Insegnamento di Elementi di informatica (6 CFU)

La rappresentazione dei numeri naturali

ing. Nadia Ranaldo
Facoltà di Ingegneria
Università degli Studi del Sannio

Materiale realizzato sulla base delle trasparenze redatte dalla prof.sa Antonella Santone

Rappresentazione dei numeri

- Naturali
- Interi
- Reali

Elaboratori: rappresentazione finita dei numeri naturali

Indichiamo con
n un numero naturale e con ρ**(n) la sua rappresentazione** mediante un numero finito di simboli

Proprietà auspicabili

1. **n ≠ m ➔ ρ(n) ≠ ρ(m)**

   due numeri diversi ➔ due rappresentazioni diverse

2. **∀n ∃! ρ(n)**

   per ogni numero esiste ed è unica la sua rappresentazione

Notazione posizionale

- Un numero naturale (es. 5) può essere rappresentato in molti modi :
  - cinque, *five*, *5*, V, 
- Rappresentazioni diverse hanno proprietà diverse
  - moltiplicare due numeri in notazione romana è molto più difficile che moltiplicare due numeri in notazione decimale ….
- Noi siamo abituati a lavorare con numeri rappresentati in notazione posizionale in base 10, ovvero rappresentati con dieci simboli: 0,1,2,3,4,5,6,7,8 e 9.

Notazione posizionale (2)

B = base

La base di un sistema di numerazione posizionale corrisponde al numero dei simboli usati per scrivere i numeri

A = {0, … ,B-1} = cifre

n = numero di cifre = lunghezza della sequenza

$I_{rapp} = [0 .. B^n-1]$ intervallo

$|I_{rapp}| = B^n$ cardinalità

Che vuol dire posizionale?

2<u>5</u>3 14<u>5</u>

Le cifre hanno un significato (o un peso) diverso a seconda della posizione

$s = c_{n-1}\ c_{n-2} \dots c_1\ c_0 \qquad c_i \in A$

cifra più significativa cifra meno significativa

Posizione j => peso B^j

Vale quindi la relazione:

$n = c_{n-1}\ B^{n-1}\ c_{n-2}\ B^{n-2} \dots c_1\ B^1\ c_0\ B^0$

Notazione posizionale in base 10

B = 10

A = {0, … ,9}

n = 3

$I_{rapp} = [0 .. 999]$ intervallo

$|I_{rapp}| = 10^3$ cardinalità

Notazione posizionale in base 10 (2)

- La rappresentazione di un numero intero in base 10 è una sequenza di cifre scelte fra 0 1 2 3 4 5 6 7 8 9:
  - es: 23, 118, 4
- Il valore di una rappresentazione $c_N \dots c_0$ è dato da
  $c_N * 10^N + c_{N-1} * 10^{N-1} \dots + c_1 * 10^1 + c_0 * 10^0$
  esempi :
  - $23 = 2*10^1 + 3 * 10^0 = 20 + 3$
  - $118 = 1*10^2 + 1*10^1 + 8 * 10^0 = 100 + 10 + 8$

Notazione posizionale in base 10 (3)

Vediamo alcune proprietà di questa notazione:

Il massimo numero rappresentabile con N cifre è 99….9 (N volte 9, la cifra che vale di più), pari a 10^N-1

- es: su tre cifre il massimo numero rappresentabile è 999 pari a 10^3-1 =1000-1

- Quindi se voglio rappresentare K diverse configurazioni (cioè *0 1 2 …K-1*) mi servono almeno *x* cifre dove 10^x è la più piccola potenza di 10 che supera K
  - es : se voglio 25 configurazioni diverse mi servono almeno 2 cifre perché 10^2=100 è la più piccola potenza di 10 maggiore di 25

Rappresentazione un base due (o binaria)

B = 2

A = {0, 1}

p = 3

$\mathbf{I_{rapp} = [0\ ..\ 7]}$

$\mathbf{|I_{rapp}| = 2^3 = 8}$

000
001
010
011
100
101
110
111

uno uno zero

Notazione posizionale in base 2

- La rappresentazione di un numero intero in base 2 è una sequenza di cifre scelte fra 0 e 1 :
  - es: 10, 110, 1
- Il valore di una rappresentazione $c_N \ldots c_0$ è dato da

  $c_N * 2^N + c_{N-1} * 2^{N-1} \ldots + c_1 * 2^1 + c_0 * 2^0$

  esempi :
  - 10 = $1*2^1 + 0*2^0 = 2$
  - 110 = $1*2^2 + 1*2^1 + 0*2^0 = 4 + 2 + 0 = 6$
  - 1 = $1*2^0 = 1$

Notazione posizionale in base 2 (2)

Per la base due valgono proprietà analoghe a quelle viste per la base 10 :

- Il massimo numero rappresentabile con N cifre è 11….1 (N volte 1, la cifra che vale di più), pari a 2^N-1
  - es: su tre cifre il massimo numero rappresentabile è 111 pari a 2^3-1 = 8 - 1 = 7
- Quindi se voglio rappresentare K diverse configurazioni (cioè *0 1 2 …K-1*) mi servono almeno *x* cifre dove 2^x è la più piccola potenza di 2 che supera K
  - es : se voglio 25 configurazioni diverse mi servono almeno 5 cifre perché 2^5=32 è la più piccola potenza di 2 maggiore di 25

Utile: potenze di 2

$2^0 = 1$
$2^1 = 2$
$2^2 = 4$
$2^3 = 8$
$2^4 = 16$
$2^5 = 32$
$2^6 = 64$
$2^7 = 128$
$2^8 = 256$
$2^9 = 512$
$2^{10} = 1024$

Due problemi

1. Data una stringa di simboli, che numero naturale rappresenta in base dieci?

conversione da una base **B** alla base decimale

2. Dato un numero naturale nella base decimale, come si rappresenta in un'altra base?

conversione dalla base decimale in una base B

Primo problema

$s = c_{p-1} \dots c_{0\ B}$ ➔ n?

1345_{10} ➔ n?

$$n = \sum_{i=0}^{p-1} c_i \cdot B^i = c_{p-1} \cdot B^{p-1} + \dots + c_1 \cdot B^1 + c_0$$

qualunque sia la base **B** e **p** è **il numero di cifre usate**

$c_i \in A$

Esempio

$$n = \sum_{i=0}^{p-1} c_i \cdot B^i$$

Consideriamo la stringa 110 di bit (cifre binarie).
Quale numero naturale rappresenta?

$p = 3$

1	1	0
c_2	c_1	c_0

$$n = \sum_{i=0}^{2} c_i \cdot 2^i = c_2 \cdot 2^2 + c_1 \cdot 2^1 + c_0 \cdot 2^0$$
$$= 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$$
$$= 4 + 2 + 0$$
$$= 6$$

Conversione: binario ➔ decimale

rappresentazione binaria	numero naturale (decimale)
000	**0**
001	**1**
010	**2**
011	**3**
100	**4**
101	**5**
110	**6**
111	**7**

basta applicare

$$n = \sum_{i=0}^{p-1} c_i \cdot B^i$$

Secondo problema

Conversione: decimale ➔ base B

Dato un numero X si cerca $c_N..c_0$ la sua rappresentazione in una base generica B

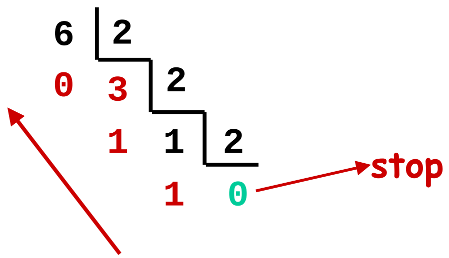
Conversione per divisione

Si calcolano i resti delle divisioni per la base **B**

- si divide X e i successivi quozienti per B fino a che non si ottiene quoziente 0
- il resto ottenuto nella divisione *i*-esima è la *i*-esima cifra (c_i) della rappresentazione in base B
- I resti, scritti in ordine inverso,corrispondono al numero in base B

Conversione: decimale ➔ binario

Si calcolano i resti delle divisioni per **2**



Divisione	Quoziente	Resto
6:2	3	0
3:2	1	1
1:2	0	1

$$\rho_2(6) = 110_2$$

Altro Esempio: base 2

B = 2

A = {0, 1}

Esempio

$$12_{10} = ?_2$$

Divisione	Quoziente	Resto
12:2	6	0
6:2	3	0
3:2	1	1
1:2	0	1

$12_{10} = 1100_2$

Servono **4** bit

Altre rappresentazioni usate

- Rappresentazione ottale
- Rappresentazione esadecimale

Utili per rappresentare sinteticamente i valori binari: la rappresentazione binaria è quella usata nei calcolatori, ma per noi umani è abbastanza scomoda da usare

perché numeri anche non grandissimi vengono rappresentati da stringhe di simboli piuttosto lunghe

Rappresentazione ottale

B = 8
A = {0,1, 2, 3, 4, 5, 6, 7}

Esempio

725_8

$= 7 \cdot 8^2 + 2 \cdot 8^1 + 5 \cdot 8^0$

$= 469_{10}$

Rappresentazione esadecimale

B = 16
A = {0,1, 2, 3, 4, 5, 6, 7, 9, A, B, C, D, E, F}

Esempio

$B7F_{16}$

$= 11 \cdot 16^2 + 7 \cdot 16^1 + 15 \cdot 16^0$

$= 2943_{10}$

Tabella riassuntiva

Decimale	Binario	Ottale	Esadecimale
0	0000	00	0
1	0001	01	1
2	0010	02	2
3	0011	03	3
4	0100	04	4
5	0101	05	5
6	0110	06	6
7	0111	07	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Esercizi

1. Dato il numero naturale decimale 44_{10} rappresentarlo in base **2**, **5**, **8** e **16**

2. Date le seguenti stringhe:

   $3BA_{16}$

   1110_2

   271_8

   Quali numeri naturali decimali rappresentano?

Soluzione: 1.

$44_{10} = 101100_2$

Divisione	Quoziente	Resto
44:2	22	0
22:2	11	0
11:2	5	1
5:2	2	1
2:2	1	0
1:2	0	1

Soluzione: 1.

$44_{10} = 134_5$

Divisione	Quoziente	Resto
44:5	8	4
8:5	1	3
1:5	0	1

Soluzione: 1.

$44_{10} = 54_8$

Divisione	Quoziente	Resto
44:8	5	4
5:8	0	5

$44_{10} = 2C_{16}$

Divisione	Quoziente	Resto
44:16	2	12 (C_{16})
2:16	0	2

Soluzione: 2.

$3BA_{16} = 3 \cdot 16^2 + 11 \cdot 16^1 + 10 =$ **954**

$1110_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 =$ **14**

$271_8 = 2 \cdot 8^2 + 7 \cdot 8^1 + 1 =$ **185**

Esercizio

- La seguente affermazione è vera?

$$1376_7 = 545_{10}$$

1376_7 non può essere una stringa in base 7

Casi particolari di conversione

B_1 ----------------> B_2

1. $B_1 = B_2^k$

Caso in cui la base di partenza è una potenza di ordine k della base di arrivo

1. $B_2 = B_1^k$

Caso in cui la base di arrivo è una potenza di ordine k della base di partenza

$B_1 = B_2^k$

B_1 ----------------> B_2

Bisogna rappresentare ogni cifra della rappresentazione in base B_1 nelle corrispondenti **k** cifre della rappresentazione in base B_2

$B_1 = 8; B_2 = 2$

$k = 3$ infatti $8 = 2^3$

Ogni cifra in base B_1 può essere rappresentata in base B_2 con 3 cifre

Esempio

72_8 ------> rappresentazione in base 2?

7 → 111

2_8 → 010

$111\ 010_2$

$B_2 = B_1^k$

B_1 ----------------> B_2

$B_1 = 2$
$k = 3$
$B_2 = 8$

Raggruppare a **k** a **k** le cifre della rappresentazione in base B_1 e convertirle in una sola cifra in base B_2

110111_2 ------> rappresentazione in base **8**?

110 111_2

6 7_8

… altro esempio

10111_2 ------> rappresentazione in base **8**?

010 111_2

2 7_8

aggiungere uno 0 per ogni cifra mancante a sinistra (cifre più significative)

Esercizi

- Si consideri il numero naturale **27** in base **10**.
- Determinare la sua rappresentazione nelle seguenti basi:

B = 2
B = 8
B = 16

Soluzione: 27 in base 2

Base 2

Divisione	Quoziente	Resto
27:2	13	1
13:2	6	1
6:2	3	0
3:2	1	1
1:2	0	1

11011_2

Soluzione: 27 in base 8

So che **27** in base **2** è: $\mathbf{11011_2}$

$\mathbf{8 = 2^3}$

Raggruppare a **3** a **3** le cifre della rappresentazione in base **2**

Base **8**

$\mathbf{011 \quad 011_2}$

$\mathbf{3 \qquad 3_8}$

aggiungere **0** a sinistra

$\mathbf{33_8}$

Soluzione: 27 in base 16

So che **27** in base **2** è: $\mathbf{11011_2}$

$\mathbf{16 = 2^4}$

Raggruppare a **4** a **4** le cifre della rappresentazione in base **2**

- Base **16**

$\mathbf{0001 \quad 1011_2}$

$\mathbf{1 \qquad B_{16}}$

aggiungere **0**

$\mathbf{1B_{16}}$

Esercizio

- Data la seguente stringa:

$\mathbf{37A_{16}}$

- Quali numero naturale (decimale) rappresenta e quale è la sua rappresentazione in base **2**?

Soluzione

$\mathbf{37A_{16} = 3 \cdot 16^2 + 7 \cdot 16^1 + 10 = 890_{10}}$

$\mathbf{16 = 2^4}$

Esplodere le cifre

Base **2**

$\mathbf{3 \qquad 7 \qquad A_{16}}$

$\mathbf{0011 \quad 0111 \quad 1010_2}$

$\mathbf{001101111010_2}$

Proprietà della rappresentazione posizionale

- Potenza ennesima: $\mathbf{B^n}$
- Moltiplicazione: $\mathbf{n \cdot B}$
- Moltiplicazione: $\mathbf{n \cdot B^s}$
- Divisone: $\mathbf{n/B}$
- Divisione: $\mathbf{n/B^s}$

Potenza ennesima: B^n

$$B^n = 1\underbrace{0 \ldots 0}_{\text{n volte 0}}$$

La potenza di B di ordine n è pari al numero rappresentato con la cifra B-1 per la cifra più significativa, seguita da n zeri

Esempi

$2^4 = 10000_2$

$10^3 = 1000_{10}$

Moltiplicazione: $n \cdot B$

- Occorre aggiungere una cifra in più ad n
  - scalare verso sinistra ed inserire uno zero nella posizione meno significativa

$$\rho(n) = c_{p-1}\ c_{p-2} \ldots c_1\ c_0$$

$$\rho(n \cdot B) = c_p\ c_{p-1}\ c_{p-2} \ldots c_1\ c_0\ 0$$

Esempi

B = 10	B = 2
p = 3	p = 3
$321_{10} \cdot 10_{10} = 3210_{10}$	$11_2 \cdot 10_2 = 110_2$

Divisione: n/B

- bisogna scartare la cifra meno significativa
  - scalare verso destra il numero n
  - inserire uno zero nella posizione più significativa

(n/B è rappresentabile con una cifra in meno rispetto ad n)

Esempio

1 0 ~~1~~

0 1 0

B = 10	B = 2
p = 3	p = 3
$345_{10} / 10_{10} = 034_{10}$	$101_2 / 10_2 = 010_2$

Moltiplicazione: n·B^s e Divisione: n/B^s

Moltiplicazione: **n·B^s**

Traslare di **s** passi la rappresentazione verso sinistra ed inserire **s** zeri da destra

Divisione: **n/B^s**

Traslare di **s** passi la rappresentazione verso destra ed inserire **s** zeri da sinistra

Aritmetica sui numeri naturali: base 2

Somma

- La somma di due numeri naturali espressi nel sistema binario viene eseguita con le stesse modalità del sistema decimale
- Si dispongono i due numeri in colonna e si sommano tra loro le cifre (bit) di ogni colonna, partendo dalla meno significativa (ovvero da destra), e ricordando quanto vale la somma e il riporto di ciascuna coppia di cifre binarie:

0	0	0	1
0 +	1 +	0 +	1 +
0 =	0 =	1 =	1 =
0	1	1	1

Somma: 01101 + 01011 = ?

B = 2

p = 5

I_{rapp} = [0 .. 31]

```
0 1 1 1 1
  0 1 1 0 1 +
  0 1 0 1 1 =
  ---------
  1 1 0 0 0
```

decimale

```
13+
11=
---
24
```

r = Riporto precedente; s = somma; r' = riporto successivo

a	b	r	s	r'
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Overflow

-Tipicamente in un computer il numero di cifre disponibili per rappresentare un numero è fissato

- In generale per rappresentare la somma di due numeri naturali di lunghezza *k* possono essere necessarie *k+1* cifre (ad esempio la somma di due numeri in base dieci ciascuno di una cifra può richiedere due cifre come nel caso seguente: 7+9=16)

- Se in un computer sono disponibili solo *k* cifre per memorizzare un numero, si genera una condizione di *overflow* (o trabocco), che significa trabocco

- In questo caso le *k* cifre disponibili per rappresentare il risultato dell'operazione, rappresentano un numero che non corrisponde alla somma dei due numeri dati, ma solo alle sue prime *k* cifre

Overflow

B = 2

p = 3

I_{rapp} = [0..7]

```
1 0 1
  1 0 1 +
  1 0 1 =
  -------
  0 1 0
```

decimale

```
 5 +
 5 =
----
1 0
```

Overflow

$r_p = 1$

Sottrazione

- La sottrazione di due numeri naturali rappresentati in base due viene eseguita (analogamente a quanto avviene in base dieci) incolonnando i numeri e sottraendo tra loro le cifre in ogni colonna, partendo dalla meno significativa (ovvero quella più a destra), e ricordando quanto vale la sottrazione ed il prestito di ciascuna coppia di cifre binarie

```
 0        0        1        0
  0 -      1 -      0 -      1 -
  0 =      0 =      1 =      1 =
  ---      ---      ---      ---
  0        1        1        0
```


Sottrazione

Esempi

p = prestito precedente

s = sottrazione

P' = prestito successivo

```
  0 1 1     ← prestiti
1 1 0 0 -
0 0 1 1 =
---------
1 0 0 1
```

```
0 1 1 1 1
  1 0 1 0 0 -
  0 0 1 1 1 =
  -----------
  0 1 1 0 1
```

a	b	p	s	p'
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Moltiplicazione

moltiplicando ← m * n → moltiplicatore

• La moltiplicazione di due numeri naturali rappresentati in base due viene eseguita incolonnando i numeri e utilizzando la stessa tecnica utilizzata per i numeri in base dieci:

- si moltiplica il moltiplicando per ogni cifra del moltiplicatore traslando a sinistra ogni risultato di tanti posti quanto è il numero d'ordine della cifra; si sommano i risultati intermedi

- Si ricorda inoltre che, se i due numeri hanno una sola cifra, i quattro casi possibili sono:

```
0 x      1 x      0 x      1 x
0 =      0 =      1 =      1 =
---      ---      ---      ---
0        0        0        1
```

Moltiplicazione

Moltiplicando: **k** bit

Moltiplicatore : **k** bit

Risultato: **kx2** bit

Esempio

```
    1  0  *
    1  1  =
   --------
    1  0
 1  0
 ----------
 1  1  0
```

- Tipicamente in un computer il numero di cifre disponibili per rappresentare un numero è fissato
- In generale per rappresentare la moltiplicazione di due numeri naturali di lunghezza *k* possono essere necessarie *kx2* cifre (ad esempio la moltiplicazione di due numeri in base dieci, ciascuno di una cifra, può richiedere due cifre come nel caso seguente: 3x4=12)
- Per evitare overflow, la moltiplicazione di due numeri a k cifre, occorre quindi che il risultato sia memorizzato con kx2 cifre

Esempio

```
           1 1 1 *
           1 1 1 =
        1 1 1
         ---------
           1 1 1
         1 1 1
     1 1 1
       -----------
       1 0 1 0 1
       1 1 1
     -------------
     1 1 0 0 0 1
```


Divisione

```
 dividendo        divisore
        ↖        ↗
          n | m
```

Algoritmo che siamo soliti usare nella base decimale:

• vedendo se e quante volte il divisore "sta" in una porzione del dividendo o in un risultato intermedio (poiché nel sistema binario le cifre possono essere soltanto 0 oppure 1, il divisore o non è contenuto nel dividendo parziale, o lo è solo una volta).

Dividendo: su **2xp** bit

Divisore, quoziente e resto: su **p** bit

Esempio

```
1  0  1  0 | 1  1
   1  1    | 1  …
--------
…  1  0  …
      …  …
-----------
   …  …  …
```

Si nota che il divisore 11 è contenuto 1 volta nel dividendo parziale 101 e che la differenza parziale è 10. Si abbassa poi la cifra successiva 0 e si procede.

Esempio

```
1 0 1 0 | 1 1
  1 1     1 1
0 1 0 0
    1 1
  0 0 1
```

Nel sistema decimale si ha:

```
10 | 3
 1   3
```

Si nota inizialmente che il divisore 11 è contenuto 1 volta nel dividendo parziale 101 e che la differenza parziale è 10
Si abbassa poi la cifra successiva 0 e si ottiene il numero 100 in cui il divisore 11 è contenuto ancora 1 volta, con resto 1

Esempio

```
1 1 1 1 1 1 0 1 | 1 0 1 0
1 0 1 0           1 1 0 0 1
0 1 0 1 1
  1 0 1 0
  0 0 0 1 1 0 1
        1 0 1 0
        0 0 1 1
```

Dividendo: su 8 bit

Divisore e resto: su 4 bit

Esercizio 1

Siano date le seguenti sequenze di bit di **8** cifre (bit):

A = 10000100
B = 11001100

Eseguire, in aritmetica binaria, le seguenti operazioni:

A+B
B-A

Soluzione: A+B

A = 10000100
B = 11001100

overflow

```
1         1 1
  1 0 0 0 0 1 0 0 +
  1 1 0 0 1 1 0 0 =
1 0 1 0 1 0 0 0 0
```

Soluzione: B-A

A = 10000100
B = 11001100

```
  1 1 0 0 1 1 0 0 -
  1 0 0 0 0 1 0 0 =
 ------------------
  0 1 0 0 1 0 0 0
```


Esercizio 3

Siano date le seguenti sequenze di bit di 4 bit:

A = 1001
B = 1101

Eseguire, in aritmetica binaria, le seguenti operazioni:

A*B	risultato su 8 bit
A/11	risultato su 2 bit

Soluzione: A*B

```
        1 0 0 1 *
        1 1 0 1 =
       ----------
        1 0 0 1
      0 0 0 0
     ------------
      0 1 0 0 1
    1 0 0 1
   --------------
    1 0 1 1 0 1
  1 0 0 1
 ----------------
1 1 1 0 1 0 1
```


Soluzione: A/11

```
1 0 0 1   | 1 1
  1 1     |-----
-----       1 1
0 0 1 1
    1 1
  -----
    0 0
```