Lezione 6

Notazione posizionale

- Ci sono solo 10 tipi di persone al mondo:
  - quelle che conoscono la rappresentazione dei numeri in base 2, e
  - quelle che non la conoscono ...

Base 2

- Per capire fino in fondo come sono rappresentate le informazioni in un calcolatore occorre conoscere la rappresentazione dei numeri in <u>base 2</u>
- Il motivo è che le informazioni sono rappresentate come sequenze di bit, ossia cifre con due soli possibili valori

Basi e cifre 1/2

- Partiamo dalla rappresentazione di un numero in una generica base
- Cominciamo dalla rappresentazione dei numeri naturali

Basi e cifre 2/2

- Rappresentazione di un numero in una data base: <u>sequenza di cifre</u>
- *Cifra*: simbolo rappresentante un numero
- *Base*: numero (naturale) di valori possibili per ciascuna cifra
- In base $b > 0$ si utilizzano b cifre distinte, per rappresentare i valori $0,\ 1,\ 1 + 1,\ 1 + 1 + 1,\ \ldots,\ b - 1$

Cifre e numeri in base 10

- Es: in base 10 le cifre sono

0 che rappresenta il valore 0

1 che rappresenta il valore 1

2 che rappresenta il valore 1+1

3 che rappresenta il valore 1+1+1

Simbolo grafico

Concetto astratto di numero naturale

.

.

.

9 che rappresenta il valore 1+1+1+1+1+1+1+1+1

Notazione posizionale

- Rappresentazione di un numero su *n* cifre in base *b*:

Posizioni

$$a_{n-1}\ a_{n-2}\ a_{n-3}\ \ldots\ a_1 a_0$$

$$a_i\ \varepsilon\ \{0, 1, \ldots, b - 1\}$$

- Es: Notazione decimale:
  $b = 10$, $a_i\ \varepsilon\ \{0, 1, 2, \ldots, 9\}$
  $345 \Rightarrow a_2 = 3, a_1 = 4, a_0 = 5$

Notazione

- Per rendere esplicita la base utilizzata, si può utilizzare la notazione

$$[\ x\ ]_b$$

$$a_i \in \{0, 1, \ldots , b - 1\}$$

dove *x* è una qualsiasi espressione, ed il cui significato è che ogni numero presente nell'espressione è rappresentato in base *b*

Esempi in base 10

$[\,345\,]_{10}$

$[\,2 * 10 + 5 * 1\,]_{10}$

Notazione posizionale

$$[\ a_{n-1} a_{n-2} a_{n-3} \dots a_1 a_0\ ]_b =$$

$$[a_0 * 1 + a_1 * b + a_2 * b^2 + a_3 * b^3 + \dots + a_{n-1} * b^{n-1}\ ]_b$$

$$= [\ \sum_{i=0,\ 1,\ \dots,\ n-1} a_i * b^i\ ]_b$$ *Peso cifra i-esima*

- Es: $b = 10$, $a_i\ \varepsilon\ \{0, 1, 2, \dots, 9\}$

$$[\ 345\ ]_{10} = [3 * 10^2 + 4 * 10 + 5 * 1]_{10}$$

"yo cuento como un cero a la izquierda"
… io conto come uno zero a sinistra

Calcoli

- Si utilizzano degli algoritmi
- Esattamente quelli imparati alle elementari per la base 10
- Esempio: per sommare due numeri, si sommano le cifre a partire da destra e si utilizza il riporto

Notazione binaria

- Base 2, 2 cifre:
  - 0, 1
- La cifra nella posizione *i-esima* ha peso 2^i
- Esempi (*configurazioni di bit*):

$[0]_{10} = [0]_2$

$[1]_{10} = [1]_2$

$[2]_{10} = [10]_2 = [1*2 + 0*1]_{10}$

$[3]_{10} = [11]_2 = [1*2 + 1*1]_{10}$

Base 16

- Una base che risulta spesso molto conveniente è la base 16
- Vediamo prima di cosa si tratta, e poi come mai è molto utilizzata

Notazione esadecimale

- Base 16, 16 cifre:
  - 0, 1, 2, ..., 9, A, B, C, D, E, F
- Valore cifre in decimale:
  - 0, 1, 2, ..., 9, 10, 11, 12, 13, 14, 15
- La cifra nella posizione *i-esima* ha peso 16^i
- Esempi:

$[0]_{10} = [0]_{16}$

$[10]_{10} = [A]_{16}$

$[18]_{10} = [12]_{16} = [1*16 + 2*1]_{10}$

Motivazione Base 16 1/3

- Ogni cifra in base sedici corrisponde ad una delle possibili combinazioni di 4 cifre in base 2
- Quindi, data la rappresentazione in base 2 di un numero naturale, la sua rappre-sentazione in base 16 si ottiene dividendo la sequenza in base in sotto-sequenze consecutive da 4 cifre ciascuna, partendo da destra, e convertendo ciascuna sotto-sequenza di quattro cifre binarie nella corrispondente cifra in base 16

Motivazione Base 16 2/3

Esempio: Dato il numero
$[1000001111]_2$

Dividiamo le cifre in gruppi da quattro da destra:
10 0000 1111
ed aggiungiamo due zeri all'inizio (senza modificare il valore del numero):
0010 0000 1111

In base 16 otteniamo:
2 0 F

Motivazione Base 16 3/3

- Viceversa, data la rappresentazione in base 16 di un numero naturale, il corrispondente numero in base 2 si ottiene convertendo semplicemente ciascuna cifra della rappresentazione in base 16 nella corrispondente sequenza di 4 cifre in base 2
- Invertendo il precedente esempio:
$[20F]_{16} = [1000001111]_2$

Rappresentazione naturali

- In una cella di memoria o in una sequenza di celle di memoria si può memorizzare con facilità un numero naturale memorizzando la configurazione di bit corrispondente alla sua rappresentazione in base 2
- Questa è la tipica modalità con cui sono memorizzati i numeri naturali
- Coincide con gli esempi che abbiamo già visto in lezioni precedenti

Rappresentazione interi 1/2

- Come rappresentare però numeri con segno?
- Non esiste un elemento all'interno delle celle, che sia destinato a memorizzare il segno
- Come potremmo cavarcela?

Rappresentazione interi 2/2

- Un'idea sarebbe quella di utilizzare uno dei bit per il segno
  - 0 per i valori positivi
  - 1 per i valori negativi
- Il problema è che <u>sprechiamo una configurazione di bit</u>, perché avremmo <u>due</u> diverse <u>rappresentazioni per</u> il numero <u>0</u>
  - Una col segno positivo
  - Una col segno negativo

Idea

- Rappresentare i numeri positivi semplicemente in base 2
- Non rappresentare i numeri negativi direttamente, ma sommargli prima una costante, che fa si che diventino positivi
- Il trucco starà nel far sì che i **veri** numeri positivi cadano in un intervallo di valori diverso da quello in cui cadono i **falsi** numeri positivi (ossia quelli ottenuti sommando una costante)
- Questa idea è alla base della **rappresentazione in complemento a 2**

Complemento a 2 1/2

- Se *n* è un numero maggiore di 0, si memorizza la sua rappresentazione in base 2
- Se *n* è un numero minore di 0, allora, anziché memorizzare il numero originale *n*, si memorizza, in base 2, il numero naturale risultante dalla somma algebrica
  $2^N + n$
  dove *N* è il numero di bit su cui si intende memorizzare il numero *n*

Complemento a 2 2/2

- Il vincolo da rispettare, affinché si possa correttamente rappresentare un numero n negativo su N bit, è che il risultato della somma
  $2^N + n$
  - deve essere un numero positivo
  - deve essere rappresentabile su N bit
- Facendo i conti, si ottiene che si possono rappresentare generici numeri n (positivi o negativi) contenuti nell'intervallo

  $[-2^{N-1}, 2^{N-1}-1]$

Nota 1/2

- Tralasciando dimostrazioni, nella rappresentazione in complemento a 2
  - una sequenza di *N* bit che rappresenta un numero intero negativo ha sempre il bit più a sinistra (più significativo) uguale ad 1
  - la rappresentazione di un numero naturale su *N* bit è uguale alla sua rappresentazione in base 2

Nota 2/2

- Quindi una configurazione di *N* bit con il bit più a sinistra ad 1 rappresenta
  - un valore positivo se sta rappresentando un numero naturale in base 2
  - un valore negativo se sta rappresentando un numero in complemento a 2

Vantaggi del complemento a 2

- C'è una sola rappresentazione per lo 0
- Gli algoritmi di calcolo delle operazioni di somma, sottrazione, moltiplicazione e divisione sono gli stessi dei numeri naturali rappresentati in base 2

Rappresentazione **int**

- Gli oggetti di tipo **int** sono tipicamente rappresentati in complemento a 2

- Adesso dovrebbe esservi più chiaro perché è vero che:

“Ci sono solo 10 tipi di persone al mondo: quelle che conoscono la rappresentazione dei numeri in base 2, e quelle che non la conoscono”

Esercizi

- Completare la quinta esercitazione