Argomenti trattati

- Rappresentazione dei numeri
- Calcoli in binario
- Rappresentazione di numeri naturali
- Rappresentazione di numeri relativi
- Rappresentazione di numeri reali (Virgola mobile)

Rappresentazione dei numeri

- Rappresentazione di insiemi numerici mediante insiemi *finiti* di stringhe di bit
- Problemi:
  - Insiemi numerici discreti (naturali, relativi): rappresentazione di numeri troppo grandi (*overflow*)
  - Insiemi numerici continui (reali): approssimazione al numero rappresentabile più vicino
  - Approssimazione a zero (*underflow*)

Rappresentazione dei numeri

- Lunghezze tipiche delle stringhe usate per rappresentare numeri

lunghezza stringa	numero di stringhe diverse	ordine di grandezza
8	256	$10^2 \div 10^3$
16	65536	$10^4 \div 10^5$
32	4294967296	$10^9 \div 10^{10}$
64	18446744073709551616	$10^{19} \div 10^{20}$

Operazioni aritmetiche

- Somme:

riporto

$$0 + 0 = 0, \quad 0 + 1 = 1, \quad 1 + 0 = 1, \quad 1 + 1 = 10$$

- Esempi:

$$\begin{array}{r} 1001 \; + \\ 101 \; = \\ \hline 1110 \end{array} \qquad \begin{array}{r} 1011 \; + \\ 111 \; = \\ \hline 10010 \end{array}$$

- Ulteriore caso elementare:

$$1 + 1 + 1 = 11$$

riporto

Operazioni aritmetiche

- Moltiplicazioni:

	0	1
0	0	0
1	0	1

- Esempio:

$$
\begin{array}{r}
1001\ \times \\
101\ = \\
\hline
1001 \\
1001 \\
\hline
101101
\end{array}
$$

Rappresentazione di numeri naturali

- Il numero *n* di bit disponibile in pratica è limitato (valori tipici di *n*: 8, 16, 32, 64)
- Si può rappresentare un insieme finito di 2^n numeri (da 0 a 2^n -1)
- Esempio: $n = 8$

$$
\begin{array}{r l}
11010101 & + \\
1000111 & = \\
\hline
100011100 &
\end{array}
\qquad\qquad
\begin{array}{r l}
213 & + \\
71 & = \\
\hline
284 &
\end{array}
$$

overflow

Il risultato (su 8 bit) è: 00011100 = 28 !

Altre proprietà notevoli

- Estensione della rappresentazione quando un numero richiede meno di *n* bit:

  si aggiungono cifre 0 a sinistra

- Esempio (n = 8):

13 (1101_2)	si rappresenta con	00001101
142 (10001110_2)	si rappresenta con	10001110
81 (1010001_2)	si rappresenta con	01010001

Altre proprietà notevoli

- Rappresentazione di 2^k:

  solo il $(k+1)$ bit da destra è uguale a 1

- Esempi (n = 8): $4 = 2^2$ 00001000

  $32 = 2^5$ 00100000

- Rappresentazione di $x \cdot 2^k$ e di $x / 2^k$:

  la rappresentazione "scorre" a sinistra e destra di k bit, rispettivamente

- Esempio (n = 8): 24 00011000

  8 00000011

  96 01100000

Numeri relativi

- Rappresentazione in *segno e modulo*

  $$(\pm\ , a_m\ , a_{m-1}\ , \ldots\ , a_1\ , a_0\ )$$

  0 indica il segno positivo e 1 il segno negativo
- Esempio (n = 8):

  27 è rappresentato da 00011011

  -27 è rappresentato da 10011011
- L'*overflow* si ha per valori maggiori in modulo di 2^{n-1}
- Le stringhe 00000000 e 10000000 rappresentano entrambe lo 0

Numeri relativi

- Rappresentazione in *complementi a 2* (su *n* bit):
  - lo zero e i numeri positivi fino a $2^{n-1}-1$ sono rappresentati normalmente
  - i numeri negativi fino a -2^{n-1} sono rappresentati dai numeri in $[\, 2^{n-1} , 2^{n} -1 \,]$
  - il rappresentante di un numero negativo x è $2^{n} - |x|$
- L'*overflow* si ha per valori maggiori di $2^{n-1}-1$ e minori di -2^{n-1} (l'intervallo e asimmetrico)
- Lo 0 è rappresentato solo dalla stringa 00000000

Rappresentazione in complemento a 2

- Il primo bit indica il segno (0 i positivi, 1 i negativi)
- Cambio di segno (complemento a 2 della rappresentazione):

  11001011 (rappresenta -53)

  00110100 + 1 = 00110101 (rappresenta 53)

- Somma algebrica (somma dei rappresentanti e resto modulo 2^n):

  -53 + 71 = 18

11001011 +	
01000111 =	
100010010	**risultato somma**
00010010	**resto modulo 2^n**

Numeri reali

- Rappresentazione di un insieme continuo
- Fissata una base b, un numero x è rappresentato dalla coppia:

$$(m , e)$$

tale che:

$$x = m \times b^{e}$$

- Il metodo prende il nome di *codifica in virgola mobile*
- m viene detto *mantissa*, e viene detto *esponente*

Normalizzazione

- Per ogni numero esistono infinite coppie che lo rappresentano
- Esempio (*b* = 10):

  346.09801 è rappresentato da

  (346.09801 , 0) oppure

  (346098.01 , -3) oppure

  (0.034609801 , 4) ... ecc.

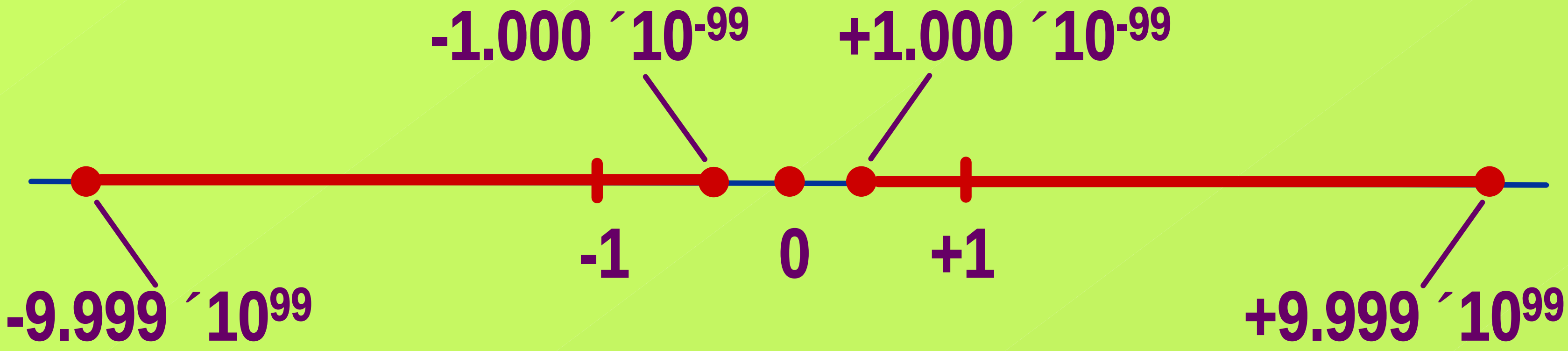
- Si fissa la posizione della virgola rispetto alla prima *cifra significativa* (di solito subito dopo)
- Unico rappresentante: (3.4609801 , 2)

Intervallo di rappresentazione

- Esempio con *b* = 10, usando 4 cifre per *m* e 2 per *e* (esclusi i segni), l'insieme rappresentato è:

$$[\,-9.999\times10^{99}\,,\,-1.000\times10^{-99}\,]\cup\{\,0\,\}\cup$$

$$[\,+1.000\times10^{-99}\,,\,+9.999\times10^{99}\,]$$



Approssimazione

- Esempio con b = 10, usando 4 cifre per m e 2 per e (esclusi i segni), i numeri positivi realmente rappresentati sono (analogamente per i negativi):

1.000×10^{-99} , 1.001×10^{-99} , 1.002×10^{-99} ,

…. …

9.997×10^{-1} , 9.998×10^{-1} , 9.999×10^{-1} ,

1.000×10^{0} , 1.001×10^{0} , 1.002×10^{0} ,

…. …

9.998×10^{99} , 9.998×10^{99} , 9.999×10^{99}

- La densità (e quindi l'approssimazione) non è costante

Errore relativo costante

- L'errore relativo massimo ($\Delta x / x$) è grossomodo costante

$$\frac{1.001\times 10^{-99} - 1.000\times 10^{-99}}{1.000\times 10^{-99}} = 0.001$$

$$\frac{1.001\times 10^{99} - 1.000\times 10^{99}}{1.000\times 10^{99}} = 0.001$$

$$\frac{9.999\times 10^{99} - 9.998\times 10^{99}}{9.998\times 10^{99}} = 0.0001$$

- Errori assoluti grandi su numeri grandi e errori assoluti piccoli su numeri piccoli

Overflow e underflow

- L'errore relativo dipende dal numero di cifre della mantissa
- Nel caso precedente 4 cifre corrispondono ad un errore relativo di $10^{-3} \div 10^{-4}$
- Gli estremi dell'intervallo di rappresentazione dipendono dal numero di cifre dell'esponente
- Nel caso precedente 2 cifre corrispondono a un *overflow* per numeri maggiori in modulo di 10^{99} e un *underflow* per numeri minori in modulo di 10^{-99}

Calcolo in virgola mobile (cenni)

- Somma algebrica e Moltiplicazione
- Propagazione degli errori dovuti all'approssimazione
- La presenza di approssimazioni esclude la possibilità di prendere decisioni in base all'uguaglianza tra quantità rappresentate in virgola mobile
- Singola e doppia precisione

rappresentazione	**numero di bit (*m* + *e*)**	**numero di cifre significative**	**intervallo (in modulo)**
singola precisione	24 + 8	$7 \div 8$	$10^{-38} \div 10^{38}$
doppia precisione	53 + 11	$16 \div 17$	$10^{-308} \div 10^{308}$