Strutture dati per i numeri reali

Nello scrivere dei numeri usiamo, a seconda del contesto, diverse rappresentazioni, come ad esempio

$$\frac{1}{2} \qquad \frac{1}{3} \qquad 0.25 \qquad 3.14159 \qquad 2^{-3}$$

Nella descrizione dei numeri nella memoria di una macchina è invece utile usare una rappresentazione unica. Ciò infatti permette di semplificare sia le strutture dati che gli algoritmi.

Inoltre ci comportiamo di solito come se non avessimo alcuna limitazione nelle possibiltà di rappresentare, usando la base 10, qualsiasi numero reale.

È evidente però che non si può avere a disposizione lo spazio necessario a rappresentare un qualsiasi numero intero o razionale, nè su un foglio di carta nè nella memoria di una macchina.

Inoltre le usuali rappresentazioni di numeri irrazionali come

$$\pi \qquad \sqrt{2} \qquad sin\frac{\pi}{4}$$

pongono il problema di rappresentare in generale, oltre tali numeri, le espressioni algebriche che li contengono.

È per queste ragioni che normalmente si fa uso di

– una rappresentazione dei numeri interi costituita da una stringa di un fissato numero di cifre (in una *base* b)

$$n_t n_{t-1} n_{t-2} \ldots n_1 n_0$$

  a cui corrisponde il numero

$$N = n_t b^t + n_{t-1} b^{t-1} + n_{t-2} b^{t-2} + \ldots + n_1 b^1 + n_0 b^0$$

– una rappresentazione dei numeri reali costituita da una coppia di numeri m, f (*mantissa, esponente*) — ciascuno rappresentato da una stringa di un fissato numero di cifre — a cui corrisponde il numero

$$x = m b^f$$

  Per utilizzare nel modo migliore le cifre riservate alla *mantissa* si assume che essa sia *normalizzata* cioè tale che

$$b^{-1} \leq |m| < 1$$

  A questa rappresentazione dei numeri reali ci si riferisce con il termine *floating point*.

Le rappresentazioni disponibili dipendono sia dal compilatore che dalla macchina.

Linguaggi come il C, il Fortran, il Pascal dispongono almeno dei seguenti tipi di dati per i numeri:

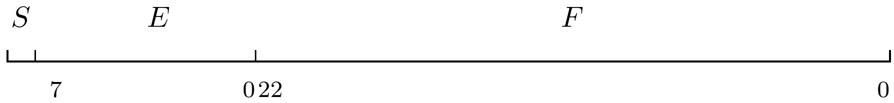
tipi per i numeri interi		*tipi per i numeri reali in generale*	
C:	short, int, long	C:	float, double
Fortran:	integer	Fortran:	real, double precision
Pascal:	integer, *subrange*	Pascal:	real

Per ogni macchina esistono delle definizioni (al livello hardware e al livello del software di base) di strutture dati per i numeri.

Il particolare compilatore usato fa corrispondere ciascuno tipo ad una delle strutture dati disponibili a basso livello, oppure definisce proprie strutture dati.

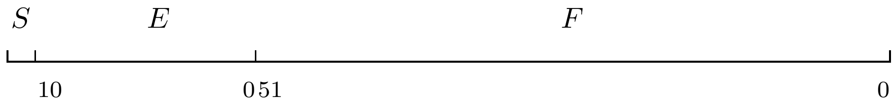
Le macchine con il processore aritmetico Intel 8087 (80287, 80387), oppure Motorola 68881, dispongono di strutture dati per gli interi costituite da 1, 2, 4, 8 byte, mentre le strutture dati *floating point*[1] sono le seguenti

single



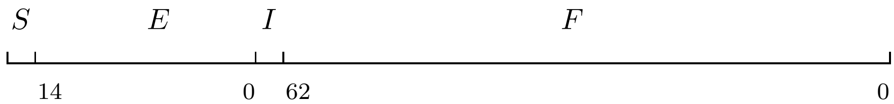
$$\begin{array}{lll} 0 < E < 255 & & x = (-1)^S 2^{(E-127)}(1 + F2^{-23}) \\ E = 0 \quad , & F \neq 0 & x = (-1)^S 2^{(-126)} F2^{-23} \\ E = 0 \quad , & F = 0 & x = 0 \end{array}$$

double



$$\begin{array}{lll} 0 < E < 2047 & & x = (-1)^S 2^{(E-1023)}(1 + F2^{-52}) \\ E = 0 \quad , & F \neq 0 & x = (-1)^S 2^{(-1022)} F2^{-52} \\ E = 0 \quad , & F = 0 & x = 0 \end{array}$$

extended



$$\begin{array}{lll} 0 < E < 32767 & & x = (-1)^S 2^{(E-16383)}(I + F2^{-63}) \\ I = 0 \quad , & F = 0 & x = 0 \end{array}$$

Il numero di byte assegnati a ciascuna struttura dati è rispettivamente 4, 8, 10. Il numero di cifre della *mantissa* è rispettivamente 24, 53, 64 (notare che nei primi due casi la mantissa ha una cifra in più di F).

I linguaggi di Computer Algebra (Reduce, Macsyma, Scratchpad, muMath, Maple, Mathematica) dispongono di strutture dati dinamiche per interi e razionali — rappresentati come coppie di interi. Tali strutture sono delle liste in cui vengono inseriti tanti elementi quanti risultano necessari durante le valutazioni numeriche. Questo permette di rappresentare *qualsiasi* numero razionale, avendo come unica limitazione la memoria disponibile.

[1] Secondo lo *IEEE standard 754.*