

Tempo a disposizione: ore 2.

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

1. Nella seguente espressione

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

l'interprete su quale macchina ospite si basa e quale la macchina astratta realizza? Il programma risultante è utile? E può essere eseguito sulla macchina ospite?

2. Se $L = \{a, b, ab\}$ e $R = \{a\}$, che linguaggio è $L \cup R^*$? In generale, se L e R sono regolari, il linguaggio $L \cup R^*$ è regolare o libero, oppure non libero? Giustificare la risposta.
3. Si consideri la seguente grammatica G con simbolo iniziale S :

$$\begin{aligned} S &\rightarrow BC \mid AC \\ A &\rightarrow a \mid BA \\ B &\rightarrow \epsilon \mid bSB \\ C &\rightarrow d \mid cC \end{aligned}$$

(i) Si calcolino i First e i Follow per tutti i nonterminali. (ii) La grammatica G è di classe LL(1)? (iii) Si rimuova la produzione epsilon per ottenere una grammatica G' senza produzioni epsilon, che sia equivalente a G .

4. Si consideri la grammatica G con simbolo iniziale S :

$$\begin{aligned} S &\rightarrow bSCa \mid c \\ C &\rightarrow \epsilon \mid cC \end{aligned}$$

(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 *bcca*.

5. Si consideri il seguente codice dove il costrutto `union` permette di realizzare un record variante (analogamente al C)

```
const int N = 10;

union tipoCella{
    int cellaInt;
    float cellaFloat;
};

struct tipoVettore{
    int m;
    float n;
    tipoCella vett[N];
};

struct tipoVettore x;
```

Si descriva il layout di memoria di `x` assumendo una parola di 32 bit, allineamento alla parola, che un `int` occupi 16 bit e un `float` occupi 32 bit.

6. Si dica cosa stampa il seguente frammento di programma assumendo scope dinamico e shallow binding:

```
{int x = 10;
 int n = 100;
 int stampa(){
     write(x);
 }
 void foo (int f(), int g()){
     int x = 30;
     int n = 300;
     f();
     g();
     ass_x();
     stampa()
 }
 {int x = 20;
  int n = 200;
  int ass_x(){
      x= x+n;
  }
  foo(ass_x(), stampa());
 }
 }
```

7. Si dica cosa stampa il seguente frammento di programma assumendo scope statico e passaggio dei parametri per nome.

```
int i = 1;
int[] A = new int[10];
void fie (int x name, int y name) {
    i= x++;
    A[i] = 10;
    y = 20;
    i= x++;
    A[i] = 30 ;
}
for (j = 0; j <= 9; j+= 1)
    {A[j] = 0};

{int i = 5;
 fie (i,A[i]);
 for (j = 0; j <= 9; j+= 1)
     {write(A[j] )}
}
write (A[i] )
```

8. Si dica motivando brevemente la risposta cosa viene stampato dal seguente frammento in un linguaggio con eccezioni

```
class X extends throwable{
}
class Y extends X{
}

void f() throws X {
    throw new X();
}

void g (int sw) throws X , Y {
    if (sw == 0) {f();}
    try {f();} catch (Y e) {write("in_g");}
}
...
try {g(1);}
    catch (X e) {write("in_main");}
```