

Tempo a disposizione: ore 2.

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

Per Paradigmi: svolgere solo: 3,5,6,7,9.

1. Dato l'espressione regolare $a^*(a|a)a^*$, costruire l'automa minimo deterministico equivalente (ovvero che riconosce lo stesso linguaggio denotato da $a^*(a|a)a^*$) a partire dalla costruzione del NFA canonico associato. Giustificare come lo si è ottenuto.
2. Mostrare che $L_1 = \{a^n b^{n+1} c^m \mid n, m \geq 0\}$ è libero deterministico, costruendo un opportuno DPDA. Sapendo che anche $L_2 = \{a^n b^m c^{m+1} \mid n, m \geq 0\}$ è libero deterministico, è libero il linguaggio $L_1 \cap L_2$? Motivare la risposta.
3. Data la seguente grammatica G con simbolo iniziale S :

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

si determini il linguaggio $L(G)$ generato da G . Si semplifichi G rimuovendo le produzioni epsilon, ottenendo la grammatica G' . Si discuta se la grammatica risultante G' sia del tutto equivalente alla grammatica 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 $cbba$.

5. L'esecuzione del seguente frammento di codice su una certa implementazione risulta nella stampa di un valore dispari.

```
int x = 10;
void fie( int j){
    j = j+1;
}
void foo( int y, int z){
    int k = z;
    fie(y);
    z = z + k;
    V[x] = 4;
}
int V[10];
for (int i=0, i<10, i++) V[i]=i;

{ int x = 2;
  foo(x,V[x]);
  write (V[x]);
}
```

Si fornisca una possibile spiegazione.

6. Si consideri il seguente frammento in uno pseudolinguaggio con e parametri di ordine superiore:

```
int x = 700;
int n = 30;
void g(){
    write(n+x)
}
void foo (int f(), int n){
    if (n==0) f();
    else foo(f,0);
    g();
}
{
    int x = 5;
    foo(g,1)
}
```

Si dica cosa stampa il frammento con (i) shallow binding e scope dinamico ; (ii) deep binding e scope statico.

7. Facendo riferimento (preferibilmente) al linguaggio Java, si dica se, in generale, il downcasting è sempre corretto oppure no, fornendo esempi di codice che giustifichino quanto detto.
8. (Solo Linguaggi) Si confrontino brevemente le caratteristiche sia funzionali che implementative dei meccanismi di comunicazione sincrona e asincrona.
9. (Solo Paradigmi) Usando uno pseudolinguaggio che ammetta l'uso dei puntatori si fornisca un frammento di codice che generi un "dangling reference". Si faccia quindi vedere come con la tecnica dei "locks and keys" non si ha più tale problema. Avremmo potuto usare la tecnica del contatore dei riferimenti invece del "locks and keys" ? Motivare la risposta.