

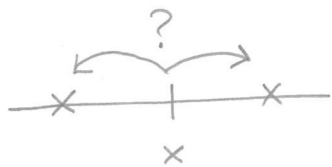
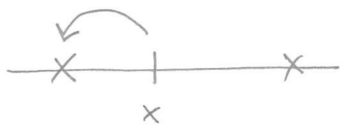
2) come il calcolatore utilizza gli elementi di  $F(\beta, m)$ ...

... per APPROSSIMARE numeri reali

• funzione ARROTONDATO  $rd: \mathbb{R} \rightarrow F(\beta, m)$

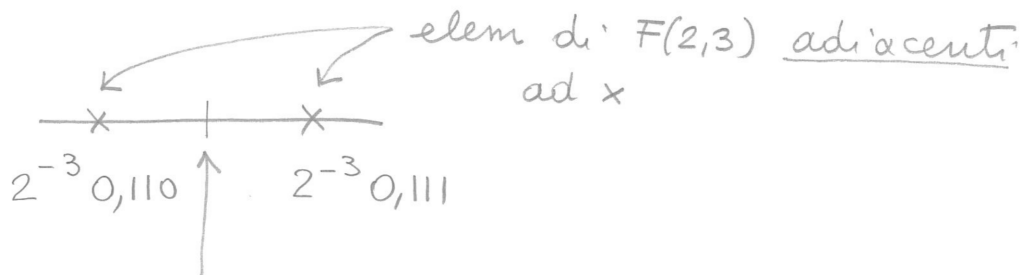
def:  $\forall x \in \mathbb{R}, rd(x) = \dots$

... l'elemento di  $F(\beta, m)$  più vicino ad  $x$



o, se c'è ambiguità, quello dei due che ha frazione che termina con 0 ( $\beta = 2$ )

Es:  $F(2, 3)$ ,  $x = \frac{1}{10}$ ,  $b = -3$ ,  $g = 0, \overline{1100}$



punto medio  
 $= 2^{-3} 0,1101$   $(> x) \Rightarrow rd(x) = 2^{-3} 0,110$

Oss Se  $\beta$  pari e  $m \geq 2$  allora: se l'ultima cifra della frazione di  $\xi \in F(\beta, m)$  è PARI (rispett. DISPARI), l'ultima cifra della frazione del successore di  $\xi$  è DISPARI (rispett. PARI).

se  $\beta$  dispari non è vero!

Es: elem consecutivi in  $F(3, 2)$  sono

$3^0$  0,1<sup>0</sup> pari

$3^0$  0,1<sup>1</sup> dispari

$3^0$  0,1<sup>2</sup> pari

$3^0$  0,2<sup>0</sup> PARI!

• td non è una funzione che il calcolatore mette a disposizione dell'utente, ma è indispensabile per capire come...

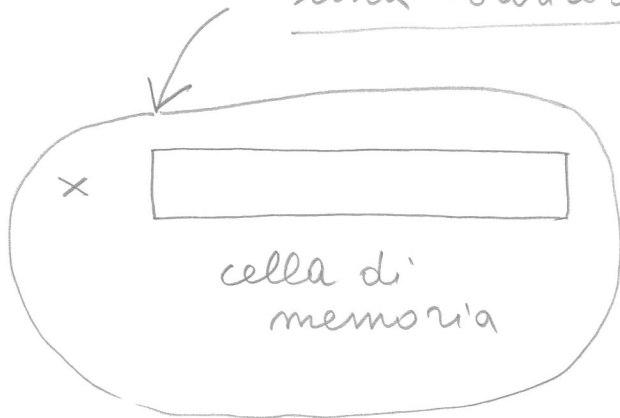
1) il calcolatore "legge" i numeri reali

2) il calcolatore fa operazioni sugli elementi di  $F(\beta, m)$ .

Es ① :  $x = 0,1$  (comando SciLab)

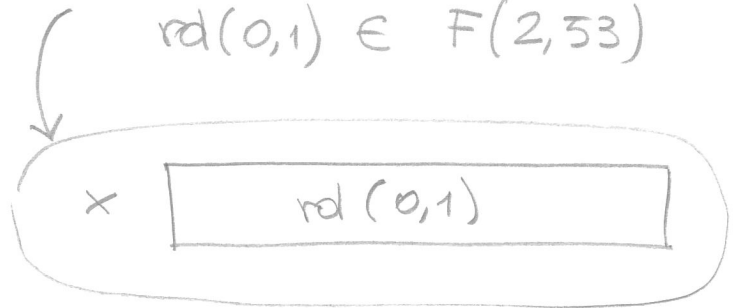
effetto...

a) se non esiste già, viene creata  
una variabile di nome  $x$



b) alla variabile viene assegnato il valore

$$\text{rd}(0,1) \in F(2,53)$$



Il calcolatore APPROSSIMA il numero reale con il suo arrotondato in  $F(\beta, m)$

Pb: che errore viene commesso?

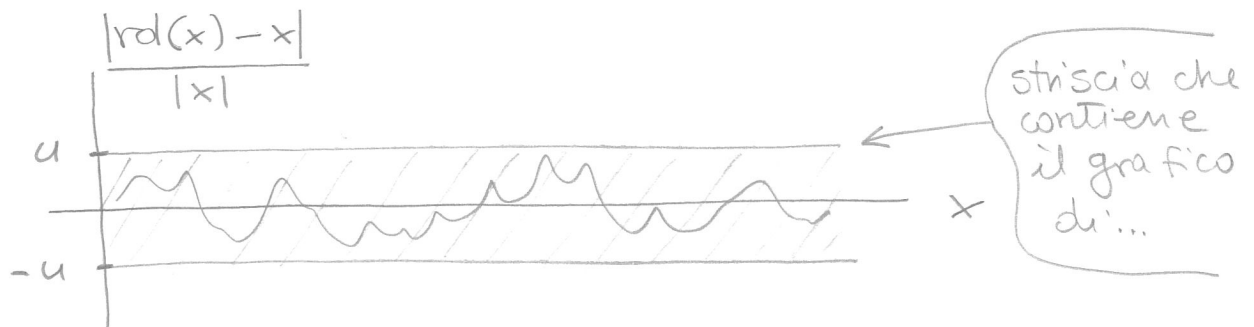
Soluzione:

$$\text{Teo: } \forall x \in \mathbb{R}, \frac{|\text{rd}(x) - x|}{|x|} \leq \frac{1}{2} \beta^{1-m} \equiv \textcircled{u}$$

PRECISIONE  
di MACCHINA

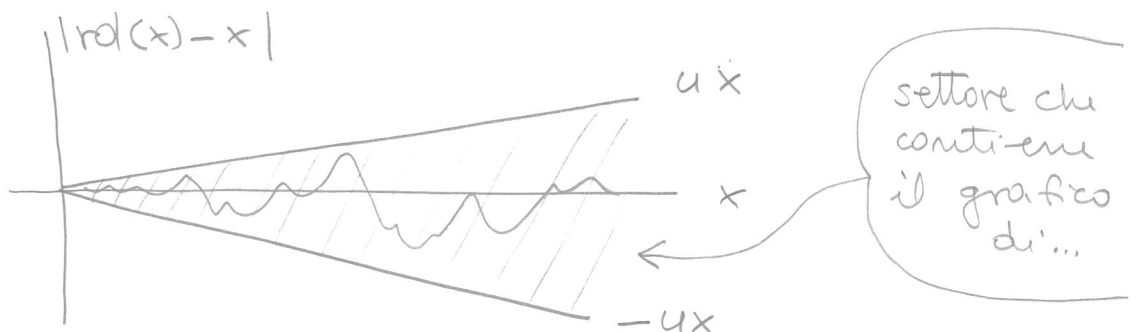
Oss:

- la stima è uniforme nel senso che la quantità che limita l'errore è indipendente da  $x$  (dipende solo dai parametri che definiscono  $F(\beta, m)$ )



- in  $F(2, 53)$  è  $u = \frac{1}{2} 2^{1-53} = 2^{-53} \approx 1.11 \cdot 10^{-16}$   
[Scilab: number-properties ("eps")]
- l'errore considerato è quello RELATIVO;  
per l'errore ASSOLUTO si ottiene la stima (non uniforme!):

$$|rd(x) - x| \leq u |x| \quad (\text{vale } \forall x \in \mathbb{R})$$



Questo accade per come sono distribuiti gli elementi di  $F(\beta, m)$ . Questi ultimi sono pensati appositamente per ottenere la stima uniforme dell'errore relativo.

(Nota: per i numeri in virgola fissa accade l'opposto: la stima dell'errore assoluto è uniforme, quella dell'errore relativo no.)

Es (2):  $\xi$  elem positivo di  $F(2, 53)$   
 $\theta$  successore di  $\xi$  ( $\Rightarrow \theta \in F(2, 53)$ )

•  $\frac{1}{2} \xi \in F(2, 53)$ ,  $\frac{1}{2} \theta \in F(2, 53)$

•  $\frac{1}{2} \xi + \frac{1}{2} \theta \notin F(2, 53)$

$\begin{array}{ccc} \xi & & \theta \\ | & & | \\ \hline & \uparrow & \\ & \frac{\xi + \theta}{2} & \end{array}$

Si ha: ( $\xi = 1$ ):

$c = 1/2 + \text{near\_float}(\text{"succ"}, 1)/2$

il successore di...

$c = 1$

$c == 1$  ← "c è uguale a 1?"

$ans = T$

variabile "di appoggio" che "contiene la risposta"

In Scilab (nel calcolatore) si ha:

$$\forall s_1, s_2 \in F(\beta, m)$$

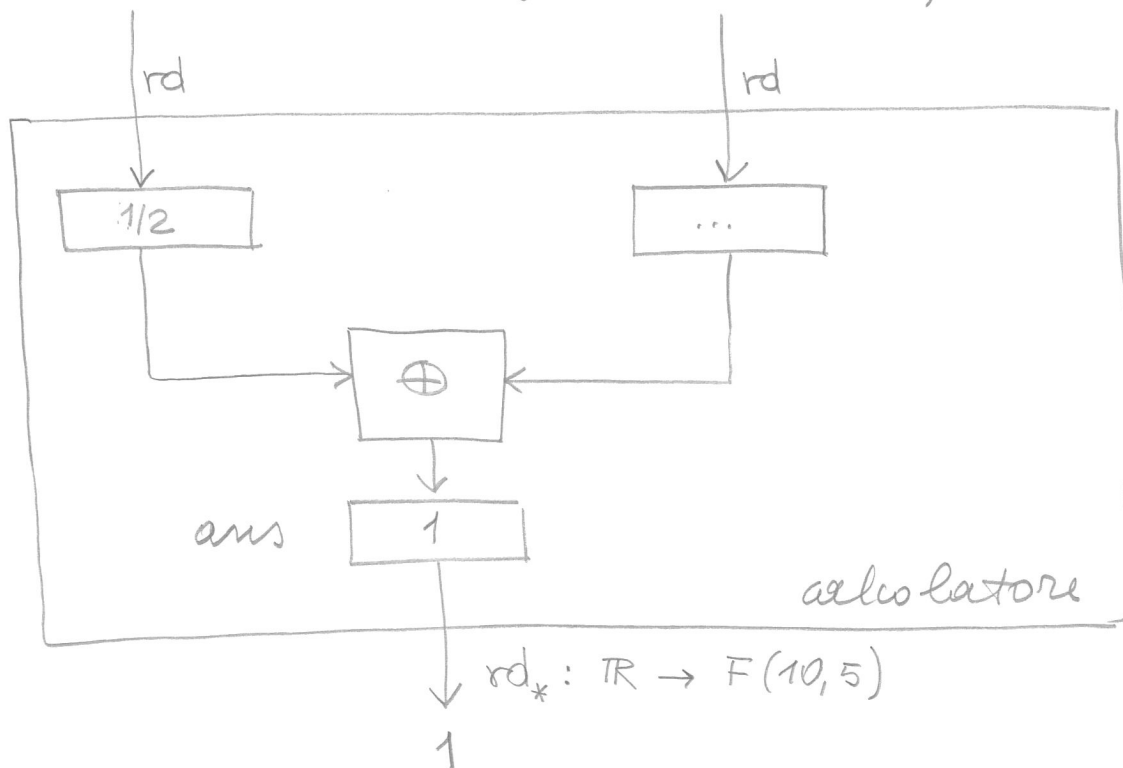
$$s_1 \oplus s_2 = \text{rd} \left( \underbrace{s_1 + s_2}_{\text{somma "esatta"}} \right)$$

"pseudo somma"

si utilizza un simbolo non usuale per chiarezza.

e quello che accade è:

$$1/2 + \text{nearfloat}(\text{"succ"}, 1)/2$$



Es. Nell'Es finale della lez precedente, si utilizzano due sequenze diverse di op per calcolare i valori del polinomio  $(x-2)^{13}$ . Le due sequenze sono equivalenti in IR (ovvero "in teoria") ma NON lo sono operando in  $F(2,53)$  (ovvero "con il calcolatore"), come evidente dal grafico allegato (in cui f\_test\_2 è relativo al calcolo dei valori con  $(x-2)^{13}$  e f\_test\_2\_h con horner...). Dunque il comportamento del calcolatore è una ragionevole conseguenza dell'aver applicato il metodo di binomio a DUE FUNZIONI DIVERSE.

VERSE.

[Scilab help >> Elementary Functions > Floating point > number\\_properties](#)

## number\_properties

determine floating-point parameters

### Calling Sequence

```
pr = number_properties(prop)
```

### Arguments

**prop** string

**pr** real or boolean scalar

### Description

This function may be used to get the characteristic numbers/properties of the floating point set denoted here by  $F(b,p,e_{min},e_{max})$  (usually the 64 bits float numbers set prescribe by IEEE 754). Numbers of  $F$  are of the form:

```
sign * m * b^e
```

$e$  is the exponent and  $m$  the mantissa:

```
m = d_1 b^(-1) + d_2 b^(-2) + ... + d_p b^(-p)
```

$d_i$  the digits are in  $[0, b-1]$  and  $e$  in  $[e_{min}, e_{max}]$ , the number is said "normalised" if  $d_1 \neq 0$ . The following may be gotten:

```
prop = "radix"
    then pr is the radix b of the set F
prop = "digits"
    then pr is the number of digits p
prop = "huge"
    then pr is the max positive float of F
prop = "tiny"
    then pr is the min positive normalised float of F
prop = "denorm"
    then pr is a boolean (%t if denormalised numbers are used)
prop = "tiniest"
    then if denorm = %t, pr is the min positive denormalised number else pr = tiny
prop = "eps"
    then pr is the epsilon machine ( generally (b^(1-p))/2 ) which is the relative max error between a real x (such than |x| in [tiny, huge]) and fl(x), its floating point approximation in F
prop = "minexp"
    then pr is e_min
prop = "maxexp"
    then pr is e_max
```

### Remarks

This function uses the lapack routine `dlamch` to get the machine parameters (the names (radix, digit, huge, etc...) are those recommended by the LIA 1 standard and are different from the corresponding lapack's ones) ; CAUTION: sometimes you can see the following definition for the epsilon machine :  $eps = b^{(1-p)}$  but in this function we use the traditional one (see `prop = "eps"` before) and so  $eps = (b^{(1-p)})/2$  if normal rounding occurs and  $eps = b^{(1-p)}$  if not.

### Examples

```
b = number_properties("radix")
eps = number_properties("eps")
```

### See Also

- [nearfloat](#) — get previous or next floating-point number
- [frexp](#) — dissect floating-point numbers into base 2 exponent and mantissa



