

Tempo a disposizione: ore 2.

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

1. Definire una grammatica G che generi il linguaggio $L = \{a^n b^n a^m b^m \mid n \geq 0, m \geq 1\}$. Il linguaggio L è libero, ma non regolare?
2. Si costruisca il più semplice automa che riconosca il linguaggio $L = \{a^n b^m \mid n \geq 1, m \geq 0\}$. Ricavare da tale automa una grammatica equivalente.
3. Si consideri la grammatica G con simbolo iniziale S :

$$\begin{aligned} S &\rightarrow SA \mid \epsilon \mid cDB \\ A &\rightarrow \epsilon \mid a \\ B &\rightarrow b \mid A \\ C &\rightarrow aSB \mid c \\ D &\rightarrow cD \end{aligned}$$

Semplificare la grammatica G , eseguendo, nell'ordine specificato, le seguenti trasformazioni: (i) rimuovere la ricorsione sinistra; (ii) eliminare le produzioni epsilon; (iii) eliminare le produzioni unitarie; (iv) rimuovere i simboli inutili. La grammatica risultante è del tutto equivalente a G ?

4. Data la seguente grammatica G con simbolo iniziale S :

$$\begin{aligned} S &\rightarrow \epsilon \mid A \\ A &\rightarrow \mathbf{b}B \mid \mathbf{c}A\mathbf{a} \mid \mathbf{c}\mathbf{a} \\ B &\rightarrow \epsilon \mid \mathbf{b}B \end{aligned}$$

si verifichi che non è di classe LR(0), ma è di classe SLR(1). Si mostri il funzionamento del parser SLR(1) sull'input cba .

5. Si dica brevemente (ma con precisione) quali sono le differenze tra un costrutto di iterazione controllato logicamente e un costrutto di iterazione controllato numericamente.
6. Si assuma che in un generico linguaggio imperativo a blocchi, senza ricorsione, il blocco A contenga una chiamata della funzione f . Il numero dei record di attivazione (RdA) presenti a run-time sulla pila fra il RdA di A e quello della chiamata di f è fissato staticamente o può variare dinamicamente? Motivare la risposta, riflettendo sul significato della parola "contenga" in questo contesto.
7. 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 e per valore.

```
int x = 50;
void pippo(name int y, value int z){
    z = y + y + x;
}
{ int x = 1;
  int y = 10;
  int z = 20;
  pippo(x++, x);
  pippo(x++, x);
  write(x);
}
write(x);
```

8. In un linguaggio funzionale come Scala si considerino le seguenti due funzioni:

```
def loop: Boolean = loop
def first_arg(x: Int, y: Boolean): Int = x.
```

Descrivere la valutazione dell'espressione

```
first_arg(5, loop)
```

secondo il modello computazionale per sostituzione con valutazione *call-by-value* (chiamata per valore). Ripetere l'esercizio con valutazione *call-by-name* (per chiamata per nome).