LABORATORIO DI PROGRAMMAZIONE
Corso di laurea in matematica

4 – LA RAPPRESENTAZIONE DEI DATI (2)

Marco Lapegna
Dipartimento di Matematica e Applicazioni
Universita' degli Studi di Napoli Federico II

wpage.unina.it/lapegna

1

Dalla precedente lezione

Rappresentazione in memoria di alcuni tipi di dati fondamentali

- Il tipo intero (basato sul sistema di numerazione binario)
- Il tipo alfanumerico (basato sul codice ASCII)
- Il tipo logico (per le operazioni logiche)

2

Il tipo di dato reale

- Principale tipo di dato per il calcolo scientifico
- Utilizzato per rappresentare numeri decimali
- Basato sulla rappresentazione scientifica

Esempi:

$35.267 = 0.35267 \cdot 10^2$
$0.00523 = 0.523 \cdot 10^{-2}$

In questo modo la "struttura" della rappresentazione e' sempre la stessa
ed e' il punto decimale che si sposta
(rappresentazione **floating point**)

3

Il tipo di dato reale

In generale

Fissata una **base** $b > 1$ $(b \in N)$

$$x = f \cdot b^e$$

$f \in R$ numero reale $\quad$ $e \in Z$ numero relativo

Osservazione

$$\pi = 3.1415... \cdot 10^0 = 0.31415... \cdot 10^1 = 0.031415 \cdot 10^2$$

Cosi' definita la rappresentazione non e' unica

4

Soluzione

Per rendere unica la rappresentazione si impone:

- Parte intera di *f* sempre uguale a 0
- Prima cifra decimale di *f* sempre diversa da 0

Cioe'

$$1 / b \leq |f| < 1$$

Esempi: b =10

$\pi = 0.31415... \cdot 10^1$

$0.00325 = 0.325 \cdot 10^{-2}$

$35782.9 = 0.357829 \cdot 10^5$

Rappresentazione floating point **normalizzata**

Definizioni

Fissato $b > 1$ ogni $x \in R$ (x numero reale)
puo' essere rappresentato in unico modo

$$x = f \cdot b^{e}$$

mediante la rappresentazione floatint point normalizzata

La parte decimale di f e' detta **mantissa** e e' detto **esponente**

Esempio: $b = 10$

$35.267 = 0.35267 \cdot 10^2$

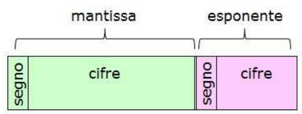
mantissa = 35267 esponente = 2

Rappresentazione in memoria

Come rappresentare il tipo di dato reale in memoria?

E' necessario **rappresentare** separatamente **mantissa** ed **esponente** (ed i rispettivi segni)

Queste informazioni devono trovare spazio in un parola di memoria



Organizzazione della parola di memoria per la rappresentazione del tipo di dato reale

Esempio

$b = 2$ $L = 8$

$x = 6.5$ cioe' $x = 110.1$
R.f.p.n. $x=+0.1101 \cdot 2^{11}$

$y = -0.21875$ cioe' $y = -0.00111$
R.f.p.n. $y=-0.1110 \cdot 2^{-10}$

$z = 5.75$ cioe' $z = 101.11$
R.f.p.n. $z=+0.10111 \cdot 2^{11}$

$w = 9.0$ cioe' $w = 1001.0$
R.f.p.n. $w=0.1101 \cdot 2^{100}$

x	0	1	1	0	1	0	1	1
y	1	1	1	1	0	1	1	0

z cifre mantissa > 4 !!

w cifre esponente > 2 !!

Alcune rappresentazioni

Limiti alla rappresentabilita’

La rappresentabilita’ di un dato di tipo reale dipende
dalla lunghezza della parola

Non tutti i numeri sono rappresentabili

2 vincoli

- sul numero di cifre della mantissa
- sul numero di cifre dell’esponente

Primo vincolo

Sia t il numero di cifre a disposizione per le cifre della mantissa

t = precisione del sistema aritmetico

Tutti i numeri reali che hanno piu’ di t cifre
non possono essere rappresentati esattamente

Primo vincolo:

Precisione finita

Secondo vincolo

Sia
•b la base
•q il numero di cifre a disposizione per le cifre dell’esponente

$E_{min} = -(b^{q-1} - 1)$ = minimo esponente rappresentabile

$E_{max} = b^{q-1} - 1$ = massimo esponente rappresentabile

Gli esponenti e al di fuori del range
$E_{min} \leq e \leq E_{max}$

non possono essere rappresentati esattamente

Secondo vincolo:

Range limitato

Esempio

$b = 10$ $t=3$ $E_{min} = -5$ $E_{max} = 5$

Il massimo numero reale rappresentabile e’
$R = 0.999 \cdot 10^{5}$

Il minimo numero reale rappresentabile e’
$R = 0.100 \cdot 10^{-5}$

L’insieme finito dei numeri reali **esattamente** rappresentabili e’ detto

Insieme dei numeri macchina

I numeri macchina

L'insieme dei numeri macchina F e' carattarizzato da quattro parametri

- *b* la base
- *t* la precisione
- E_{min} l'esponente minimo
- E_{max} l'esponente massimo

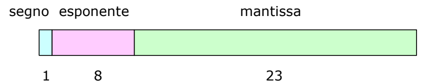
costanti macchina del sistema aritmetico floating point a precisione finita

Lo standard IEEE

Nel 1989 viene definito lo standard IEEE 754
che costituisce il formato di rappresentazione f.p. piu' diffuso

- Disponibile in singola (32 bit), doppia (64 bit) e quadrupla (128 bit) precisione
- Permette anche la rappresentazione di situazioni eccezionali come infinito, forme indeterminate e i numeri denormalizzati

Es. singola precisione (32 bit)
- segno = 1 bit
- esponente = 8 bit
- mantissa = 23 bit



standard IEEE a 32 bit

$$x = (-1)^s \cdot f \cdot 2^e$$

esempio: x = 51.625

$$x = (-1)^s \cdot f \cdot 2^e$$

determiniamo **s, e, f** della rappresentazione di x

- **bit del segno della mantissa**
  S = **0**
- **mantissa**
  51.625 = 51 + 0.625

  51 = 110011 0.625 = 0.101

  f = 110011.101 = 1.**10011101**x2^5

osservazione:
il primo bit della mantissa **e' sempre uguale a 1**, e per tale motivo si sottintende (guadagnando un bit di precisione). E' il cosiddetto **bit implicito**

esempio: x = 51.625

Nella **rappresentazione** IEEE l'esponente e' rappresentato come
intero per eccesso 127

- **esponente** (eccesso 127)

  da f = 1.10011101 · 2^5 si ha **e** = 5 + 127 = **132**

  e = 10000100

la rappresentazione di 51.65 in formato standard IEEE 754 e'

0	10000100	10011101000000000000000

osservazione

La rappresentazione degli esponenti per eccesso 127
trasforma l'intervallo [0, 255] nell'intervallo [-127, 128]

in realta' gli esponenti **rappresentabili** nell'artimetica standard IEEE
appartengono all'intervallo **[-126, 127]**

gli esponenti **-127** (corrispondente a 00000000)
e **128** (corrispondente a 11111111)

vengono usati per gestire **situazioni eccezionali**
(e = 0 numeri denormalizzati, cioe' bit implicito=0
e = 255 infinito e forme indeterminate)

IEEE singola precision (32 bit)
b = 2
t = 23 (+1)
$\left.\begin{array}{l}\textbf{Emin = -127}\\ \textbf{Emax = 128}\end{array}\right\}$ 8 bit

IEEE singola precision (64 bit)
b = 2
t = 52 (+1)
$\left.\begin{array}{l}\textbf{Emin = -1023}\\ \textbf{Emax = 1024}\end{array}\right\}$ 11 bit

Problema

Se un numero reale x non rispetta i precedenti vincoli
e' possibile rappresentarlo mediante un numero macchina?

Piu' precisamente

dato $x = f \cdot b^{e}$

Cosa accade se:

- $e > E_{max}$ oppure $e < E_{min}$
- Numero di cifre di $x > t$

Overflow e underflow

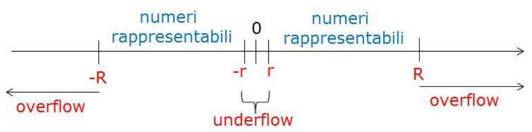
$x = f \cdot b^{e}$

La situazione in cui
$e > E_{max}$
si chiama **OVERFLOW**

In questo caso la
rappresentazione e' impossibile

La situazione in cui
$e < E_{min}$
si chiama **UNDERFLOW**

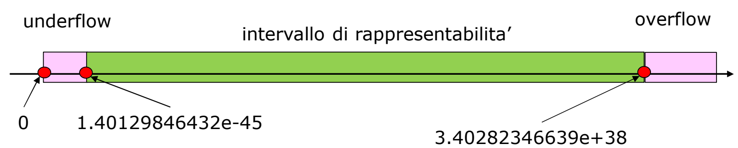
In questo caso
x si rappresenta con lo 0



Il range di rappresentabilita' del tipo di dato reale

esempio: aritmetica IEEE

- minimo reale rappresentabile (non normalizzato)
  - s = 0
  - e = 0
  - m = 0.00000000000000000000001 = 1.40129846432e-45
- massimo reale rappresentabile
  - s = 0
  - e = 11111110
  - m = 1.11111111111111111111111 = 3.40282346639e+38



Rappresentazione floating point

$x = f \cdot b^{e}$

Se
$E_{min} < e < E_{max}$

Ma il numero di cifre della mantissa $> t$

Si rappresenta x con un numero macchina vicino

Esempi: $b = 10 \quad t=3$

$X = 0.1251 \cdot 10^2 \rightarrow fl(x)= 0.125 \cdot 10^2$

$X = 0.1259 \cdot 10^2 \rightarrow fl(x)= 0.126 \cdot 10^2$

fl(x) e' la **rappresentazione floating point** di *x*

Arrotondamento e troncamento

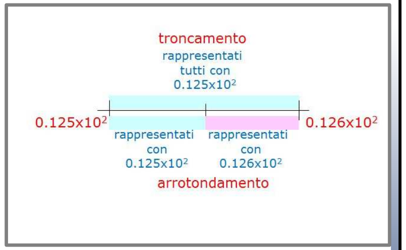
Due metodi per *fl(x)*

- **Troncamento**
(si eliminano le cifre decimali in eccesso)

$fl(0.1251 \cdot 10^2) = 0.125 \cdot 10^2$
$fl(0.1259 \cdot 10^2) = \mathbf{0.125} \cdot 10^2$

- **Arrotondamento**
(si approssima x con il numero macchina piu' vicino)

$fl(0.1251 \cdot 10^2) = 0.125 \cdot 10^2$
$fl(0.1259 \cdot 10^2) = \mathbf{0.126} \cdot 10^2$



Arrotondamento e troncamento

Operazioni sul tipo di dato reale

- Addizione f.p.
- Sottrazione f.p.
- Moltiplicazione f.p.
- Divisione f.p.

Analogamente al tipo di dato intero
il risultato e' definito solo se

- Gli operandi sono rappresentabili
- Il risultato e' rappresentabile

Riassumendo

- I numeri reali sono rappresentati in memoria mediante la notazione floating point normalizzata
- E' necessario rappresentare mantissa e esponente
- Non tutti i numeri sono esattamente rappresentabili
- Esiste un massimo ed un minimo reale positivo rappresentabile
- I numeri con numero di cifre maggiore della precisione vengono rappresentati mediante troncamento o arrotondamento

Un po' di storia (4)

John Backus (1924-2007)

- Americano, laureato in matematica nel 1949, lavora per l'IBM dall'anno successivo con il compito di programmatore (in linguaggio macchina e assembly)
- Nel 1954 dirige il team per lo sviluppo del Fortran, primo moderno linguaggio di programmazione ad alto livello, usato prima per l'IBM 704 e poi divenuto uno standard.
- Collabora allo sviluppo di altri linguaggi (Algol, FL) e di un metodo per la specifica dei linguaggi di programmazione (BNF)
- Riceve numerosi premi tra cui il Turing Award (1977) per i suoi contributi allo sviluppo dei linguaggi di programmazione.


L'IBM 704 (courtesy of Computer History Museum)


John Backus (courtesy of Computer History Museum)