Algoritmi Numerici (Parte I)

[Lezione 2] Il complemento a due (Numeri interi)

Alessandro Antonucci
alessandro.antonucci@supsi.ch

https://colab.research.google.com/drive/1651n86ZQ3RASkAz6sgNR8D66w5L4Gax8

I Numeri Naturali e l'Addizione

- Numero $=$ quantità elementi in un insieme (cardinalità)
- Es. card(studenti I1A) = 23, card(docenti I1A) = 7
- Numeri cardinali detti naturali (`unsigned int`)
- Proprietà ordinale $\mathbb{N} = \{\mathbf{0}, \mathbf{1}, \mathbf{2}, \cdots, \mathbf{31}, \ldots\}$
- Addizione: cardinalità dell'insieme unione

  Es. card(docenti $\cup$ studenti I1A) = 38 =: 31 + 7
- $\mathbb{N}$ (cardinalità infinita) chiuso rispetto all'addizione

  la somma di due numeri naturali è un numero naturale

$$\boldsymbol{a} \in \mathbb{N}, \boldsymbol{b} \in \mathbb{N} \Rightarrow \mathrm{SOMMA}(\boldsymbol{a}, \boldsymbol{b}) \in \mathbb{N}$$

$$\mathrm{SOMMA}(\mathbf{31}, \mathbf{7}) = \mathbf{38}$$

I Numeri Interi e la Sottrazione

- Sottrazione: operazione inversa rispetto all'addizione

  Se $b = \mathrm{SOMMA}(a, c)$ e noto c, trovare a

- I1A totale 38, 7 docenti, quanti studenti?

  $\mathrm{SOTTRAZIONE}(38, 7) = 31$

- $a \in \mathbb{N}$, $b \in \mathbb{N} \Rightarrow \mathrm{SOTTRAZIONE}(a, b) \in \mathbb{N}$ solo se $a \geq b$

- Se $a < b$: $\mathrm{SOTTRAZIONE}(a, b)$ è numero negativo e

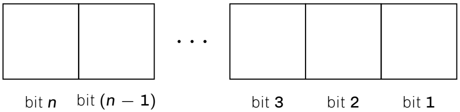
  $\mathrm{SOTTRAZIONE}(a, b) := -\mathrm{SOTTRAZIONE}(b, a)$

  $\mathrm{SOTTRAZIONE}(5, 7) = -\mathrm{SOTTRAZIONE}(7, 5) = -2$

- Unione naturali e negativi è l'insieme numeri interi $\mathbb{Z}$

- $\mathbb{Z} = \{\ldots, -1, 0, +1, +2, \ldots\}$ (cardinalità infinita come $\mathbb{N}$)

Memorie a n bit



- Singolo bit (**b**inary dig**it**) solo valori 0 o 1
- Memoria a n bit? 2^n configurazioni
- Rappr. naturali? Sistema posizionale
  Con n bit, range $\{0, 1, \ldots, 2^n - 1\} \subset \mathbb{N}$
- Rappresentare gli interi? Idee?
- Un bit per il segno, il resto posizionale?
- Non compatto e no somma in colonna
- Metodo alternativo: complemento a due!

$= 0$	0	0	0	0	$= 0$
$1 =$	0	0	0	1	$= 1$
$2 =$	0	0	1	0	$= 2$
$3 =$	0	0	1	1	$= 3$
$4 =$	0	1	0	0	$= 4$
$5 =$	0	1	0	1	$= 5$
$6 =$	0	1	1	0	$= 6$
$7 =$	0	1	1	1	$= 7$
$8 =$	1	0	0	0	$= -8$
$9 =$	1	0	0	1	$= -7$
$10 =$	1	0	1	0	$= -6$
$11 =$	1	0	1	1	$= -5$
$12 =$	1	1	0	0	$= -4$
$13 =$	1	1	0	1	$= -3$
$14 =$	1	1	1	0	$= -2$
$15 =$	1	1	1	1	$= -1$

Il Complemento a due

linguaggio rappresentazione interi con memoria a $\boldsymbol{n}$ bit

ALGORITMO DI LETTURA

- INPUT = sequenza di $\boldsymbol{n}$ bit
- Leggi la sequenza come un naturale con Horner
  $\boldsymbol{x}$ = HORNER(sequenza)
- SE primo bit = $\mathbf{0}$ ALLORA OUTPUT = $\boldsymbol{x}$
- ALTRIMENTI (primo bit = $\mathbf{1}$) OUTPUT = $-(\mathbf{2}^{\boldsymbol{n}} - \boldsymbol{x})$

- $0111\ldots1$ è il numero più grande (positivo)
- $1000\ldots0$ il più piccolo (negativo)
- Range coperto $\{-\mathbf{2}^{\boldsymbol{n}-\mathbf{1}}, \ldots, -\mathbf{1}, \mathbf{0}, \mathbf{1}, \ldots, \mathbf{2}^{\boldsymbol{n}-\mathbf{1}} - \mathbf{1}\} \subset \mathbb{Z}$

Un Algoritmo Migliore per il Complemento a Due

ALGORITMO DI LETTURA (bis)

- INPUT = sequenza di $\boldsymbol{n}$ bit
- SE primo bit = **0**

  ALLORA OUTPUT = HORNER(sequenza)

- ALTRIMENTI
  - Leggi bit da dx verso sx e ricopiali FINCHÉ non trovi **1**
  - Quando trovi un **1** ricopialo, poi nega tutti i bit successivi
  - Chiama ***y*** il risultato
  - OUTPUT = - HORNER(***y***)

L'algoritmo funziona perché implementa
la sottrazione $\mathbf{2}^{\boldsymbol{n}} - \boldsymbol{x}$ *in maniera automatica*

Nota: NEGAZIONE(1) = 0 , NEGAZIONE(0) = 1

Esercizio su complemento a 2

Leggere sequenza di 8 bit (compattata) $\boldsymbol{F2_{16}}$ secondo regole complemento a due (ad 8 bit)

- Metodo 1
  - $\boldsymbol{F2} \rightarrow \mathbf{11110010}$
  - $\boldsymbol{HORNER}(\mathbf{11110010}) = \mathbf{242_{10}}$
  - numero$= -(\mathbf{2^8} - \mathbf{242}) = -(\mathbf{256} - \mathbf{242}) = -\mathbf{14}$
- Metodo 2
  - Nego bit **11110010** che diventani **00001110**
  - $\boldsymbol{HORNER}(\mathbf{00001110}) = \boldsymbol{HORNER}(\mathbf{1110}) = \mathbf{14}$
  - numero $= -\mathbf{14}$

Esercizio su complemento a 2 (inverso)

Rappresenta il numero $-\mathbf{34}$

col complemento a due a 8 bit

$$-\mathbf{34} = -(\mathbf{2}^n - x)$$

$$x = \mathbf{2}^n - \mathbf{34} = \mathbf{256} - \mathbf{34} = \mathbf{222}$$

Horner inverso su **222** (in base **2**)

$$-\mathbf{34} \rightarrow \mathbf{11011110}$$

222 mod 2 = 0
111 mod 2 = 1
55 mod 2 = 1
27 mod 2 = 1
13 mod 2 = 1
6 mod 2 = 0
3 mod 2 = 1
1 mod 2 = 1
0

Esercitazione (Es. 1 prova scritta 2009)

Il formato **`int`** rappresenta i numeri interi mediante complemento a due a 32 bit

`short/long int` fanno lo stesso con 16/64 bit

Eseguire la seguente somma binaria (compattata in esadecimale) ed interpretarla come somma di **`short int`** (ovvero interi) e di **`unsigned short int`** (ovvero naturali):

$$8EE8_{16} + 2028_{16}$$

Esercitazione (soluzione)

8EE8	1000|1110|1110|1000	36’584	-28’952
2028	0010|0000|0010|1000	8’232	+8’232
AF10	1010|1111|0001|0000	44’816	-20’720

Esercitazione (Es. prova scritta 2015)

Ricostruire la sequenza di 16 bit che corrisponde al numero -32'700 secondo le regole del formato `short int`.

$$-32'700 = -(2^{16} - x) \Rightarrow x = 65'536-32'700 = 32'836$$

Rappresento x in base 2 (mi appoggio alla base 16)

32'836 **mod** 16 = 4
2'052 **mod** 16 = 4
128 **mod** 16 = 0
8 **mod** 16 = 8

La sequenza è 8044 $\rightarrow$ 1000|0000|0100|0100