# Laboratorio di Programmazione

Laurea in Bioinformatica

Web: http://www.scienze.univr.it/fol/main?ent=oi&id=49485

Docente: *Carlo Drioli*
Email: *carlo.drioli@univr.it*

2009/2010

Lucidi a cura di
Carlo Drioli

*Rappresentazioni Numeriche*

# Rappresentazioni numeriche

Classi numeriche generalmente rappresentate nei sistemi di calcolo:

- Interi positivi (Naturali)
- Interi con segno (Relativi)
- Reali esprimibili come rapporto fra interi (Razionali)
- Reali in generale (inclusi Irrazionali e Trascendenti): approssimati

# Sistema di numerazione posizionale

Nel sistema posizionale le cifre assumono valore diverso a seconda della posizione nella stringa numerica

Il valore di una stringa di simboli: $s = d_{n-1}d_{n-2}\cdots d_0$
in una data base *B* e'data dalla formula:

$$s_{baseB} = \sum_{i=0}^{n-1} d_i B^i = d_{n-1}B^{n-1} + d_{n-2}B^{n-2} + \cdots d_0 B^0$$

Esempio:

$$2003_{base10} = 2 \times 10^3 + 0 \times 10^2 + 0 \times 10^1 + 3 \times 10^0$$

Sistemi posizionali principali:

| sistema | $B$ | simboli |
|---|---|---|
| **binario** | 2 | $\{0,1\}$ |
| **ottale** | 8 | $\{0,1,2,3,4,5,6,7\}$ |
| **decimale** | 10 | $\{0,1,2,3,4,5,6,7,8,9\}$ |
| **esadecimale** | 16 | $\{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F\}$ |

# Conversione da base B a base 10

E' la naturale applicazione della formula posizionale.

Esempi:

$$721_8 \rightarrow 7 \times 8^2 + 2 \times 8^1 + 1 \times 8^0 = 7 \times 64 + 17 = 465_{10}$$

$$134_5 \rightarrow 1 \times 5^2 + 3 \times 5^1 + 4 \times 5^0 = 25 + 19 = 44_{10}$$

$$7D1_{16} \rightarrow 7 \times 16^2 + 13 \times 16^1 + 1 \times 16^0 = 25 + 19 = 2001_{10}$$

$$156_5 \rightarrow \ldots \text{ possibile?}$$

# Conversione da base 10 a base B

Un possibile metodo utilizza divisioni successive per *B,* il cui resto fornisce di volta in volta le cifre del numero nella nuova base.

Esempio:

| Operazione | Risultato | Resto |
|---|---|---|
| 25/2 | 12 | 1 |
| 12/2 | 6 | 0 |
| 6/2 | 3 | 0 |
| 3/2 | 1 | 1 |
| 1/2 | 0 | 1 |

$$25_{10} \rightarrow 11001_2$$

# Conversione da base 10 a base B

In pseudocodice:

```
variabili x, B, dec, i: numeri interi;
leggi dec;
leggi B; //nuova base
i = 0;
QUANDO dec != 0 ESEGUI
    x = dec % B;
    scrivi x; //(i+1)-ma cifra, a partire
              //da quella meno significativa
    dec = dec / B; //divisione fra interi!
    i = i + 1;
RIPETI
```

# Rappresentazione di Interi negativi

Esistono varie alternative:

- Modulo e segno
- Complemento a 1
- Complemento a 2
- Eccesso $2^{(m-1)}$

# Rappresentazione di numeri relativi con modulo e segno

- Di n bit a disposizione per codificare il numero, il primo e' usato per il segno e i rimanenti per il modulo.

| $d_n$ | $d_{n-1}$ | $d_{n-2}$ | ... | $d_1$ |
|---|---|---|---|---|

- Con n bit, si possono rappresentare i seguenti intervalli di valori:
  **positivi:** $2^{(n-1)}$ valori, da 0 a $2^{(n-1)}$ -1
  **negativi:** $2^{(n-1)}$ valori, da $-(2^{(n-1)}$ -1) a 0
- Ridondanza: lo zero e'rappresentato 2 volte
- Esempio:

$$-15\colon\ |15| = 1111_2 \rightarrow 15 = 01111_2,\ -15 = 11111_2$$

## Rappresentazione di numeri relativi in complemento a due

- Il complemento a 2 di una rappresentazione binaria si ottiene complementando le cifre (0→1, 1→0) e sommando 1.
- In questa rappresentazione, gli interi positivi sono codificati in modulo e segno, i negativi in complemento a 2.
- Con n bit, si possono rappresentare i seguenti intervalli di valori:
  **positivi:** $2^{(n-1)}$ valori, da 0 a $2^{(n-1)}$ -1
  **negativi:** $2^{(n-1)}$ valori, da $-2^{(n-1)}$ a -1
- Non c'e' ridondanza: lo zero ha una sola rappresentazione
- La rappresentazione in complemento a 2 semplifica le operazioni sugli interi

## Rappresentazione di numeri relativi in complemento a due

- Esempio:
  rappresentazione del numero -5 con 8 bit in complemento a 2

$$00000101\ (5) \rightarrow 11111010 \rightarrow 11111011\ (-5)$$

- Se un numero inizia per 0 e' positivo, altrimenti e' negativo
- Il complemento a 2 di un numero negativo riporta al numero positivo pari al modulo del numero negativo

## Somma di due numeri in complemento a due

- Nella somma di due numeri di segno opposto in Ca2 il segno del risultato e' determinato automaticamente:

$$\begin{array}{rl} 11111\quad 111 & \text{(riporto)} \\ 0000\quad 1111 & (15) \\ +\ 1111\quad 1011 & (\text{-}5) \\ \hline 0000\quad 1010 & (10) \end{array}$$

- Overflow: si verifica overflow se gli ultimi due riporti hanno valore diverso. Esempio:

$$\begin{array}{rl} 0111 & \text{(riporto)} \\ 0111 & (7) \\ +\ 0011 & (3) \\ \hline 1010 & (-6) \end{array}$$

## Numeri con parte frazionaria

- Per estensione della rappresentazione posizionale:

$$s = d_{n-1}d_{n-2}\cdots d_0.d_{-1}d_{-2}\cdots d_{-p} = \sum_{i=-p}^{n-1} d_i B^i$$

- Esempio:

$$0.234 = 2\cdot 10^{-1} + 3\cdot 10^{-2} + 4\cdot 10^{-3}$$

# Conversione di base di numeri con parte frazionaria (da base 10 a B)

- Nel cambio di base di numeri frazionari, la conversione si effettua sulla parte intera con il metodo visto per i numeri interi, e sulla parte frazionaria con procedimento duale. Questo si basa su moltiplicazioni successive per *B* e troncamento, fornendo di volta in volta le cifre della parte frazionaria nella nuova base. La procedura termina quando si e'raggiunto il n. di cifre desiderato (precisione)

- Esempio: $0.8125_{10} \rightarrow (0.1101)_2$

| Operazione | risultato | troncamento |
|---|---|---|
| $0.8125 \cdot 2$ | 1.625 | 1 |
| $0.625 \cdot 2$ | 1.25 | 1 |
| $0.25 \cdot 2$ | 0.5 | 0 |
| $0.5 \cdot 2$ | 1.0 | 1 |

# Conversione di base di numeri con parte frazionaria (da base 10 a B)

- In pseudocodice:

```
variabili x, B, i, m: numeri interi;
varibile dec: numero reale;
leggi dec; //numero <1
leggi B; //nuova base
leggi m; //n. max cifre (precisione)
scrivi "0.";
i = 0;
QUANDO ( i < m ) ESEGUI
    SE dec != 0
        x = dec * B; //N.B.: moltiplicazione con troncamento del ris.
        scrivi x; //(i+1)-ma cifra, a partire da quella piu' significativa
        dec = dec * B - x;
    FINESE
    i = i + 1;
RIPETI
```