

Tempo a disposizione: ore 2.

1. Si dia il linguaggio generato dalla grammatica $G = (\{A, B, C\}, \{a, b\}, A, P)$ dove P sono le produzioni seguenti e ϵ indica la stringa vuota:

$$\begin{aligned} A &::= aCa \mid bCb \mid B \\ B &::= a \mid b \mid \epsilon \\ C &::= aC \end{aligned}$$

2. Sia $\mathcal{I}_{L_1}^L$ un interprete per L_1 scritto in L e $\mathcal{C}_{L_2, L_3}^{L_1}$ un compilatore da L_2 a L_3 scritto in L_1 . Si consideri ora l'espressione $\mathcal{I}_{L_1}^L(\mathcal{C}_{L_2, L_3}^{L_1}(x))$. Qual è la più ovvia condizione su x (cioè: cosa ci aspettiamo che sia x) affinché questa espressione abbia un chiaro significato? In tale condizione, cosa denota l'espressione assegnata?
3. Usando un qualsiasi pseudolinguaggio si fornisca un frammento di codice che generi un "dangling reference". Si faccia quindi vedere come con la tecnica delle "lock and keys" non si ha più tale problema.
4. Si consideri un linguaggio con scope dinamico, implementato mediante catena CRT, nel quale tutti gli identificatori sono noti staticamente. Si prenda in considerazione l'operazione di "accesso ad una variabile non locale" nel corpo di una funzione; nel seguito indicheremo tale variabile con x . A quale dei seguenti parametri è proporzionale il tempo necessario all'esecuzione di tale operazione? Si motivi brevemente la risposta.
 - (i) Il numero di variabili presenti nel programma.
 - (ii) Il numero di variabili presenti nei blocchi compresi tra quello di dichiarazione di x e quello in cui vi si accede.
 - (iii) Il numero di Record di Attivazione presenti sulla pila e compresi tra quello contenente la dichiarazione di x e quello in cui vi si accede;
 - (iv) Il numero di dichiarazioni di x presenti nei blocchi compresi fra il blocco più esterno contenente una x visibile e quello in cui si accede a x ;
 - (v) Il tempo è costante, e quindi indipendente da questi parametri.
 - (vi) Nessuna delle risposte precedenti.
5. Si dica cosa stampa il seguente frammento in uno pseudolinguaggio con passaggio per nome e scope statico.

```
int x = 10;
int y = 0;
void foo(name int v, name int z){
    int y = 10;
    v = 4;
    z++;
    if (z==v) write(x);
    else write (y);
}
foo(y,y);
write(x);
write(y);
```

6. Si consideri il seguente frammento in uno pseudolinguaggio con tipi statici, dove f è una certa funzione di due argomenti:

```
int i,j;
float y,z;
y = i + f(i,j);
z = f(y,i);
i = i+j;
```

Si forniscano (i) una possibile intestazione per la funzione f e (ii) le ipotesi che occorre fare sul sistema di tipi dello pseudolinguaggio affinché il frammento sia corretto, considerando l'intestazione definita in (i) e assumendo che l'operatore $+$ non sia overloaded.

7. Si dica cosa stampa il seguente frammento in un linguaggio con scope statico, passaggio per valore ed eccezioni:

```
int a = 6;

void g(){
    a--;
    write (a);
    throw X;
}

void f(int x){
    int a = 9;
    if (x==0) g();
    else try{ g(); } catch X { a--; write(a); f(0); }
}

try {f(1);} catch X {a--;}
write(a);
```

8. Si considerino le seguenti definizioni Java

```
interface I{
    void f();
    int s(int y);
}

interface J{
    void g();
}

class A implements I{
    int x;
    void f(){
        x++;
    }
    int s(int y){
        return x+x;
    }
}

class B extends A implements J{
    void g(){
        x--;
    }
    void f(){
        x--;
    }
}

A a = new B();
B b = (B) a;
a.x = 10;
b.x = 20;
a.f();
b.f();
System.out.println(a.x);
System.out.println(b.x);
```

- (i) Cosa eredita B da A? Cosa eredita da I? E cosa da J?
(ii) Cosa stampa il frammento?