

Tempo a disposizione: ore 2.

SCRIVERE LE SOLUZIONI A 1-4 E 5-8 SU DUE FOGLI DIVERSI

1. Dare una grammatica libera da contesto che genera il linguaggio

$$\{(ba)^n(ab)^m \mid n \geq 1, m \geq 0\}$$

2. Si diano tre esempi di vincoli di semantica statica.
3. Si dica, motivando la risposta, quali delle seguenti regole (o produzioni) possono comparire in una grammatica libera da contesto (i simboli maiuscoli sono non-terminali, quelli minuscoli sono terminali):

$$A \rightarrow aAa, \quad AB \rightarrow A, \quad A \rightarrow AA, \quad A \rightarrow aa, \quad a \rightarrow A, \quad a \rightarrow a$$

4. Si dica cosa stampa il seguente frammento in uno pseudolinguaggio con passaggio per riferimento e scope dinamico

```
int x = 2;
void foo(reference int y){
    x = x+1;
    y = y+10;
    x = x+y;
    write(x);
}
{int x = 10;
  foo(x);
  write(x);
}
```

5. Si dica, motivando la risposta, cosa viene stampato dal seguente frammento di codice scritto in uno pseudo-linguaggio che usa scoping statico e passaggio di parametri per valore e per nome.

```
int x = 1;
int y = 2;
int z = 3;
void pippo(value int x, name int y){
    x = x + x;
    y = y + y;
    z = z + z;
    write(x,y,z)
}
{ int x = 10;
  int y = 20;
  int z = 30;
  pippo(x++, y++);
  write(x,y,z);
}
write(x,y,z));
```

(la primitiva `write(x)` permette di stampare un valore intero; un comando della forma `foo(w++)`; passa a `foo` il valore corrente di `w` e poi incrementa `w` di uno).

6. Si assuma che in un generico linguaggio imperativo a blocchi, 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.
7. Si consideri il seguente frammento di codice Java, dove *A* e *B* sono due classi diverse, di tipo non confrontabile:

```
A[] aa = new A[10];
Object[] oo = aa;
oo[0] = new B;
```

Si dica se il precedente frammento è corretto oppure no dal punto di vista dei tipi.

8. Si dica cosa viene stampato dal seguente frammento di codice scritto in uno pseudo-linguaggio che permette il passaggio di funzioni come parametro, usa deep binding, scoping statico e passaggio per valore.

```
int f( int function g (value int n), value int k) {
    int somma(value int z){
        return z+k;
    };
    int x = 10;
    int y = 10;

    if k = 1 return g(x)+g(y) else
    {int x = 30;
     int y = 30;
     return f (somma, k-1)
    }
}
int foo(value int n){
}
int x = 50;
int y = 50;
x= f(foo,2);
write(x);
```