Sistemi di numerazione

Definizione

Chiamiamo sistema di numerazione un insieme di *simboli* e di *regole* che permettono di rappresentare i numeri.

Per definire un sistema di numerazione è necessario indicare:

- un insieme di simboli dette le cifre
- una sintassi, cioè un insieme di regole che specificano come costruire i vari numeri.

Esempio

Nella numerazione romana alcuni dei simboli sono: I, V, X, L, C, ...
Nella numerazione decimale simboli sono 10: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Oltre alla differenza dei simboli è però altrettanto importante la differenza delle regole secondo le quali si scrivono i numeri (sintassi).

Per renderci conto di questo consideriamo i seguenti numeri romani

XVI	sappiamo tutti che equivale al nostro 16
III	di nuovo sappiamo che equivale al nostro 3

se noi però pensassimo di tradurre semplicemente simbolo per simbolo in cifre arabe otterremmo

X V I	I I I
10 5 1	**1 1 1**

con un valore numerico completamente diverso. Come mai?

Entra cioè in gioco la differenza di sintassi.

Senza insistere osserviamo che

- il sistema di numerazione romano è **additivo:**

  XVI=10+5+1=16

- Il nostro sistema di numerazione è **posizionale:**
  il valore della cifra dipende cioè dalla posizione che occupa:

  $111= 1*10^2+1*10^1+1*10^0=1*10^2+1*10+1$

I SISTEMI DI NUMERAZIONE

Un sistema di numerazione è un insieme finito di simboli a cui vengono applicate delle regole.
Vi sono due categorie di sistemi di numerazione, quelli posizionali, quali il sistema decimale, binario, ottale ed esadecimale, che saranno oggetto di studio di questo modulo, e quelli non posizionali, quali il sistema di numerazione romano.
Ognuno di questi sistemi adotta i suoi simboli e le sue regole sia per rappresentare un numero che per effettuare le operazioni. Ogni sistema posizionale prende il nome dal numero di simboli che utilizza per rappresentare i numeri.
I sistemi posizionali la cui base è un multiplo di 2 godono dell'interessante proprietà di poter effettuare in modo estremamente semplice una conversione da un sistema all'altro.

1. SISTEMI POSIZIONALI

Esistono diversi sistemi di numerazione, quali il sistema di numerazione romano, il sistema decimale, binario, ottale, esadecimale e via dicendo.

Qualunque sistema di numerazione è composto da un insieme finito di simboli e da un insieme di regole.

Nei sistemi di numerazione posizionali, il numero di simboli di cui dispone il sistema stesso prende il nome di **base** del sistema. Pertanto se **B** è la base del sistema di numerazione l'insieme dei simboli del sistema stesso sarà sempre {0, ..., B-1}

Esempio 1.a:

La base del sistema decimale è composta dai seguenti 10 simboli: 0,1,2,3,4,5,6,7,8,9 mentre la base del sistema binario è composto solo da 2 simboli: 0,1.

Le **regole** permettono di associare ad una sequenza di cifre un valore numerico e di effettuare le operazioni. Diversamente dal sistema romano, gli altri sistemi di numerazione citati sono sistemi posizionali, cioè ciascuna cifra ha un peso diverso a seconda della posizione che occupa nel numero.

Esempio 1.b:

La cifra 7 vale 70 nel numero 170, ma vale 7 nel numero 117.

Nei sistemi posizionali ogni numero può essere rappresentato come segue:

$$\sum_{k=0}^{n-1} C_k * B^k = C_{n-1} * B^{n-1} + C_{n-2} * B^{n-2} + \ldots + C_2 * B^2 + C_1 * B^1 + C_0 * B^0$$

dove $\mathbf{C_n}$ rappresenta la cifra più significativa, $\mathbf{C_0}$ la cifra meno significativa e **B** la base del sistema.

Quando può generarsi confusione, a pedice del numero viene messa la base.

Esempio 1.c:

$129_{10} = 1*10^2 + 2*10^1 + 8*1^0$
$129_{16} = 1*16^2 + 2*16^1 + 8*16^0$
$10_2 = 1*2^1 + 0*0^0$

2. SISTEMA DI NUMERAZIONE BINARIO

Il sistema di numerazione binario è il sistema utilizzato dagli elaboratori automatici. *Le cifre del sistema binario sono solo due:0 ed 1 e prendono anche il nome di bit*. Nei calcolatori il bit rappresenta la più piccola unità di informazione.

Vediamo di seguito come effettuare la conversione da un numero binario ad un numero decimale e viceversa.

2.1. CONVERSIONE DAL SISTEMA DECIMALE AL SISTEMA BINARIO

La conversione da decimale a binario è basata sull'uso delle potenze del 2, che sono rappresentate in Figura 2.1.a.

POTENZA	VALORE
2^0	1
2^1	2
2^2	4
2^3	8
2^4	16
2^5	31
2^6	64
2^7	128
2^8	256
⋮	⋮

Figura 2.1.a: Potenze del sistema binario

Per effettuare la conversione si procede come segue:
- si parte sempre dalla potenza del vicina al numero e si va indietro verso le potenze meno significative, ottenendo così una sequenza di potenze del 2.
- dove manca la potenza si mette 0 dove c'è la potenza si mette 1.

Esempio 2.1.a:

$89_2 = 64 + 16 + 8 + 1 = 1011001$
$36_2 = 32 + 4 = 100100$

Un'altra metodologia per convertire un numero decimale in binario, è quella *per divisioni successive*, che consiste nel fare la divisione per due e prendere il resto. I resti, man mano che si ottengono, devono essere allineati da destra verso sinistra. La divisione termina quando il dividendo è uguale a zero.

Esempio 2.1.b:

Dato il numero 25, trasformarlo in binario.
Effettuando le divisioni si ottiene il numero binario 11001.
25:2=12 resto 1
12:2=6 resto 0
6:2=3 resto 0
3:2=1 resto 1
1:2=0 resto 1

Dagli esempi si evidenzia che il numero di bit necessario cresce all'aumentare della grandezza del numero.
Normalmente però si ragiona con un numero fisso di bit, tipicamente 8, 16, 32.
Tali unità si chiamano rispettivamente: BYTE(8 bit), WORD(16 bit), DWORD(32 bit).

2.2. CONVERSIONE DAL SISTEMA BINARIO AL SISTEMA DECIMALE

La conversione di un numero da binario a decimale si ottiene con la formula già vista:

$$\sum_{k=0}^{n-1} C_k * B^k = C_{n-1} * B^{n-1} + C_{n-2} * B^{n-2} + \ldots + C_2 * B^2 + C_1 * B^1 + C_0 * B^0$$

Nel caso specifico del sistema binario **B** assume sempre il valore **2**.

Esempio 2.2.a:

$101011_2 = 1*2^5+0*2^4+1*2^3+0*2^2+1*2^1+1*2^0= 32+8+2+1 = 43_{10}$

SISTEMA DI NUMERAZIONE ESADECIMALE

Nel sistema esadecimale vi sono 16 simboli. Oltre alle cifre 0…9 del sistema decimale si utilizzano i simboli A (10), B (11), C(12), D(13), E(14), F(15).

Il sistema è ancora posizionale, la conversione da base 16 a base 10 e viceversa risulta pertanto simile alla base 2.

Esempio 4.a:

Dato il numero $3AC_{16}$ convertirlo in decimale.

$3AC_{16} = 3*16^2+10*16^1+12*16^0 = 768+160+12 = 940_{10}$

Dato il numero 28_{10} convertirlo in esadecimale.

28:16=1 resto C
1:16=0 resto 1

Pertanto $28_{10} = 1C_{16}$

CONVERSIONE BINARIO ESADECIMALE

Per convertire un numero da binario a esadecimale si parte da destra e si *raggruppano le cifre in insiemi di 4 bit*, quindi si associa ad ogni gruppo il corrispondente valore esadecimale.

Esempio

Convertire in esadecimale il numero binario 1110101100.

0011	1010	1100
3	A	C

Pertanto $1110101100_2 = 3AC_{16}$

Viceversa per convertire un numero da esadecimale a binario si considera ogni singola cifra esadecimale, e si associa l'equivalente numero binario su 4 bit.

Un codice decimale si dice pesato se la relazione che intercorre tra la cifra decimale D da rappresentare e l'insieme delle cifre binarie ad esso associato è del tipo:

$$D = \sum_{i=0}^{m-1} b_i \cdot p_i$$

dove: *- m : numero di bit del codice*
- bi : generico bit del codice
- pi : peso corrispondente.

Il codice decimale ***BCD*** *(Binary Coded Decimal) è un codice pesato con pesi (8 - 4 - 2 - 1).*

LE OPERAZIONI

Nei sistemi posizionali le operazioni seguono tutte le medesime regole:

- se sommando due cifre si ottiene un numero maggiore o uguale alla base, si ha un riporto o carry.
- 2)se sottraendo due cifre, il minuendo è minore del sottraendo si fa un prestito dalla cifra di ordine superiore.

Esempi

Di seguito sono mostrate le operazioni di addizione, sottrazione, moltiplicazione e divisione, nel sistema di numerazione binario:

Addizione

1 1 1 1 1 riporti

100111_2+

$111101_2=$

1100100_2

Sottrazione

0 1 1 0 prestito

100111_2-

$11101_2=$

1010_2

Moltiplicazione

1011_2*

$101_2=$

1011

000

1011

110111_2

Divisione

```
110111 | 101
101      1011
==111
  101
   101
   101
   ===
```

Quando si eseguono le operazioni in binario occorre tenere presente che:

- dopo la cifra **1**, il sistema prosegue con i numeri **10, 11, 100, 101...** e via dicendo;
- 1+1=0 con riporto di 1;
- 0-1=1 con prestito di 1 dall'unità di ordine superiore.

Esempi

Di seguito sono mostrate le operazioni di addizione e sottrazione nel sistema di numerazione esadecimale:

Addizione

1 riporto

$171_{16}+$

$99_{16}=$

$20A_{16}$

Sottrazione

1 prestito

$55_{16}-$

$19_{16}=$

$3C_{16}$

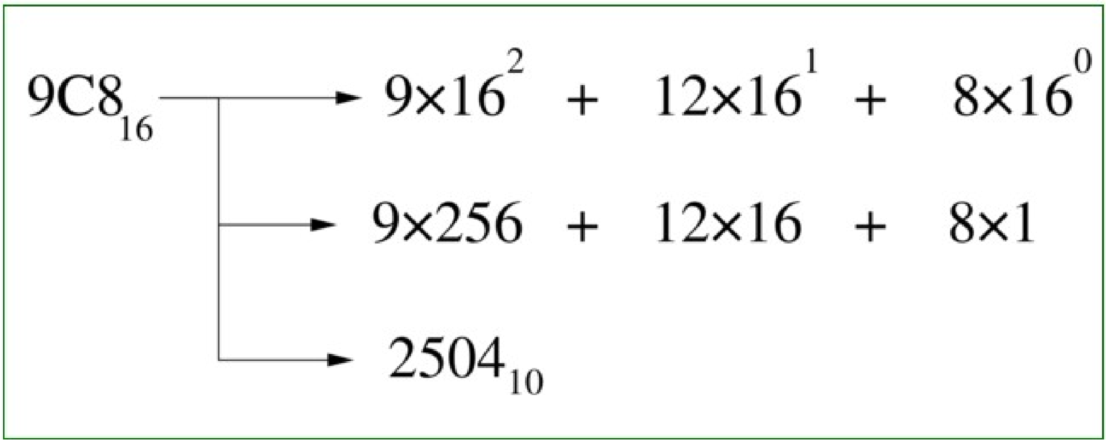
La storia
http://www.homolaicus.com/scienza/calcolo/sis_num.htm

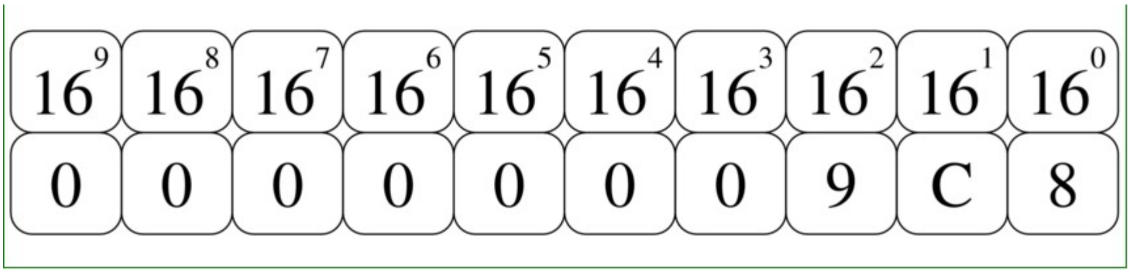
Notazione posizionale
http://www.diit.unict.it/users/michele/didattica/fondamenti/fondamenti/sisteminum.html

- Metodo divisioni successive
  http://www.diit.unict.it/users/michele/didattica/fondamenti/fondamenti/conv.html
- Codici (**BCD**, ASCII …)
  http://www.diit.unict.it/users/michele/didattica/fondamenti/fondamenti/caratteri.html

Sistemi di numerazione: scomposizione di numero posizionale
http://infocom.uniroma1.it/alef/AL/a217.htm

Figura 10.17. Esempio di scomposizione di un numero in base sedici.





Sistema di numerazione

http://it.wikipedia.org/wiki/Sistema_di_numerazione

Definizione
http://www.itg-rondani.it/dida/Matem/ipermonica/numeri/sist_num/sistemi.htm

LA RAPPRESENTAZIONE DEI NUMERI

La rappresentazione dei numeri in un sistema di elaborazione dipende dall'insieme numerico che si considera. L'insieme dei numeri naturali presenta infatti problematiche diverse rispetto all'insieme dei numeri interi, reali o complessi. La rappresentazione dei numeri naturali può infatti essere effettuata in BCD, oppure se si considera tale insieme come un sottoinsieme dei numeri interi, allora segue le regole di rappresentazione del suddetto insieme. I numeri interi possono essere rappresentati in modulo e segno, in complemento ad uno ed in complemento a due.

I numeri reali seguono ancora altre regole, e la loro rappresentazione può essere effettuata in Virgola Fissa o in Virgola Mobile. Una regola comune rispetto ad un sistema di elaborazione è che quest'ultimo può accettare solo numeri a precisione finita, dove cioè il massimo numero delle cifre dipende dal tipo di elaboratore.
Poiché l'elaboratore lavora su cifre binarie, la lunghezza dei bit destinati ad ospitare un numero, definisce anche la lunghezza della parola di memoria.

1. LA RAPPRESENTAZIONE DEI NUMERI

Se si considera l'insieme N dei numeri naturali, questo è un insieme infinito di numeri senza segno e pertanto la rappresentazione nel sistema di elaborazione dati dipende dal numero di bit che viene riservato per rappresentare il numero.

Supponiamo che il numero di bit utilizzato sia n ed M sia il numero delle possibili rappresentazioni.
Se n = 8 allora $\mathbf{M} = \mathbf{2^n} = 2^8 = 256$
Cioè con 8 bit si possono rappresentare 256 numeri naturali e quindi tutti i numeri decimali da 0 a 255, come mostrato di seguito.

DECIMALE	BINARIO	ESADECIMALE
0	00000000	00
255	11111111	FF

Quindi il campo dei naturali su 8 bit in binario è: 00000000...11111111, mentre il campo dei naturali su 8 bit in esadecimale è: 00H,......FFH.

Se n=16 allora $M = 2^n = 2^{16} = 65536 = 64K$

Cioè con 16 bit si possono rappresentare 65536 numeri naturali e quindi tutti i numeri decimali da 0 a 65535, come mostrato di seguito.

DECIMALE	BINARIO	ESADECIMALE
0	0000 0000 0000 0000	0000H
65535	1111 1111 1111 1111	FFFFH

Quando si opera con i numeri naturali su n bit, è possibile durante le operazioni di somma, uscire dal campo di rappresentazione. Il fenomeno si chiama carry o riporto.

Esempio 1.a:

Supponiamo di voler sommare i numeri 171 e 99. Entrambi possono essere rappresentati con 8 bit. Ma effettuando la somma si ottiene un numero maggiore di 255 e questo produce un carry. 8 bit non sono quindi sufficienti a rappresentare completamente il risultato; il risultato richiede 9 bit, pertanto per uniformarsi alle strutture di memoria, è necessario utilizzare 16 bit.

```
               11   11
  171+         10101011+
   99=         01100011=
  ----        ----------
  270         [1]00001110
               |
             Carry
```

Con la divisione invece si possono generare numeri che non esistono nel campo di rappresentazione.

Esempio 1.b:

Si consideri la divisione tra i due numeri naturali 18 e 7. Il risultato è 2,5714..., che chiaramente non è un numero naturale.

Ne segue che se si definisce il range dei numeri naturali rappresentabili, tutti i numeri al di fuori di esso (numeri interi negativi, numeri reali e numeri complessi), produrranno un errore.

Non solo, essendo l'insieme limitato gode della proprietà di chiusura sulle operazioni di addizione, sottrazione e moltiplicazione.

Tali proprietà potrebbero essere violate quando il risultato è maggiore del più grande elemento che il calcolatore può trattare(**overflow**) o se il risultato è minore del più piccolo elemento(**underflow**). Come visto poi per la divisione il risultato potrebbe non appartenente all'insieme.

Esempio 1.c:

Si supponga che nell'elaboratore su cui stiamo lavorando si sia dichiarato che i numeri naturali occupano 8 bit.

Sappiamo allora che il più piccolo numero rappresentabile è 0 ed il più grande è 255.

Pertanto effettuando seguente somma:
128+200= 328 si ha un overflow.
Mentre effettuando la seguente sottrazione:
128-200= -72 si ha un underflow.

2. LA RAPPRESENTAZIONE DEI NUMERI NATURALI IN BCD

La rappresentazione **Binary Coded Decimal**(Decimale Codificato in Binario), permette di rappresentare i numeri naturali senza effettuare precedentemente la conversione binaria del numero.

La conversione da decimale a binario avviene di fatto rappresentando ogni cifra come una sequenza di quattro bit. Pertanto poiché su 4 bit si hanno 16 diverse rappresentazioni, ma in realtà se ne utilizzano solo 10, ne segue che dato un numero decimale con n cifre, occorrono n*4 bit per rappresentarlo in binario. Di conseguenza, l'intervallo di valori decimali rappresentabili su n bit è dato da:

0<= x <= + 9*1...1

Il numero di 1 che rientrano nella moltiplicazione è pari al numero di bit che si hanno nella rappresentazione diviso per 4.

Esempio 2.a:

Rappresentare in BCD il numero 2345.

Le cifre 2,3,4,5 si rappresentano in binario su 4 bit come segue:
2=0010
3=0011
4=0100
5=0101

Pertanto il numero
2345_{10} =$0010\ 0011\ 0100\ 0101_2$

L'intervallo di valori decimali rappresentabili su 16 bit è dato da:
0_{10}<= x <= + 9*1111=9999

Inoltre, sempre dovuto al fatto che su 4 bit si hanno 16 diverse rappresentazioni, la somma dei valori rappresentati in BCD non risulta corretta, quando si sommano due cifre il cui valore risulta maggiore di 9, in quanto tale rappresentazione non appartiene ovviamente al campo di rappresentazione.

Per risolvere questo problema si usa il valore correttivo 0110_2, che viene sommato solo ai gruppi di 4 bit che hanno provocato un riporto, o il cui valore è maggiore di 1001, cioè più grande del massimo valore rappresentabile.

Esempio 2.b:

Sommare, utilizzando la rappresentazione BCD, il numero 2945.

```
       1
  0010 1001 0100 0101+
  0100 1000 1000 1000=
  0111 0001 1100 1101

        Fuori
        configurazione

    Produce
  un riporto

Fatta la somma, si aggiunge il valore
correttivo:

            1111 1   1 1
       0111 0001 1100 1101+
            0110 0110 0110=
       0111 1000 0011 0011
```

La Storia
http://www.tecnoteca.it/museo/01

Rappresentazione dei numeri

http://it.wikipedia.org/wiki/Rappresentazione_dei_numeri_relativi

- Complemento a uno http://it.wikipedia.org/wiki/Complemento_a_uno
- Complemento a due http://it.wikipedia.org/wiki/Complemento_a_due

Rappresentazione dei numeri naturali rispetto a una base e operazioni aritmetiche

http://progettomatematica.dm.unibo.it/Basi/basi.html

Dispense **Prof. Franco Zambonelli**
http://polaris.ing.unimo.it/didattica/corsore/LucidiPDF/Numeri.pdf