

PROVA SCRITTA DI CALCOLATORI ELETTRONICI DEL 11/2/2015

(Tempo a disposizione: 3,5 ore)

ESERCIZIO 1 (Tutti):

Si realizzi una rete sequenziale sincrona R con un ingresso X ed una uscita Z. La rete riconosce stringhe della forma b_0b_1S dove S è definita come segue:

- (a) se $b_0b_1=00$, S è vuota (per cui la stringa riconosciuta in questo caso è 00)
- (b) se $b_0b_1=01$, S vale $\alpha 0$, con $0 \notin \alpha$
- (c) se $b_0b_1=10$, S vale $\alpha 0\beta 00\gamma 0$, con $0 \notin \alpha, \gamma$ ed, inoltre, $00 \notin \beta$
- (d) se $b_0b_1=11$, S vale $\alpha 0\beta 00\gamma 000\delta 00\epsilon 0$, con $0 \notin \alpha, \epsilon$; $00 \notin \beta, \delta$ ed, inoltre, $000 \notin \gamma$

La rete fornisce in uscita un 1 in corrispondenza dell'ultimo bit della sequenza S riconosciuta e riprende, quindi, il suo funzionamento da principio. Segue un esempio di possibile funzionamento di R:

t:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
X:	0	1	1	1	1	0	0	0	1	0	0	0	1	0	1	1	0	0	0	...
Z:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	...

La rete legge innanzitutto la sequenza 01, e si trova dunque nel caso (b). Agli istanti di tempo 2, 3 e 4 legge degli 1, che costituiscono la stringa α , per poi ricevere un bit a 0 nell'istante di tempo 5 che comporta il riconoscimento della sequenza (per cui $z=1$). A questo punto, il lavoro della rete ricomincia daccapo.

ESERCIZIO 2 (DM270 – A. A. 2012/13 e 2013/14):

Estendere il set di istruzioni della macchina a stack con l'operazione **MaxSuff**, definita come segue. Data una lista chiusa il cui puntatore di testa è memorizzato in cima allo stack, l'istruzione restituisce come nuova cima dello stack il puntatore al primo nodo della lista che è testa del più lungo suffisso ordinato in senso crescente della lista di partenza.

La figura sulla sinistra mostra lo stato della memoria al termine dell'esecuzione dell'istruzione, laddove l'indirizzo risultato, 7503, punta al più lungo suffisso ordinato, che contiene i valori 2, 4, 18, della lista in input la cui testa è memorizzata all'indirizzo 8231.

SP	1052	8231
1054	1053	7503
:	:	:
3400	:	-1
3401	:	18
:	:	:
4056	:	3401
4057	:	4
:	:	:
6500	:	7503
6501	:	7
:	:	:
7502	:	4057
7503	:	2
:	:	:
8230	:	6501
8231	:	13

ESERCIZIO 2 (DM270 – A. A. precedenti):

Estendere il set di istruzioni della macchina a registri con l'operazione **SUM@# Ri, Rj, Rk**, definita come segue. A partire dalla locazione il cui indirizzo è memorizzato in Ri è memorizzato un array V di n elementi, dove n è il valore contenuto in Rj. L'istruzione restituisce in Rk la lunghezza del più lungo suffisso ordinato in senso crescente di V.

Si consideri il seguente esempio in cui $R_i=947$ e $R_j=7$. Al termine dell'esecuzione dell'istruzione, Rk conterrà il valore 3 perché il più lungo suffisso ordinato in senso crescente contiene 3 elementi ossia 2, 7 e 11.

947	948	949	950	951	952	953
13	8	-1	4	2	7	11

ESERCIZIO 2 (DM509):

Estendere il set di istruzioni della macchina a stack con l'operazione **PrefLeng**, definita come segue.

A partire dalla locazione X della RAM è memorizzato un array di 32 elementi. L'istruzione memorizzerà nell'accumulatore la lunghezza del più lungo prefisso ordinato in senso crescente.

Si consideri il seguente esempio in cui $X=947$. Al termine dell'esecuzione dell'istruzione l'accumulatore conterrà il valore 4.

	3	7	11	15	3	19	
947	948	949	950	951	978		

ESERCIZIO 3 (DM 270 – 9CFU/6CFU):

Scrivere una procedura assembly che riceve un vettore di word V e restituisce la lunghezza del più lungo suffisso ordinato in senso crescente di V.

Scrivere inoltre il programma principale che invochi opportunamente la procedura descritta.

Ad esempio, sia $V = [13, 8, -1, 4, 2, 7, 11]$. La procedura restituirà il valore 3 perché il più lungo suffisso ordinato in senso crescente contiene 3 elementi ossia 2, 7 e 11.