

Compito del 22 giugno 2005

★ **SOLUZIONI** ★

1/a

Indovinello

Nel paesino di Bianconero, i manigoldi dicono sempre il falso, e i cavalieri sempre il vero. Di Bianconero sono A e B. “Siamo entrambi manigoldi”, dice il primo. Chi è chi?

Risolvere l'indovinello adoperando le proprietà dell'algebra di Boole e/o le tabelle di verità. Associare il valore 1 alle proposizioni vere, e 0 a quelle false.

soluzione

Indichiamo con α , β e γ le variabili booleane associate rispettivamente ai valori di verità ($1 \mapsto$ vero, $0 \mapsto$ falso) delle proposizioni “A è un cavaliere”, “B è un cavaliere”, “A e B sono entrambi manigoldi”. L'ultima frase è quella pronunciata da A: il suo valore di verità dipende dai valori di verità delle altre due frasi, e si può scrivere che $\gamma = \Gamma(\alpha, \beta)$, con $\Gamma(\cdot)$ funzione booleana di due variabili. Nel nostro caso, si ha

$$\Gamma(\alpha, \beta) = \bar{\alpha}\bar{\beta} . \quad (1)$$

Data una qualsiasi proposizione P, essa è logicamente compatibile con la natura di chi la pronuncia (cavaliere o manigoldo), quando essa risulta vera se a pronunciarla è un cavaliere, e/o (or inclusivo) risulta falsa se a pronunciarla è un manigoldo. Dunque la particolare proposizione del problema è compatibile con la natura di A se $\alpha = \Gamma(\alpha, \beta)$ è verificata per qualche coppia (α, β) . In altre parole, gli eventuali valori delle variabili α e β per cui (l'operatore NXOR verifica l'uguaglianza di due variabili)

$$\overline{\alpha \oplus \Gamma(\alpha, \beta)} = \alpha \Gamma(\alpha, \beta) + \bar{\alpha} \overline{\Gamma(\alpha, \beta)} = 1 \quad (2)$$

forniscono la soluzione del problema dato. Si noti che la soluzione non è in generale unica, potendo accadere che le espressioni $\alpha \Gamma(\alpha, \beta)$ e $\bar{\alpha} \overline{\Gamma(\alpha, \beta)}$ valgano 1 simultaneamente o separatamente per diverse coppie (α, β) . Può anche accadere che nessuna coppia (α, β) soddisfi l'equazione 2, nel qual caso il problema non ammette soluzione. Sostituendo l'eq. 1 nella 2 si ricava immediatamente che nel nostro caso dev'essere $\bar{\alpha}\beta = 1$, ossia che la soluzione è unica, e si ha che A è un manigoldo ($\alpha = 0$) e B è un cavaliere ($\beta = 1$).

Nel paesino di Bianconero, i manigoldi dicono sempre il falso, e i cavalieri sempre il vero. Di Bianconero sono A e B. “Uno di noi almeno è un manigoldo”, dice il primo. Chi è chi?

soluzione

Lavoriamo stavolta con le tabelle di verità, costruendo valore per valore prima la funzione $\Gamma(\alpha, \beta)$, e poi la *funzione di compatibilità* $\Theta(\alpha, \beta) = \alpha \Gamma(\alpha, \beta) + \bar{\alpha} \overline{\Gamma(\alpha, \beta)}$: le eventuali righe in cui tale funzione assume valore 1 forniscono le configurazioni soluzione del problema. Quando la funzione di compatibilità è espressa da un solo mintermine, allora la soluzione è unica.

α	β	$\gamma = \Gamma(\alpha, \beta)$	$\alpha \gamma$	$\bar{\alpha} \bar{\gamma}$	$\Theta(\alpha, \beta)$
0	0	1	0	0	0
0	1	1	0	0	0
1	0	1	1	0	1
1	1	0	0	0	0

È quanto accade in questo caso. La soluzione è che A è un cavaliere ($\alpha = 1$) e B è un manigoldo ($\beta = 0$).

Approfondimento. *Supponiamo che A pronunci la frase $P_A = \text{“Se B è un manigoldo, io sono un cavaliere”}$, e B pronunci la frase $P_B = \text{“A ed io siamo diversi”}$: cosa si può concludere?* Stavolta la sola affermazione di A non permette di risolvere in modo univoco il problema dell'identità di A e B. Infatti, la funzione di compatibilità $\Theta_A(\alpha, \beta) = \alpha \Gamma_A(\alpha, \beta) + \bar{\alpha} \overline{\Gamma_A(\alpha, \beta)}$, con $\Gamma_A(\alpha, \beta) = \bar{\beta} \rightarrow \alpha = \beta + \alpha = \bar{\beta} \cdot \bar{\alpha}$, vale $\Theta_A(\alpha, \beta) = \bar{\beta} + \alpha = \beta \rightarrow \alpha$: da essa si ricava soltanto che non può accadere che A sia manigoldo ($\alpha = 0$) e B cavaliere ($\beta = 1$). (L'operatore \rightarrow , definito da $p \rightarrow q = \bar{p} + q$, prende il nome di “implicazione materiale” in logica proposizionale: esso stabilisce che la conclusione q segue sempre dalla premessa p , salvo nel caso in cui la premessa sia vera ($p = 1$) e la conclusione falsa ($q = 0$) — ad es., “se l'asino vola allora $1 + 1 = 3$ ” è vera, mentre “se il merlo vola allora $1 + 1 = 3$ ” è falsa.) La frase di B consente peraltro di rimuovere l'incertezza sull'identità di A e B. Infatti, è $\Gamma_B(\alpha, \beta) = \alpha \oplus \beta$, da cui $\Theta_B(\alpha, \beta) = \bar{\alpha}$. Ora, dovendosi avere congiuntamente (prodotto logico) $\Theta_A = 1$ e $\Theta_B = 1$, si può costruire la funzione di compatibilità totale del problema $\Theta(\alpha, \beta) = \Theta_A(\alpha, \beta) \Theta_B(\alpha, \beta) = \bar{\alpha} \bar{\beta}$, che assume valore 1 solo se entrambi A e B sono manigoldi. Con le tabelle di verità:

α	β	$\Gamma_A(\alpha, \beta)$	$\Theta_A(\alpha, \beta)$	$\Gamma_B(\alpha, \beta)$	$\Theta_B(\alpha, \beta)$	$\Theta(\alpha, \beta)$
0	0	0	1	0	1	1
0	1	1	0	1	1	0
1	0	1	1	1	0	0
1	1	1	1	0	0	0

2/a La DOS trap INT 21h, funzione AH=2ah (“get date”), restituisce in CX l’anno corrente (1980–2099), in DH il mese dell’anno (01h=gennaio), in DL il giorno del mese (1–31) e in AL il giorno della settimana (00h=domenica). Scrivere un programma Assembly 8086 che calcoli e stampi a video in quale giorno della settimana cade il giorno 17 giugno dell’anno corrente.

soluzione

TITLE dayofweek: per esame 22/6/2005

```

comment *
Si deve calcolare il giorno della settimana
corrispondente ad un giorno prefissato dell’anno in corso
(il programma deve funzionare in qualsiasi giorno
dell’anno in corso).
*

;-----
; Definizione costanti
CR EQU 13          ; carriage return
LF EQU 10          ; line feed
DOLLAR EQU '$'

;-----
;
PILA SEGMENT STACK 'STACK' ; definizione del segmento di stack
        DB 16 DUP('STACK') ; lo stack e' riempito con la stringa 'stack'
                                ; per identificarlo meglio in fase di debug
PILA ENDS

;-----
;
DATI SEGMENT PUBLIC 'DATA' ; definizione del segmento di stack
settimana DB 'Sunday' ,DOLLAR
daylen    equ $-settimana
          DB 'Monday' ,DOLLAR
          DB 'Tuesday' ,DOLLAR
          DB 'Wednesday',DOLLAR
          DB 'Thursday' ,DOLLAR
          DB 'Friday' ,DOLLAR
          DB 'Saturday' ,DOLLAR
mesi      dw 31,28,31,30,31,30,31,31,30,31,30,31 ; trenta di'...
venturus_flag db 0 ; default: il giorno prefissato e' gia' trascorso
doy_current  dw ?
doy_target   dw ?
dom_target   db 17
moy_target   db 6
frase        db "The 17th of June of the current year is a ",DOLLAR
CRLF LF      DB CR,LF,LF,DOLLAR
DATI ENDS

```

```

;=====
;
;   M A C R O
;
;=====
day_of_year macro doy    ; non si contemplano gli anni bisestili...
local innerloop,done
    xor ch,ch
    mov cl,dh            ; in cx: month (January=01h)
    xor dh,dh            ; in dx: day of month
    cmp cx,1
    je done
    dec cx
    xor bx,bx
innerloop:
    add dx,mesi[bx]
    add bx,2
    loop innerloop
done:
    mov doy,dx
endm

display macro stringa
    push dx
    push ax
    mov dx,offset stringa
    mov ah,09h
    int 21h
    pop ax
    pop dx
endm

;-----
;
CSEG SEGMENT PUBLIC 'CODE'

MAIN PROC FAR
    ASSUME CS:CSEG,DS:DATI,SS:PILA,ES:NOTHING;

    MOV AX,SEG DATI
    MOV DS,AX

getdate:
    mov ah,2ah
    int 21h                ; CX=year (qui inutile), DH=month (N.B. gennaio=01h),
                          ; DL=day of month, AL=day of week
    day_of_year doy_current

    mov dh,moy_target
    mov dl,dom_target
    day_of_year doy_target

    push ax                ; salva current day of week
    mov ax,doy_current
    mov dx,doy_target
    sub ax,dx

```

```

        cmp ax,0
        jge diff_in_ax
        mov venturus_flag,1 ; il giorno prefissato deve ancora arrivare
        neg ax ; cambia segno alla differenza per renderla positiva
diff_in_ax:
        ; non si effettua la divisione per evitare divide overflow:
        ; si fanno ripetute sottrazioni anziche' eseguire il codice
        ;
        mov dh,7
        ;
        div dh ; AL<-AX:DH; il resto va in AH
        ;
        mov bl,ah
divide_loop:
        sub ax,7
        cmp ax,0
        jge divide_loop
        add ax,7 ; ax in [0,6]
        mov bl,al ; in al il resto della divisione, in [0,6]
        pop ax ; ripristina current day of week in al, in [0,6]
check_venturus:
        cmp venturus_flag,1
        je venturus
        ; se il giorno prefissato e' gia' trascorso, il giorno
        ; della settimana corretto si ottiene sottraendo bl
        ; dal giorno della settimana corrente, che e' in al...
        sub al,bl ; al in [-6,6]
        cmp al,0
        jge dow_in_AL
        add al,7 ; al in [0,6]
        jmp dow_in_AL
        ;...altrimenti bisogna aggiungere bl ad al
venturus:
        add al,bl ; in [0,12]
        cmp al,7
        jl dow_in_AL
        sub al,7 ; al in [0,6]
dow_in_AL:
        mov bh,daylen
        mul bh ; ax<-al*bh+offset settimana=offset del day_of_week da stampare
        add ax,offset settimana

        display frase

        mov dx,ax ; display del giorno della settimana trovato
        mov ah,09h ; vedi macro display
        int 21h

        DISPLAY CRLF

        mov ah,4ch ; ritorno al DOS
        int 21h

MAIN ENDP

CSEG ENDS

END MAIN ; il programma comincia all'indirizzo di MAIN

```