RAPPRESENTAZIONE DEI NUMERI IN UN CALCOLATORE

Un calcolatore è in grado di rappresentare solo un numero finito di cifre

$\Downarrow$

approssimazione dei numeri reali

$\Downarrow$

risultati delle operazioni non esattamente rappresentabili

$\Downarrow$

creazione e propagazione di errori in una successione di operazioni (algoritmo)

- **Numeri di macchina o numeri finiti**
- **Operazioni sui numeri finiti**
- **Propagazione degli errori: errori inerenti ed errori di arrotondamento**
- **Malcondizionamento di un problema e stabilità di un algoritmo.**

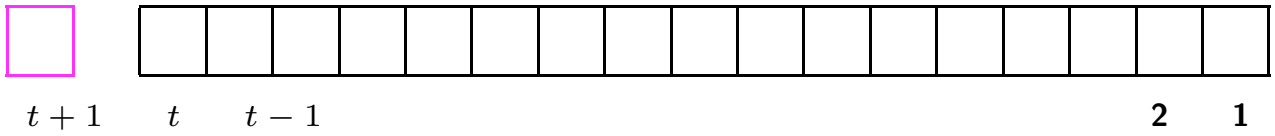
Numeri finiti o numeri di macchina

A causa della limitata lunghezza della parola di memoria, sono rappresentabili effettivamente su calcolatore:

- **un intervallo limitato di interi (numeri fixed point o a punto fisso);**
- **un insieme finito di numeri razionali (numeri floating point o a virgola mobile).**

Numeri fixed point (β, t)

β base di rappresentazione
t cifre a disposizione per la rappresentazione del valore assoluto del numero
$t+1$ cifre a disposizione per la rappresentazione del numero (con segno).



N un intero non negativo

$$N = (d_p d_{p-1} ... d_1)_\beta$$

$fi(N)$

$t \geq p$

0	0	...	0	d_p	d_{p-1}	...	d_1

$fi(N) = N$

$t < p$

0	d_t	d_{t-1}	...	...	...	d_2	d_1

$fi(N) \neq N$

OVERFLOW INTERO

Osservazione

Se $t < p$ allora $fi(N)$ è il resto della divisione di N per β^t.

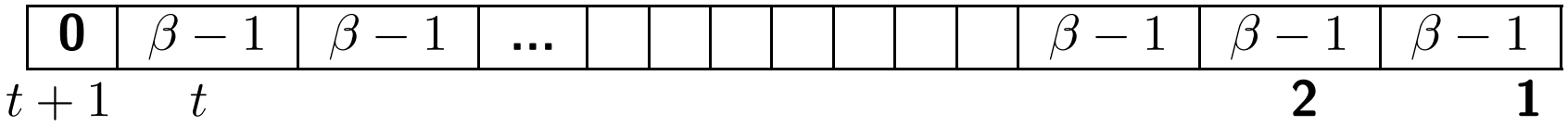
ESEMPIO: $\beta = 10 \quad t = 3 \quad N = 12436$

$$fi(N) = \boxed{0}\boxed{4}\boxed{3}\boxed{6}$$

Infatti

$$\begin{aligned} N &= 1 \times 10^4 + 2 \times 10^3 + 4 \times 10^2 + 3 \times 10^1 + 6 \times 10^0 \\ &= 12 \times 10^3 + 436 \end{aligned}$$

Il più grande intero rappresentabile esattamente



$$\bar{N} = (\beta - 1)\beta^{t-1} + ... + (\beta - 1)\beta^0 = \beta^t - 1$$

Sono rappresentato esattamente solo gli interi non negativi dell'intervallo

$$[0, \beta^t - 1]$$

$$\begin{aligned}\sum_{i=0}^{n} x^i &= x^n + x^{n-1} + ... + x^2 + x + 1 \\ &= \frac{x^{n+1}-1}{x-1}\end{aligned}$$

Infatti si verifica che

$$\begin{aligned}(x^n + x^{n-1} + ... + x^2 + x + 1)(x - 1) &= x^{n+1} + x^n + ... + x^2 + x + \\ &\quad -x^n - x^{n-1} - ... - x^2 - x \\ &= x^{n+1} - 1\end{aligned}$$

N intero negativo

Consideriamo per semplicità il caso particolare $\beta = 2$. rappresentazione complemento a 2 in $t+1$ cifre:

$$fi(N) = (2^{t+1} - |N|)_{t+1}$$

Una regola pratica per ottenere $fi(N)$è di

1. **prendere $|N|$;**
2. **scambiare 0 con 1 e 1 con 0**
3. **aggiungere 1.**

La cifra $t+1$-esima (o il primo bit) è uguale a 1.

Il più piccolo intero negativo esattamente rappresentabile

$$\mathbf{1000...00} = fi(-\mathbf{1000...00}) = -\mathbf{2}^{t}$$

Infatti

$$\begin{array}{r} \mathbf{1000...00} \\ \Downarrow \\ \mathbf{0111...11} \quad + \\ \mathbf{1} \quad = \\ \hline \mathbf{1000...00} \end{array}$$

Intervallo degli interi esattamente rappresentabili $[-\beta^t, \beta^t - 1]$. Al di fuori dell'intervallo si incorre nell'overflow intero.

Esempi

Interi a 32 bit $\beta = 2, t + 1 = 32$**:**
sono rappresentabili esattamente gli interi in $[-2^{31}, 2^{31} - 1]$**.**
Interi a 16 bit $\beta = 2, t + 1 = 16$**;**
sono rappresentabili gli interi in $[-32768, 32767]$**.**

ESEMPI

$$
\begin{aligned}
fi((1235)_{10}) &= fi((1001101001)_2) = 0000001001101001 \\
fi(-(1235)_{10}) &= fi(-(1001101001)_2) = 1111110110010111 \\
fi(-1) &= 1111111111111111
\end{aligned}
$$

Insieme dei numeri fixed point interi rappresentabili con $\beta = 2, t + 1 = 4$

$fi(N)$	N
0111	**7**
0110	**6**
0101	**5**
0100	**4**
0011	**3**
0010	**2**
0001	**1**
0000	**0**
1111	**-1**
1110	**-2**
1101	**-3**
1100	**-4**
1011	**-5**
1010	**-6**
1001	**-7**
1000	**-8**

ESERCIZIO

Sia $\beta = 2$ e $t + 1 = 16$. Rappresentare in fixed point i numeri 1023 e -31128.

$$\begin{aligned} fi(1023) &= fi((1111111111)_2) = 0000001111111111 \\ fi(-(31128)_{10}) &= fi(-111100110011000)_2) \\ &= 000011001100111 + 1 \\ &= 1000011001101000 \end{aligned}$$

Rappresentare in base 10 il seguente numero fixed point ($\beta = 2$ e $t + 1 = 16$).

$$\begin{array}{rl} & 1000111000000010 \\ 1000111000000010 & - \\ \hline 1000111000000001 & 1 = \\ \Downarrow & \\ 0111000111111110 & = 29128 \Longrightarrow N = -29128 \end{array}$$

Aritmetica dei numeri fixed point

Supponiamo di lavorare con interi positivi N tali che $fi(N) = N$.

SOMMA

$$\boldsymbol{fi(N_1 + N_2) = fi(fi(N_1) + fi(N_2)) = (N_1 + N_2)_{t+1}}.$$

ESEMPI. $\beta = 2, t = 5$.

- $\boldsymbol{N_1 = 010010(= 18_{10}),\ N_2 = 000101(= 5_{10})}$

  $$\begin{array}{r} \mathbf{010010} \\ \mathbf{000101} \\ \hline \mathbf{010111} \end{array} \Longrightarrow \boldsymbol{fi(N_1 + N_2) = 010111(= 23_{10})}.$$

- **OVERFLOW**

  $\boldsymbol{N_1 = 010011(= 19_{10}),\ N_2 = 001110(= 14_{10})}$

  $$\begin{array}{r} \mathbf{010011} \\ \mathbf{001110} \\ \hline \mathbf{100001} \end{array} \Longrightarrow \boldsymbol{fi(N_1 + N_2) = 100001 = fi(-31_{10})}$$ **invece di** $\mathbf{33_{10}}$

DIFFERENZA

$fi(N_1 - N_2) = (fi(N_1) + fi(-N_2))_{t+1}$

In pratica:

$fi(N_1 - N_2) = (N_1 + 2^{t+1} - N_2)_{t+1} = (2^{t+1} + (N_1 - N_2))_{t+1}$

1. **Caso $N_1 \geq N_2$.**
   con $\beta = 2, t+1 = 6, N_1 = 001111 (= 15_{10}), N_2 = 000111 (= 7_{10})$,

$$\begin{array}{rl} N_1 & 001111 \\ -N_2 & 111001 \\ \hline & 1001000 \end{array} \rightarrow fi(8_{10})$$

2. **Caso $N_1 < N_2$.**
   Se $N_1 = 000111 (= 7_{10}), N_2 = 001111 (= 15_{10})$,

$$\begin{array}{rl} N_1 & 000111 \\ -N_2 & 110001 \\ \hline & 111000 \end{array} \rightarrow fi(-8_{10})$$

PRODOTTO

Riconducibile ad addizioni e scorrimenti delle cifre a destra. Poichè il prodotto di interi in $t+1$ cifre può produrre un risultato in $2(t+1)$ cifre, spesso si incorre in overflow intero. $\beta = 2, t+1 = 6$.

$N_1 = 000011 = 3_{10}, N_2 = 000101 = 5_{10}; fi(N_1 * N_2) = 15_{10}$.

				0	0	0	0	1	1	×	
				0	0	0	1	0	1	=	
				0	0	0	0	1	1		
			0	0	0	0	0	0			
		0	0	0	0	1	1				
	0	0	0	0	0	0					
0	0	0	0	0	0						
0	0	0	0	0	0	0	1	1	1	1	

$$\mathbf{R1} = \boxed{000011}$$

$$\mathbf{R} = \begin{array}{cc} \mathbf{RH} & \mathbf{RL} \\ \boxed{000000} & \boxed{000101} \end{array}$$

$$\mathbf{Passo\ 1} = \begin{cases} \mathbf{RH} \leftarrow \mathbf{RH} + \mathbf{RL}_0 \times \mathbf{R1} \\ \begin{array}{cc} \mathbf{RH} & \mathbf{RL} \\ \boxed{000011} & \boxed{000101} \end{array} \\ \mathbf{Shift\ a\ destra\ di\ R} \\ \begin{array}{cc} \mathbf{RH} & \mathbf{RL} \\ \boxed{000001} & \boxed{100010} \end{array} \end{cases}$$

$$\mathbf{Passo\ 2} = \begin{cases} \mathbf{RH} \leftarrow \mathbf{RH} + \mathbf{RL}_0 \times \mathbf{R1} \\ \begin{array}{cc} \mathbf{RH} & \mathbf{RL} \\ \boxed{000001} & \boxed{100010} \end{array} \\ \mathbf{Shift\ a\ destra\ di\ R} \\ \begin{array}{cc} \mathbf{RH} & \mathbf{RL} \\ \boxed{000000} & \boxed{110001} \end{array} \end{cases}$$

Passo 3 =

$RH \leftarrow RH + RL_0 \times R1$

RH	RL
000011	110001

Shift a destra di R

RH	RL
000001	111000

Passo 4 =

$RH \leftarrow RH + RL_0 \times R1$

RH	RL
000001	111000

Shift a destra di R

RH	RL
000000	111100

Passo 5 =

$RH \leftarrow RH + RL_0 \times R1$

RH	RL
000000	111100

Shift a destra di R

RH	RL
000000	011110

Passo 6 $=$

$\mathrm{RH} \leftarrow \mathrm{RH} + \mathrm{RL}_0 \times \mathrm{R1}$

RH	RL
000000	111100

Shift a destra di R

RH	RL
000000	001111

QUOZIENTE

Si usano due accumulatori A e B. In A si mette il divisore e nella parte R di B il dividendo. Si riconduce a sottrazioni e scorrimenti. Il primo passo è di eseguire s scorrimenti a destra di R fino a che $fi(N_2) \leq R < \beta fi(N_2)$. Si eseguono poi $s+1$ passi uguali (l'ultimo senza scorrimento verso sinistra).

$\beta = 2, t+1 = 6$.

$N_1 = 010000 = 16_{10}, N_2 = 000101 = 5_{10}; fi(N_1/N_2) = 3_{10}$**, con resto 1.**

	000101	**A**		
B	**010000**	**000000**	**s=1**	
B	**001000**	**000000**		
B	**000011**	**000001**	**q=1**	$\leftarrow R - fi(N_2)$
B	**000110**	**000010**	$\leftarrow$	
B	**000001**	**000011**	**q=1**	$R \leftarrow R - fi(N_2)$
	resto R	**quoziente**		

L'aritmetica tra numeri fixed point è esatta purchè si resti nell'intervallo rappresentabile.

NUMERI FLOATING POINT

L'insieme dei numeri reali è SIMULATO su un calcolatore mediante un insieme di numeri finiti F o numeri floating point;

$$\alpha = \pm m\beta^p$$

$$m = (.a_1 a_2 .... a_t a_{t+1} ...)_\beta$$

L' insieme F dipende da quattro parametri:

- β: **base di rappresentazione;**
- t: **numero di cifre per la rappresentazione della mantissa;**
- L: **valore del più piccolo esponente rappresentabile;**
- U: **valore del più grande esponente rappresentabile.**

$\Longrightarrow$ **F**(β, t, L, U)**.**

Convenzioni del formato IEEE

(Institute of Electrical and Eletronic Engineerings), documento 754 dell'ANSI: $\beta = 2$

$$\alpha = (-1)^s (1.a_2 a_3 ... a_t a_{t+1} ...) 2^p$$

- **segno**: s **su 1 bit** $\begin{cases} 0 \text{ se } \alpha > 0 \\ 1 \text{ se } \alpha < 0 \end{cases}$
- **esponente** p: **è un intero,** $L \leq p \leq U$**; si rappresenta per traslazione in** l **bit, ossia:**

  rappresentazione di $p = p + bias$
- **mantissa**: **vengono rappresentate i** t **bit più significativi, troncando; fisicamente poichè il primo bit è sempre 1, vengono rappresentati solo** $t - 1$ **bit** $(a_2 a_3 ... a_t)$**.**

	SEMPLICE PRECISIONE	DOPPIA PRECISIONE
N. totale di bit	**32 (4 byte)**	**64 (8 byte)**
segno s	**1 bit**	**1 bit**
t	**24**	**53**
l	**8**	**11**
bias	**127 (01111111)**	**1023 (01111111111)**
U	**127**	**1023**
L	**-126**	**-1022**

Esempio

Come si rappresenta in virgola mobile (pr. semplice) il numero 4.25?

(a) Convertire il numero da base 10 a base 2

$$(4.25)_{10} = (100.01)_2$$

(b) Normalizzare il numero

$$(100.01)_2 = 1.0001 \cdot 2^2$$

(c) Sommare il bias all'esponente

$$\begin{array}{rll} \text{esponente} & 00000010 & + \\ \text{bias} & 01111111 & = \\ \hline & 10000001 & \rightarrow (127+2)_{10} \end{array}$$

0	10000001	00010000000000000000000
segno	esponente	mantissa

Viceversa

Quale numero reale α rappresentato dalla seguente stringa?

1 10000000 10010010000000000000000

segno **esponente** **mantissa**

$$\text{esponente - bias} = \begin{array}{r} 10000000 \quad - \\ 01111111 \\ \hline 00000001 \quad \longrightarrow 1 \end{array}$$

$$\text{mantissa} = \frac{1}{2} + \frac{1}{16} + \frac{1}{128} = 0.5703125$$

$$\alpha = -1.5703125 \cdot 2^1 = -3.140625$$

Più piccolo numero rappresentabile in valore assoluto

$1.0000....\ 2^{-126} \simeq 10^{-38}$

0 00000001 00000000000000000000000

segno esponente mantissa

-126+127=1

Per numeri più piccoli si ha UNDERFLOW floating point $\Longrightarrow 0$.

Più grande numero rappresentabile in valore assoluto

$1.1111...1\ 2^{127} = (1 + 1 - 2^{-23})2^{127} \simeq 10^{38}$

0 11111110 11111111111111111111111

segno esponente mantissa

127+127=254

Per valori più grandi si ha OVERFLOW floating point $\Longrightarrow$ arresto dell'alaborazione.

Rappresentazione degli esponenti

$p = 0$	è rappresentato con 01111111.
$p = 1$	è rappresentato con 100000000.
$p = -1$	è rappresentato con 01111110.
$p = 127 = U$	è rappresentato con 11111110=$(254)_{10}$.
$p = -126 = L$	è rappresentato con 00000001=$(1)_{10}$.

Gli esponenti negativi o nulli hanno il bit più significativo uguale a 0, gli esponenti positivi uguale a 1.

RAPPRESENTAZIONI SPECIALI

mantissa	esponente		
0	0	rappresenta	$(-1)^s 0$
0	$11111111 = (255)_{10}$	rappresenta	$\pm\infty$
$\neq 0$	$11111111 = (255)_{10}$	rappresenta	NaN

NUMERI DENORMALIZZATI: sono numeri più piccoli di 2^{-126}.

esponente	mantissa	numero
00000000	10...0	2^{-127}
00000000	010...0	2^{-128}
....		
00000000	0...001	2^{-149}

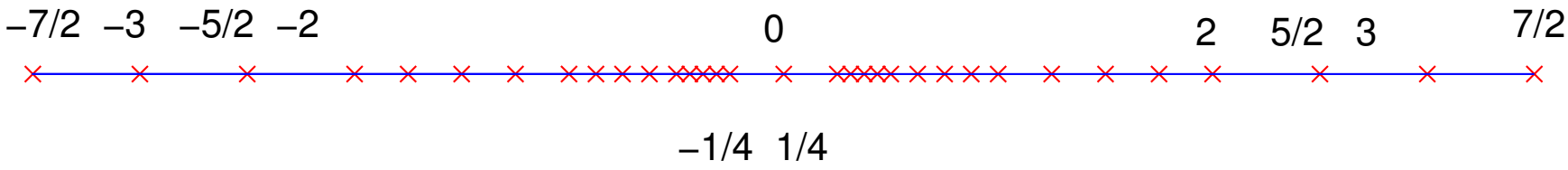
L'insieme dei NUMERI FINITI $F(\beta, t, L, U)$ non è CONTINUO, bensì è un insieme FINITO e LIMITATO.
Possiede esattamente $2(\beta - 1)\beta^{t-1}(U - L + 1) + 1$ elementi

Insieme dei numeri floating point con $\beta = 2, t = 3, L = -2, U = 1$

$F(\beta, t, L, U)$ possiede 33 elementi in totale.

$$\pm 1.a_1 a_2 \cdot 2^p, \quad p = -2, -1, 0, 1; a_1 = 0, 1; a_2 = 0, 1$$

$m \backslash p$	-2	-1	0	1	
1.00	$\pm 1/4$	$\pm 1/2$	± 1	± 2	
1.01	$\pm 5/16$	$\pm 5/8$	$\pm 5/4$	$\pm 5/2$	$+0$
1.10	$\pm 6/16$	$\pm 3/4$	$\pm 3/2$	± 3	
1.11	$\pm 7/16$	$\pm 7/8$	$\pm 7/4$	$\pm 7/2$	


−7/2
−3
−5/2
−2
0
2
5/2
3
7/2
−1/4
1/4

Osservazioni

Attorno a 0 si trova un intervallo $(-\beta^L, \beta^L)$ rappresentato come 0 (underflow).

Numeri maggiori di $(2 - \beta^{-t+1})\beta^U$ e minori di $-(2 - \beta^{-t+1})\beta^U$ non sono rappresentabili (overflow).

I numeri più piccoli sono meglio rappresentati.

$$\Downarrow$$

scaling dei dati negli algoritmi.

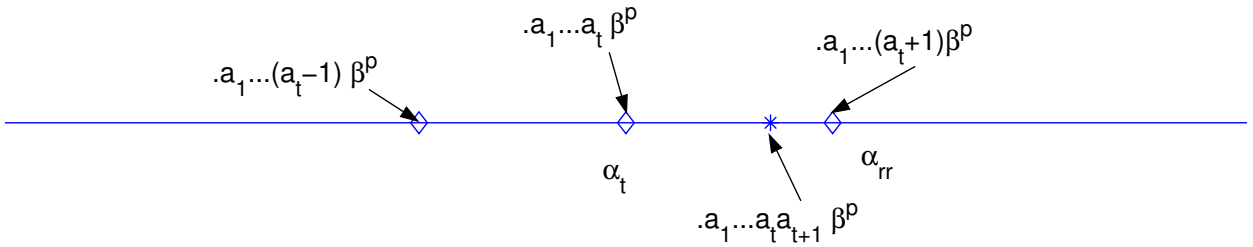
Come rappresentiamo un numero reale che non appartiene ad F?

Sia $\alpha \in \mathbb{R}$

$$\alpha = (.a_1 a_2 ... a_t a_{t+1} a_{t+2} .....)\beta^p$$

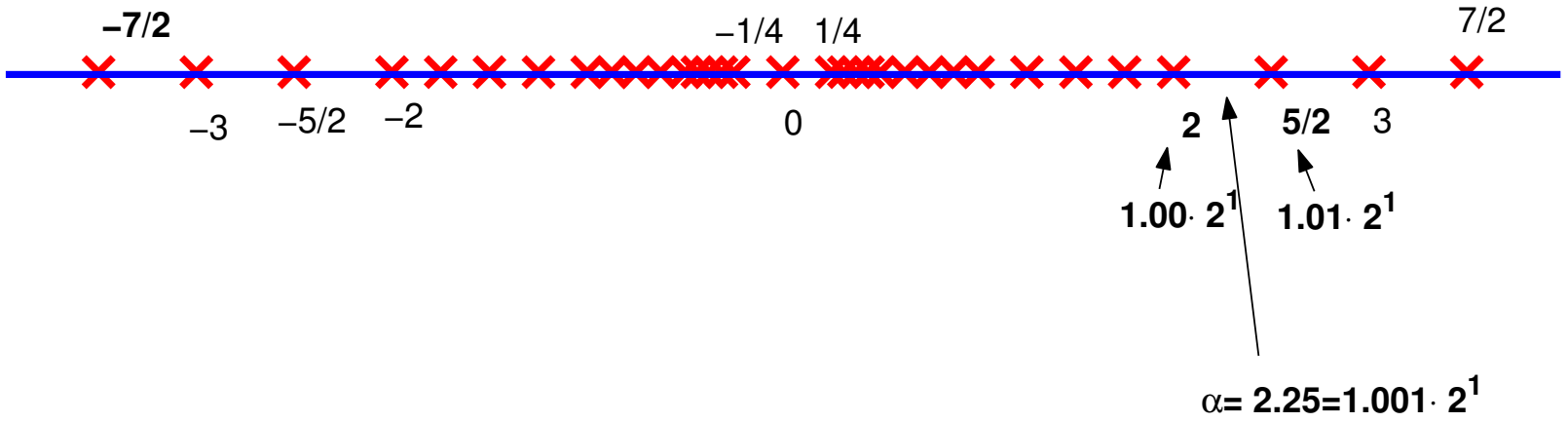
possiamo approssimare α **con un elemento di** $F(\beta, t, L, U)$

ARROTONDAMENTO O TRONCAMENTO



Osservazione

$\beta = 2, t = 3, L = -2, U = 1$



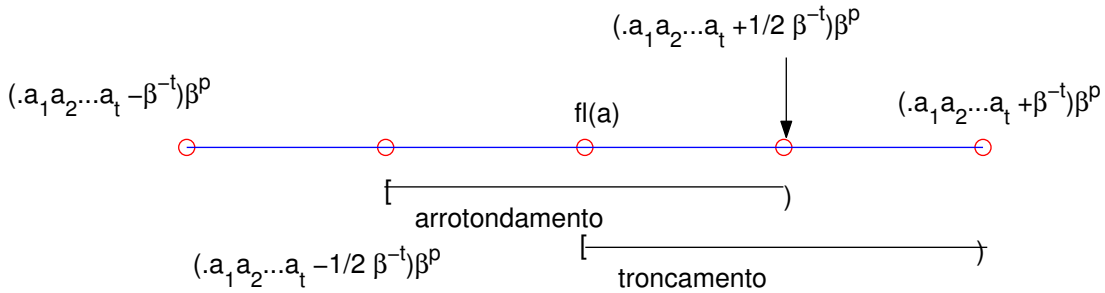
troncamento $1.00 \cdot 2^1$

arrotondamento $1.01 \cdot 2^1$

Ma anche 1.001100
1.001111
1.00111101
$\ldots$

Conseguenza 1

Ogni elemento di F rappresenta se stesso e un intero intervallo di numeri reali.



Definizione di $fl(\alpha)$:
Se $\alpha \in F$, $fl(\alpha) = \alpha$.
Se $\alpha \notin F$, $|fl(\alpha) - \alpha| \leq |y - \alpha| \quad \forall y \in F$.

$fl(\alpha)$ è il numero di F più prossimo ad α.

Conseguenza 2

Abbiamo commesso un errore nella rappresentazione di α

Definizione di errore

Sia $\alpha \in \mathbb{R}$ e α^* una sua approssimazione.

- **ERRORE ASSOLUTO: $E_a = |\alpha - \alpha^*|$;**

- **ERRORE RELATIVO ($\alpha \neq 0$): $E_r = \frac{E_a}{|\alpha|}$;**

ESEMPI.

$\alpha = 0.3\ 10^1 \quad \alpha^* = 0.31\ 10^1 \qquad E_a = 0.1 \qquad E_r = 0.3333..\ 10^{-1}$

$\alpha = 0.3\ 10^{-3} \quad \alpha^* = 0.31\ 10^{-3} \quad E_a = 0.1\ 10^{-4} \quad E_r = 0.3333..\ 10^{-1}$

$\alpha = 0.3\ 10^4 \quad \alpha^* = 0.31\ 10^4 \qquad E_a = 0.1\ 10^3 \quad E_r = 0.3333..\ 10^{-1}$

TEOREMA SULL'ERRORE DI RAPPRESENTAZIONE

Sia $\alpha \in \mathbb{R}, \alpha \neq 0$.

$$\left|\frac{fl(\alpha) - \alpha}{\alpha}\right| \leq k\beta^{1-t}$$

con $k = 1$ nel caso di troncamento e $k = 1/2$ nel caso di arrotondamento, oppure

$$\boxed{fl(\alpha) = \alpha(1+\epsilon) \quad |\epsilon| \leq k\beta^{1-t}.}$$

$u = k\beta^{1-t}$ si dice PRECISIONE DI MACCHINA e dipende solo da β e da t. Essa è caratteristica del calcolatore che si usa e rappresenta il più piccolo numero sentito dalla macchina relativamente, ossia è caratterizzato dal fatto che

$$\mathrm{fl}(1+\mathrm{u}) > 1 \quad \textbf{mentre } \mathrm{fl}(1+\mathrm{a}) = 1 \ \forall \mathrm{a} < \mathrm{u}$$

Infatti

$$(.1\beta^1 + k\beta^{1-t}) = \beta(\beta^{-1} + k\beta^{-t}) > 1$$

Ordine di accuratezza

Definizione. La funzione $f(x)$ è detta un o-piccolo della funzione $g(x)$ per $x \to x_0$ e denotata con $f(x) = o(g(x))$ se esiste una funzione $k(x) \geq 0$ tale che

$$|f(x)| \leq k(x)|g(x)|, \qquad \lim_{x \to x_0} k(x) = 0$$

In tal caso si dice che $f(x)$ è trascurabile rispetto a $g(x)$ per x che tende a x_0.
In pratica f/g tende a 0 per $x \to x_0$. La notazione $f(x) = o(1)$ indica che $f(x)$ tende a 0 per $x \to x_0$.
Definizione. La funzione $f(x)$ è detta un O-grande della funzione $g(x)$ per $x \to x_0$ e denotata con $f(x) = O(g(x))$ se esiste una costante $C > 0$ tale che

$$|f(x)| \leq C|g(x)|$$

per x in un intorno di x_0.
In pratica f/g si mantiene limitato in un intorno di x_0. La notazione $f(x) = O(1)$ indica che $f(x)$ si mantiene limitata in un intorno di x_0.

Per esempio, date le funzioni $f(x) = \frac{x^3}{1+x}$ e $g(x) = x^2$, avremo per $x \to 0$:

$$\frac{x^3}{1+x} = O(x^2)$$

Infatti $\frac{x^3}{1+x} \leq \frac{x^3}{x} = x^2$ per $x \geq 0$. In pratica la notazione O-grande consente di descrivere il comportamento di una funzione in termini di funzioni elementari note ($x^n, x^{1/n}, a^x, log_a x, ...$).

Definizione. La successione $\{x_n\}$ è detta un O-grande della successione $\{y_n\}$ se esistono costanti C ed N tali che

$$|x_n| \leq C|y_n| \qquad n \geq N$$

Ad esempio la successione

$$\frac{n^2-1}{n^3} = O(\frac{1}{n})$$

perchè $(n^2-1)/n^3 \leq n^2/n^3 = 1/n$ per $n \geq 1$.