

Tempo a disposizione: ore 2.

**Svolgere gli esercizi 1-4 e 5-8 su due fogli differenti.**

1. La seguente espressione

$$\mathcal{I}_{L_0}^{L_1}(\mathcal{C}_{L_1, L_0}^{L_0}, \mathcal{I}_{L_0}^{L_1})$$

calcola qualcosa di utile? Se rimpiazziamo, nell'espressione sopra, la seconda occorrenza di  $\mathcal{I}_{L_0}^{L_1}$  con  $\mathcal{I}_{L_1}^{L_0}$ , cosa otteniamo?

2. Se  $L$  è libero ed  $R$  è libero deterministico, il linguaggio  $L \cup \overline{R} = \{w \in A^* \mid w \in L \vee w \notin R\}$  è regolare o libero, oppure non libero? Giustificare la risposta.
3. Mostrare che  $L = \{a^n b^{2n+1} \mid n \geq 0\}$  è libero deterministico, costruendo un opportuno DPDA che riconosca  $L\$$  per pila vuota.
4. Si consideri la grammatica  $G$  con simbolo iniziale  $S$ :

$$\begin{aligned} S &\rightarrow Sa \mid bB \\ B &\rightarrow \epsilon \mid bB \end{aligned}$$

(i) Costruire l'automa canonico LR(0). (ii) Costruire la tabella di parsing SLR(1) e verificare se ci sono conflitti. (iii) Mostrare il funzionamento del parser SLR(1) per l'input  $baa$ .

5. E' possibile definire in Java due tipi A e B, tali che A sia un supertipo di B e *non* vi sia ereditarietà tra A e B? Se si fornire un esempio, se no motivare la risposta.
6. Si dica cosa è l'aliasing e perché può rendere difficoltosa la verifica della correttezza dei programmi. E' possibile inserire nel compilatore di un ipotetico linguaggio un controllo che permetta di identificare tutte le situazioni di aliasing? Perché?
7. In alcuni linguaggi a run-time sono controllati gli accessi agli array per evitare di accedere ad una posizione oltre i limiti fissati per gli indici dell'array. Supponendo che un array locale di una procedura sia memorizzato nel record di attivazione della procedura stessa (come spesso avviene), quali problemi potrebbero sorgere dal non effettuare questo controllo a run-time?
8. Si dica cosa viene stampato dal seguente frammento di codice scritto in uno pseudo-linguaggio che usa scoping statico e passaggio di parametri per nome.

```
int x=2
void foo (nome int y){
    x = x + 1;
    write(x);
    y = (y++) + (y++);
    write(y);
}
{ int x=50
  foo (x);
  write (x);
}
```