

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 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 riferimento.

```
int x = 0;
void pippo(value int y, reference int z){
    z = x + y + z;
}
{ int x = 1;
  int y = 10;
  int z = 20;
  pippo(x++, x);
  pippo(x++, x);
  write(x);
}
write(x);
```

(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[] as = new A[10];
Object[] ao = as;
B[0] = new B();
```

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

8. Si consideri l'iterazione determinata, realizzata mediante un comando della forma

```
for I = inizio to fine by passo do C
```

dove l'indice *I* è una variabile che non può essere modificata (né esplicitamente, né implicitamente) nel corso dell'esecuzione del corpo *C*, mentre *inizio*, *fine* e *passo* sono delle costanti.

Si dica, motivando la risposta, se un tale comando può essere espresso mediante un comando `while` e, vice versa, se un comando `while` può essere espresso da un comando di iterazione determinata nel contesto un linguaggio imperativo che abbia anche i comandi di assegnamento, sequenza e condizionale.