

Fondamenti di Programmazione con Laboratorio CdL in Matematica

Prima Verifica Intermedia 11 novembre 2019

N.B.: Negli esercizi di programmazione, vengono valutati anche l'uso delle condizioni booleane, la leggibilità e l'efficienza del codice proposto. Inoltre, non è consentito l'uso di variabili globali e di istruzioni che alterino il normale flusso dell'esecuzione all'interno di cicli e provochino l'uscita forzata. Infine, è possibile presupporre le seguenti direttive di compilazione:

```
# define FALSE 0
# define TRUE 1
```

ESERCIZIO 1 (8 punti)

- Costruire l'NFA A che riconosce il linguaggio L composto da tutte le stringhe w in $\{a, b, c\}^*$ che **contengano** ab o bb .
- Ricavare il DFA corrispondente ad A , **usando** l'algoritmo di costruzione per sottoinsiemi.
- Costruire il DFA A' che riconosce il linguaggio L' composto da tutte le stringhe w in $\{a, b, c\}^*$ tali che **non** contengano occorrenze né di ab , né di bb .

ESERCIZIO 2 (7 punti)

Si scriva in C una funzione che legga da linea di comando (*senza* usare array di appoggio) una sequenza di interi positivi n_1, n_2, \dots che termina non appena l' i -esimo numero della sequenza n_i risulti essere il predecessore dell' $(i-1)$ -esimo numero n_{i-1} (ovvero quando $n_i = n_{i-1} - 1$). La funzione sviluppata deve restituire il numero di elementi letti.

Ad esempio la funzione restituirebbe 11 se la sequenza fosse:

6, 8, 49, 34, 99, 32, 98, 5, 3, 78, 77

N.B.: Non occorre scrivere il `main`.

ESERCIZIO 3 (7 punti)

Si scriva in C una funzione che, dato un array \mathbf{a} di interi, di dimensione `dim` che si suppone **multiplo di 3**, controlli che sia composto da una serie di **triple** formate da un intero, il suo doppio e il suo triplo (ovvero da triple (x, y, z) tali che $y = 2x$ e $z = 3x$). Le triple devono essere **consecutive**, **disgiunte** e partire dalla prima posizione (ovvero devono essere tali che la prima si trova alle posizioni 0, 1 e 2, la seconda alle posizioni 3, 4 e 5, e così via).

La funzione deve restituire **TRUE** se l'array è formato da triple strutturate come richiesto, **FALSE** altrimenti. Ad esempio, se l'array fosse

1	2	3	14	28	42	5	10	15
---	---	---	----	----	----	---	----	----

la funzione restituirebbe **TRUE**, mentre restituirebbe **FALSE** se l'array fosse

3	9	5	25	9	81	28	45	7
---	---	---	----	---	----	----	----	---

N.B.: Non occorre scrivere il `main`: basta scrivere la chiamata alla funzione con gli opportuni parametri.

ESERCIZIO 4 (4 punti)

Si descriva a parole (massimo 10 righe) l'algoritmo **INSERTION SORT**, e se ne scriva l'*invariante di ciclo*, ovvero la proprietà che è vera all'inizio di ogni iterazione o esecuzione del ciclo.

ESERCIZIO 5 (4 punti)

Date due espressioni aritmetiche a_0, a_1 e uno stato σ , scrivere la valutazione dell'espressione composta $a_0 + a_1$, ovvero completare la regola premessa/conclusione della semantica operativa:

$$\frac{\dots \quad \dots}{\langle a_0 + a_1, \sigma \rangle \rightarrow \dots}$$

Come si applicherebbe all'espressione $a = (Y + 3) + (4 + 2)$, partendo dallo stato σ_0 in cui $\sigma_0(Y) = 1$, ovvero quale sarebbe l'albero di valutazioni?