vettore $(y_1, \ldots, y_n)$ di valori in $\mathbb{F}$ tali che $y_i \neq f(\alpha_i)$, per al più e indici i (dove sono occorsi gli errori). Il problema consiste nel risalire a $(f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_n))$, conoscendo *solo* $(y_1, \ldots, y_n)$.

Se sapessimo quali sono i valori y_i errati, potremmo ignorarli ed attraverso l'interpolazione solo sui valori corretti ricostruire il polinomio $f(x)$ (e quindi il relativo vettore trasmesso $(f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_n))$). Purtroppo, noi *non* sappiamo chi tra gli y_i sono i valori corretti (ovvero per cui vale $y_i = f(\alpha_i)$) e chi sono i valori errati (ovvero per cui vale $y_i \neq f(\alpha_i)$).

Definiamo il polinomio $E(x)$ nel seguente modo:

$$E(x) = \prod_{i : y_i \neq f(\alpha_i)} (x - \alpha_i). \tag{3}$$

Ovviamente, non essendo noti i valori y_i errati, cioè quelli diversi dai $f(\alpha_i)$, non è noto nenanche il polinomio $E(x)$. Ciononostante, possiamo utilizzarlo nell'analisi. Chiaramente, dalla definizione, esso soddisfa le seguenti

proprietà (ovvero, equazioni che i coefficienti di E e f devono soddisfare):

$$E(\alpha_i)(f(\alpha_i) - y_i) = 0 \qquad \forall\ 1 \leq i \leq n. \tag{4}$$

Infatti, se $y_i = f(\alpha_i)$ si annulla il termine $(f(\alpha_i) - y_i)$, se invece $y_i \neq f(\alpha_i))$ si annulla il termine $E(\alpha_i)$, dalla definizione di $E(x)$.

Poniamo

$$N(x) = E(x)f(x). \tag{5}$$

Notiamo che grado$(E(x)) = e$ e grado$(N(x))$ = grado(E) + grado$(f) \leq e + k - 1$. Sulla base della (4), abbiamo di fatto il seguente risultato, che formalizziamo esplicitamente per future referenze.

Lemma 1 *Esiste un polinomio $E(x)$, di grado e, ed esiste un polinomio $N(x)$, di grado $\leq e - k + 1$, per cui vale la seguente relazione*

$$N(\alpha_i) - y_i E(\alpha_i) = 0, \tag{6}$$

per ogni $1 \leq i \leq n$.

L'idea che sfrutteremo è che possiamo trovare una coppia di polinomi siffatti risolvendo un opportuno sistema di equazioni.

Lemma 2 *Possiamo trovare un polinomio $E'(x)$, di grado pari al più ad e, ed un polinomio $N'(x)$, di grado pari al più a $e + k - 1$, non entrambi zero, che soddisfano la proprietà:*

$$N'(\alpha_i) - y_i E'(\alpha_i) = 0, \tag{7}$$

per ogni $1 \leq i \leq n$.

Dimostrazione. Poniamo

$$E'(x) = \sum_{j=0}^{e} a_j x^j, \qquad N'(x) = \sum_{j=0}^{e+k-1} b_j x^j,$$

dove a_j e b_j sono coefficienti *incogniti*. A causa della (7), tali coefficienti incogniti ovviamente soddisfano il seguente sistema di n equazioni lineari:

$$\sum_{j=0}^{e+k-1} b_j (\alpha_i)^j - y_i \sum_{j=0}^{e} a_j (\alpha_i)^j = 0 \qquad \forall\ 1 \leq i \leq n.$$

Sappiamo che un tale sistema ammette *almeno* una soluzione (ciò dal Lemma 1), che ovviamente possiamo trovare mediante i metodi dell'algebra lineare. □

Ovviamente, non abbiamo alcuna garanzia che i polinomi $E'(x)$ e $N'(x)$ che troveremo saranno uguali ai polinomi $E(x)$ (definito nella (3)) ed al polinomio $N(x)$ (definito nella (5)), cui siamo interessati. Tuttavia, il seguente risultato ci dice che possiamo recuperare il polinomio f da *ogni* coppia $E'(x)$ ed $N'(x)$ che possiamo aver trovato (che è poi quello che ci interessa, in quanto una volta noto $f(x)$ possiamo calcolarci il vettore trasmesso $(f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_n))$).

Lemma 3 *Siano $E'(x)$ ed $N'(x)$* arbitrari *polinomi che soddisfano la (7). Sotto l'ipotesi che il numero di errori occorso e sia tale che $e \leq (n - k)/2$, si ha che*

$$f(x) = \frac{N'(x)}{E'(x)}. \tag{8}$$

Dimostrazione. Consideriamo il polinomio $R(x) = N'(x) - E'(x)f(x)$. Notiamo che per ogni i tale che $f(\alpha_i) = y_i$ (ovvero per tutti le componenti i-esime della parola codice su cui *non* è occorso alcun errore, ed esse sono in numero di $n - e$), abbiamo che

$$R(\alpha_i) = N'(\alpha_i) - E'(\alpha_i)f(\alpha_i) = N'(\alpha_i) - E'(\alpha_i)y_i = 0.$$

Ciò dal Lemma 3. Ne segue che il polinomio $R(x)$ ha almeno $n - e$ zeri. D'altra parte, il grado di $R(x)$ è al più pari a

$$\max(\text{grado}(N'), \text{grado}(E') + \text{grado}(f)) \leq e + k - 1.$$

Sotto la nostra ipotesi che $e \leq (n - k)/2$ abbiamo che $n - e > e + k - 1$. Ovvero, il polinomio $R(x)$ ha più zeri del suo grado, quindi $R(x)$ deve essere necessariamente il polinomio zero, e quindi $N'(x) = E'(x)f(x)$. □

In conclusione, potendo dedurre il polinomio $f(x)$ dai polinomi $N'(x)$ e $E'(x)$, attravero la (8), possiamo calcolarci la parola codice originale $(f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_n))$.

É interessante notare che i Codici di Reed-Solomon sono attualmente usati in numerose situazioni pratiche.

Ad esempio, sono usati per memorizzare dati (musica, film, ...) su CD e DVD in maniera tale che tali dati siano recuperabili anche in presenza di malfunzionamenti del supporto (ad esempio, graffi sul CD o DVD, si veda qui https://www.researchgate.net/publication/265634360_Reed-Solomon_codes_and_the_compact_disc), per rappresentare dati via QR (come quello in figura, che codifica le parole "Teoria dell'Informazione") in maniera tale che la lettura del QR possa avvenire correttamente anche se il codice è "sporco, o spiegazzato" (ovvero se nella fase di lettura del QR accadono errori), per la trasmissione dati nello spazio (ed es., i satelliti Voyager I e II, lanciati nel 1977 ed attualmente usciti dal sistema solare, usano codici di Reed-Solomon per la trasmissione verso la Terra), ed in molti altri ambiti[1] Ulteriori esempi di applicazioni nella vita reale qui: https://en.wikipedia.org/wiki/Reed-Solomon_error_correction

[1]La prossima volta che qualcuno vi chiede a cosa serve la Matematica, fategli questi esempi per illustrare che senza di essa non vi sarebbe la possibilità di comunicare!