Appunti di informatica

Lezione 3

anno accademico 2016-2017

Mario Verdicchio

Precedenze degli operatori

- ! precede &&, che precede ||,
  che precede ⊕
- !A&&B||C è equivalente a ((!A)&&B)||C
- Le parentesi vincono sulle precedenze:
  in !((A||B)&&C) la disgiunzione viene
  eseguita per prima, poi la congiunzione,
  da ultimo la negazione

Uso delle tavole di verità

- Le tavole di verità possono essere usate per dimostrare l'equivalenza di certe affermazioni
- Due affermazioni si dicono (logicamente) equivalenti quando hanno lo stesso valore di verità in tutti i casi
- Ad esempio, A xor B è equivalente a (A or B) and not(A and B)

A	B	A XOR B	A OR B	A AND B	NOT(A AND B)	(A OR B) AND NOT(A AND B)
V	V	F	V	V	F	F
V	F	V	V	F	V	V
F	V	V	V	F	V	V
F	F	F	F	F	V	F

Altre equivalenze logiche

- notA and A ≡ falso
- notA or A ≡ vero
- falso and A ≡ falso
- falso or A ≡ A
- vero or A ≡ vero
- vero and A ≡ A

- not(A and B) ≡
  not A or not B
- not(A or B) ≡
  not A and not B
- Tramite le equivalente logiche certe affermazioni possono essere semplificate in affermazioni equivalenti
- Ad esempio:

  (not A and A) or (not B and C)
  è equivalente a
  not B and C

Condizioni semplici

- Si definisce condizione semplice un'affermazione che confronta due valori
- Ha tipicamente la forma:

  valore – operatore di confronto – valore
- Esempi (con relativo valore di verità):

  $4 < 10$ (vera)
  $45 \geq 98$ (falsa)
  $x < 100$ (dipende dal valore di x al momento della verifica)

Condizioni composte

- Si costruisce una condizione composta a partire da condizioni semplici per mezzo di operatori logici
- Esempi [e relativi valori di verità]:

  $(4 < 10)$ and $(50 \geq 400)$ [falsa]

  $(10 \leq 10)$ or $(100 > 100)$ [vera]

  not $(x \leq 5)$ [dipende dal valore di x]

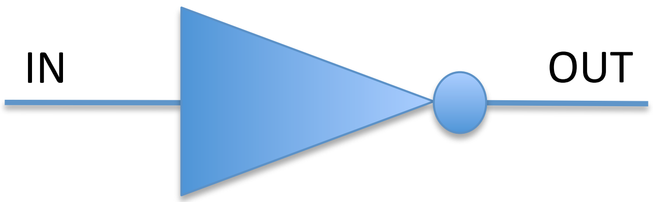
Tipi di condizioni in base alla verità

- Tautologie: condizioni, come notA or A, che sono sempre vere (qualunque sia il valore di A)
- Contraddizioni: condizioni, come notA and A, che sono sempre false (qualunque sia il valore di A)
- Tutte le altre condizioni che possono essere sia vere sia false, a seconda dei valori delle loro componenti, si dicono contingenze

Porte logiche (1)

- Negazione, NOT

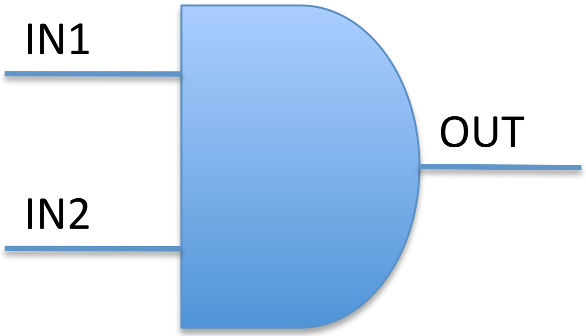
IN	OUT
0	1
1	0



Porte logiche (2)

- Congiunzione, AND

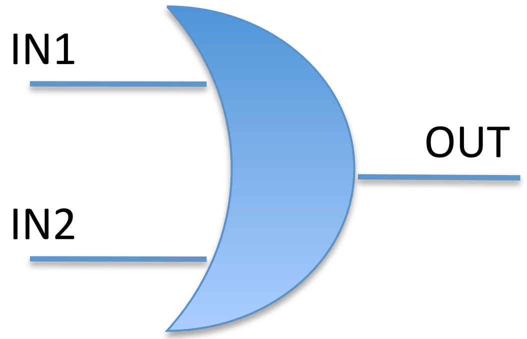
IN1	IN2	OUT= IN1 AND IN2
0	0	0
0	1	0
1	0	0
1	1	1



Porte logiche (3)

- Disgiunzione, OR

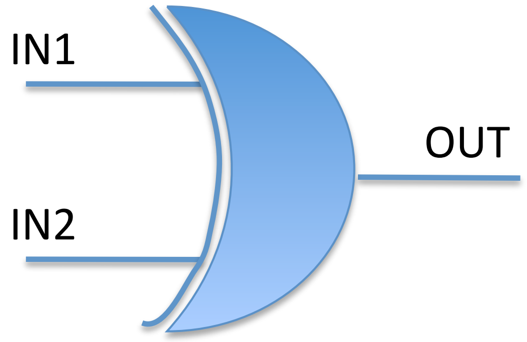
IN1	IN2	OUT= IN1 OR IN2
0	0	0
0	1	1
1	0	1
1	1	1



Porte logiche (4)

- Disgiunzione esclusiva, XOR

IN1	IN2	OUT= IN1 XOR IN2
0	0	0
0	1	1
1	0	1
1	1	0



Codifica binaria

- Il processore, abbiamo detto, lavora solo con segnali elettrici
- Segnali elettrici a valori alti di tensione vengono fatti corrispondere al numero 1
- Segnali elettrici a valori bassi di tensione vengono fatti corrispondere al numero 0
- Questa si chiama codifica binaria
- Da questo punto di vista, il calcolatore opera solo con due cifre: 0 e 1
- Tutta l’informazione che un calcolatore elabora viene espressa con queste due cifre, per mezzo della codifica binaria

Numerazione: le basi

- Noi siamo abituati a una numerazione basata su 10 cifre: da 0 a 9
- La base della nostra numerazione è il 10:

  $147 = 7 \times 10^0 + 4 \times 10^1 + 1 \times 10^2$
- La base del sistema binario è il 2:

  $1011 = 1 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + 1 \times 2^3$

  Se si svolge il calcolo si ottiene il numero che, in base 10, corrisponde a 1011 in base 2.

Base 2 -> Base 10

- 1011_2, ossia 1011 in base 2, a che numero in base 10 corrisponde?
- Stiamo cercando la x tale che: $1011_2 = x_{10}$
- Basta ricordarsi la definizione di base 2:

  $1011 = 1 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 = 11$
- Perciò 1011 in base 2 vuol dire 11 in base 10:

  $1011_2 = 11_{10}$
- Il “nostro” 11 è per il calcolatore 1011

Base 10 -> Base 2

- Il metodo per esprimere in base 2 un numero dato in base 10 è il seguente
- Cerchiamo la x tale che: $25_{10} = x_2$
- Si procede con una sequenza di divisioni per 2, fintantoché il quoziente non diventa 0, e scrivendo la sequenza dei resti in ordine inverso

Come si scrive 25 in base 2?

- 25 : 2 = 12 con resto 1
- 12 : 2 = 6 con resto 0
- 6 : 2 = 3 con resto 0
- 3 : 2 = 1 con resto 1
- 1 : 2 = 0 con resto 1
- Una volta ottenuto il quoziente pari a 0, scriviamo i resti in ordine inverso:

  $25_{10} = 11001_2$

Verifica del metodo

- 11001 è davvero la codifica in base 2 di 25?
- Basta svolgere i calcoli basati sulla definizione di sistema binario
- $11001_2 = 1 \times 2^0 + 1 \times 2^3 + 1 \times 2^4 = 1 + 8 + 16 = 25_{10}$ (gli addendi con lo 0 sono stati omessi perché ovviamente non influiscono sulla somma)