L'Aritmetica del Calcolatore

Luca Gemignani
luca.gemignani@unipi.it

23 febbraio 2018

Indice

Lezione 1: Rappresentazione in Base e Numeri di Macchina.

Sia $B \in \mathbb{N}$, $B > 1$. Il seguente teorema caratterizza la rappresentazione di un numero $x \in \mathbb{R}$, $x \neq 0$, in base B.

Teorema 1.1. Dato $x \in \mathbb{R}$, $x \neq 0$, esistono e sono univocamente determinati

1. un intero $p \in \mathbb{Z}$ detto *esponente* della rappresentazione ;
2. una successione di numeri naturali $\{d_i\}_{i\geq 1}$, $d_1 \neq 0$, $0 \leq d_i \leq B-1$, non definitivamente uguali a $B-1$ dette *cifre* della rappresentazione;

tali per cui si ha

$$x = \mathrm{sign}(x) B^p \sum_{i=1}^{+\infty} d_i B^{-i}. \tag{1}$$

La rappresentazione (1) di un numero reale x di dice rappresentazione *normalizzata* in *virgola mobile* (*floating point*) in quanto l'esponente p è determinato in modo da avere parte intera nulla e prima cifra dopo la virgola non nulla.

Si osserva che

- Le condizioni $d_1 \neq 0$ (rappresentazione normalizzata) e d_j non definitivamente uguale a $B-1$ sono introdotte per garantire l'unicità della rappresentazione. A titolo di esempio in base $B = 10$ si ha

$$1 = +10^1(1 \cdot 10^{-1}) = +10^2(0 \cdot 10^{-1} + 1 \cdot 10^{-2})$$

ma la seconda non risulta accettabile perchè ha la prima cifra nulla. Analogamente in base $B = 10$ si ha

$$1 = +10^1(1 \cdot 10^{-1}) = 0.\bar{9} = +10^0 \sum_{i=1}^{+\infty} 9 \cdot 10^{-i}$$

ma la seconda non risulta accettabile perchè ha le cifre tutte e quindi definitivamente uguali a $9 = 10 - 1$.

- Il numero $x = 0$ non ammette rappresentazione normalizzata. In macchina verrà trattato e memorizzato in modo speciale.
- La rappresentazione floating point dei numeri reali si estende all'insieme dei numeri complessi $z = a + \mathrm{i}b$ rappresentati come coppie di numeri reali.

Poichè i registri delle unità aritmetiche ed i dispositivi di memoria di un calcolatore consentono la memorizzazione di un numero finito di cifre binarie *l'insieme dei numeri di macchina*, cioè l'insieme dei numeri reali esattamente rappresentabili in macchina ha cardinalità finita. Risulta pertano essenziale determinare criteri e condizioni che consentano la rappresentazione quanto più accurata possibile dei numeri reali nel calcolatore.

Dal teorema di rappresentazione in base segue che la rappresentazione di un numero reale nel calcolatore può avvenire assegnando delle posizioni di memoria per il segno, per l'esponente e per le cifre della rappresentazione. In tal modo la rappresentazione di un numero in macchina assume la seguente struttura

Segno	Esponente	Cifre della rappresentazione

Esempio 1.1. I personal computer che implementano lo standard IEEE 754-1985 prevedono la memorizzazione su registri lunghi 32 bit –$1 + 8 + 23$– per la *singola precisione* e 64 bit –$1 + 11 + 52$– per la *doppia precisione*.

Definizione 1.1. Si definisce *insieme dei numeri di macchina* con t cifre, base B e range $(-m, M)$ l'insieme dei numeri reali

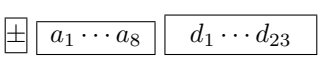
$$\mathbb{F}(B, t, m, M) =$$
$$\{0\} \cup \{x \in \mathbb{R} \colon x = \mathrm{sign}(x) B^P \textstyle\sum_{i=1}^{t} d_i B^{-i}, 0 \le d_i \le B - 1, d_1 \neq 0, -m \le p \le M\}.$$

Si osserva che

- L'insieme dei numeri di macchina $\mathbb{F}(B, t, m, M)$ ha cardinalità finita $N = 2B^{t-1}(B - 1)(M + m + 1) + 1$;
- Se $x \in \mathbb{F}(B, t, m, M)$ e $x \neq 0$ allora $\omega = B^{-m-1} \le |x| \le B^M(1 - B^{-t}) = \Omega$. Ne segue che non è possibile rappresentare esattamente numeri non nulli di modulo minore ad ω. Per aggirare questa limitazione lo standard IEEE754 prevede anche una rappresentazione *denormalizzata*. Quando $p = -m$ la condizione $d_1 \neq 0$ può essere abbandonata e quindi vengono rappresentati numeri positivi e negativi compresi in modulo tra B^{-m-t} e $B^{-m}(B^{-1} - B^{-t})$. Analogamente se $p = M$ si introducono rappresentazioni speciali per i simboli $\pm\infty$ e NaN *not a number*.

- L'insieme dei numeri di macchina $\mathbb{F}(B,t,m,M)$ è simmetrico rispetto all'origine. Posto $x=(-1)^s B^p\alpha \in \mathbb{F}(B,t,m,M)$ allora il successivo numero di macchina risulta essere $y=(-1)^s B^p(\alpha+B^{-t})$ per cui $|y-x|=B^{p-t}$ variabile con p ovvero con l'ordine di grandezza dei numeri considerati.

Esempio 1.2. Si consideri una rappresentazione su 32 bit in base 2 del tipo



Per l'esponente abbiamo 256 possibili valori che si riducono a 254 eliminando le configurazioni con tutti zero e tutti 1. Mediante una tecnica di traslazione questi 254 valori sono utilizzati per rapprentare gli interi nell'intervallo $[-126,127]$. Se $a_1=\ldots=a_8=0$ allora il numero rappresentato è $x=\pm(0.d_1d_2\ldots d_{23})_2\cdot 2^{-126}$. Se $a_1=\ldots=a_8=1$ allora $x=\pm\infty$ se $d_1=\ldots=d_{23}=0$ e $x=$NaN altrimenti. Per valori dell'esponente compresi tra 1 e 254 il numero rappresentato è $x=\pm(1.d_1d_2\ldots d_{23})_2\cdot 2^{p-127}$ (in notazione normalizzata la prima cifra non deve essere rappresentata). Il più piccolo numero positivo normalizzato è $\omega=(1.00\ldots0)_2\cdot 2^{-126}=2^{-126}$. Il più grande numero normalizzato è $\Omega=(1.11\ldots1)\cdot 2^{127}=2^{128}(1-2^{-24})$. Per il range si trova $m=125$ e $M=128$. Il numero di cifre è $t=24$. Il primo numero di macchina più grande di $1=(1.00\ldots0)_2\cdot 2^0$ è $1+2^{-23}$.

Lezione 2: Aritmetica di Macchina.

Rappresentare un numero reale non nullo $x\in\mathbb{R}$, $x\neq 0$, in macchina significa *approssimare* x con un numero $\tilde{x}\in\mathbb{F}(B,t,m,M)$ commettendo un errore relativo di rappresentazione

$$\epsilon_x=\frac{\tilde{x}-x}{x},\quad x\neq 0,$$

quanto più piccolo possibile in valore assoluto. La preferenza per l'errore relativo anzichè l'errore assoluto $\eta_x=\tilde{x}-x$ dipende dalla distribuzione dei numeri nel sistema floating point che garantisce per il primo ma non per il secondo l'indipendenza della stima rispetto alla grandezza del numero x.

Dato $x\in\mathbb{R}$, $x\neq 0$, distinguiamo 2 casi

1. $|x|<\omega$ (*underflow*) o $|x|>\Omega$ (*overflow*);
2. $\omega\leq|x|\leq\Omega$.

Nel secondo caso le tecniche di approssimazione previste dallo standard IEEE 754 sono:

1. *round to the nearest* (arrotondamento): il numero x viene approssimato con il numero rappresentabile $\tilde{x}$ più vicino;

2. *round toward zero* (troncamento): il numero x viene approssimato con il più grande numero rappresentabile $\tilde{x}$ il cui valore assoluto risulti minore od uguale al valore assoluto di x;

3. *round toward plus infinity*: il numero x viene approssimato al più piccolo numero rappresentabile maggiore del dato;

4. *round toward minus infinity*: il numero x viene approssimato al più grande numero rappresentabile minore del dato.

Assumiamo per semplicità di considerare una macchina che opera con troncamento sull'insieme $\mathbb{F}(B, t, m, M)$. Per convenzione indichiamo con $\mathrm{trn}(x) = \tilde{x}$ il risultato dell'approssimazione di x con troncamento e più generalmente $\mathrm{fl}(x)$ l'approssimazione in macchina del dato x nel sistema floating point considerato. Il primo risultato fornisce una maggiorazione uniforme dell'errore di rappresentazione.

Teorema 2.1. Sia $x \in \mathbb{R}$ con $\omega \leq |x| \leq \Omega$. Si ha

$$|\epsilon_x| = |\frac{\mathrm{trn}(x) - x}{x}| \leq u = B^{1-t}.$$

Dimostrazione. Sia $x = (-1)^s B^p \alpha$. L'errore assoluto $|\,\mathrm{trn}(x) - x|$ risulta maggiorato dalla distanza tra due numeri di macchina consecutivi e quindi

$$|\frac{\mathrm{trn}(x) - x}{x}| \leq B^{p-t}.$$

Inoltre $|x| \geq B^{p-1}$. Pertanto vale

$$|\epsilon_x| = |\frac{\mathrm{trn}(x) - x}{x}| \leq \frac{B^{p-t}}{B^{p-1}} = B^{1-t} = u.$$

□

Si osservi che:

- La quantità u detta *precisione di macchina* è indipendente dalla grandezza del numero e caratteristica dell'aritmetica floating point (insieme dei numeri rappresentabili + tecnica di approssimazione) implementata sulla macchina su cui stiamo operando. Se ad esempio operiamo con arrotondamento la distanza tra x e la sua approssimazione di macchina e quindi la precisione di macchina si dimezzano.

- Poichè $1 = B^1 \cdot B^{-1}$, per determinare la precisione di macchina possiamo determinare il piu‘ piccolo numero di macchina maggiore di 1. Il seguente script Matlab fornisce il valore richiesto

```
eps=0.5;
eps1=eps+1;
```

```
while(eps1>1)
eps=0.5*eps;
eps1=eps+1;
end
eps=2*eps
```

- Dal teorema precedente si ricava che dato $x \in \mathbb{R}$ in assenza di situazioni di overflow ed underflow per la sua rappresentazione in macchina vale $\mathrm{fl}(x) = x(1+\epsilon_x)$ con $|\epsilon_x| \leq u$ ed u la precisione di macchina relativa. Questa relazione esprime il modo in cuio viene generalmente descritto il legame tra un numero reale e la sua rappresentazione in macchina.

Per le operazioni aritmetiche eseguite in un sistema floating point si pone un analogo problema di approssimazione in quanto il risultato dell'operazione eseguita tra due numeri di macchina in generale non sarà un numero di macchina. Indichiamo con $\oplus, \ominus, \otimes, \oslash$ le operazioni aritmetiche di macchina corrispondenti rispettivamente all'addizione, sottrazione, prodotto e divisione. Richiediamo che le operazioni di macchina siano interne all'insieme dei numeri di macchina e forniscano ovviamente un'approssimazione quanto più accurata possibile del risultato esatto. Una ragionevole definizione risulta pertanto la seguente :

$$\forall a, b \in \mathbb{F}(B, t, m, M), \quad a \oplus b = \mathrm{fl}(a+b),$$

e similmente per le altre operazioni. In tal modo in assenza di situazioni di overflow ed underflow dal teorema precedente segue che

$$\forall a, b \in \mathbb{F}(B, t, m, M), \quad a \oplus b = \mathrm{fl}(a+b) = (a+b)(1+\epsilon_1), \quad |\epsilon_1| \leq u,$$

con ϵ_1 detto *errore locale dell'operazione.*

Se $a, b \in \mathbb{R}$ in assenza di situazioni di overflow ed underflow si ha

$$\mathrm{fl}(a+b) = \mathrm{fl}(a) \oplus \mathrm{fl}(b) = (a(1+\epsilon_a)+b(1+\epsilon_b))(1+\epsilon_1) \doteq (a+b)+a\epsilon_a+b\epsilon_b+(a+b)\epsilon_1,$$

dove con il simbolo $\doteq$ si intende che l'uguaglianza vale considerando le sole componenti lineari negli errori e trascurando le componenti di ordini superiore al primo *analisi al primo ordine.* Si ottiene quindi che se $a, b \in \mathbb{R}$, $a+b \neq 0$, in assenza di situazioni di overflow ed underflow vale

$$\frac{\mathrm{fl}(a+b)-(a+b)}{a+b} \doteq \frac{a}{a+b}\epsilon_a + \frac{b}{a+b}\epsilon_b + \epsilon_1,$$

che esprime la dipendenza dell'errore totale commesso nel calcolo della somma tra due numeri reali rispetto agli errori generati dall'approssimazione dei dati iniziali *errore inerente* e agli errori generati dall'algoritmo di calcolo *errore algoritmico* visto come sequenza di operazioni aritmetiche.

Nella prossima lezione tratteremo più in dettaglio questa dipendenza mostrando che la decomposizione è generale e si applica al calcolo di una generica funzione razionale.