

Tempo a disposizione: ore 2.

1. Si consideri la grammatica $G = (\{A, B, C\}, \{a, b, c\}, A, P)$ dove P è l'insieme seguente:

$$\begin{aligned} A &::= aA \mid bA \mid B \\ B &::= aBb \mid C \\ C &::= c \mid \epsilon \end{aligned}$$

Sia $L = \mathcal{L}(G)$ il linguaggio generato. Quali delle seguenti affermazioni sono vere?

- L contiene solo stringhe di lunghezza pari.
 - L contiene solo stringhe di lunghezza dispari.
 - L contiene la stringa vuota.
 - L non contiene la stringa $aabb$.
 - L contiene tutte le palindrome su a, b .
 - $L = \{a^n b^p a^m c b^m \mid m, n, p \geq 0\}$.
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 dica se l'espressione $\mathcal{I}_{L_1}^L(\mathcal{C}_{L_2, L_3}^{L_1}, \mathcal{C}_{L_2, L_3}^{L_2})$ ha senso; in caso positivo, cosa denota?
3. Si dica cosa stampa il seguente frammento in uno pseudolinguaggio con passaggio per nome:

```
int X[10];
int i = 1;
X[0] = 10;
X[1] = 10;
X[2] = 10;
void foo (name int Y, J){
    X[J] = J-1;
    write(Y);
    J++;
    X[J]=J;
    write(Y);
}
foo(X[i], i);
write(X[i]);
```

4. La definizione di certo linguaggio di programmazione specifica che la valutazione procede da sinistra a destra. Inoltre, nel valutare un'espressione complessa, eventuali sottoespressioni che vi compaiono più di una volta devono essere valutate una sola volta, usando il valore così calcolato anche per le altre occorrenze della stessa sottoespressione. Infine, un assegnamento è una particolare forma di espressione complessa. Si consideri il seguente frammento di codice:

```
int x = 1;
int A[5];
for (int i=0; i<5; i++)
    A[i] = i;
A[x++] = A[x++]+x;
```

Qual è lo stato del vettore A dopo l'assegnamento?

5. In un certo linguaggio di programmazione è presente un comando della forma `lswap N1, N2`, dove N1 e N2 sono due nomi che devono essere dotati di l-valore. L'esecuzione di `lswap N1, N2` scambia gli l-valori di N1 e N2. Cosa stampa il seguente frammento di codice?

```
int x = 1;
int y = 2;
lswap x, y;
x++;
y = x+1;
lswap x, y;
write(x, y);
```

6. Si consideri un linguaggio con scope statico, implementato mediante display, nel quale gli identificatori sono noti staticamente. Si assuma inoltre che ogni nome usato sia dichiarato in un solo blocco. Si prendano in considerazione le operazioni di (NL) = “accesso ad una variabile non locale x” e (L) = “accesso ad una variabile locale y” (nel contesto di un blocco). Per ognuna delle due operazioni di dica a quale dei seguenti parametri è proporzionale il tempo necessario all’esecuzione dell’operazione, motivando brevemente la risposta.

- (i) Il numero di variabili presenti nel programma;
- (ii) Il numero di variabili presenti nei blocchi compresi tra quello di dichiarazione della variabile e quello in cui si accede alla variabile stessa;
- (iii) Il numero di record di attivazione presenti sulla pila e compresi tra quello contenente la dichiarazione della variabile e quello in cui vi si accede;
- (iv) il numero di blocchi che contengono testualmente il blocco in cui si accede alla variabile e che sono contenuti in quello nel quale la variabile è dichiarata;
- (v) Il tempo è costante, e quindi indipendente da questi parametri;
- (vi) nessuna delle risposte precedenti.

7. È dato il seguente frammento di codice in uno pseudolinguaggio con `goto`, scope dinamico e blocchi annidati etichettati (indicati con `A :{...}`):

```
A: { int x = 5;
    int y = 4;
    goto C;
    B: {int x = 4;
        int z = 3;
        goto E;
    }
    C: {int x = 3;
        D: {int x = 2;
            }
        goto B;
    E: {int x = 1; // (**)
    }
}
```

Lo scope dinamico è gestito mediante tabella centrale dell’ambiente (CRT). Si illustri graficamente la situazione della CRT (esplicita, senza pila nascosta) nel momento in cui l’esecuzione raggiunge il punto segnato con il commento (**).

8. Si considerino le seguenti classi Java:

```
public class A {
    int x = 5;
    int fie () {return x;}
}

public class B extends A{
    int x = 0;
    int fie () { return x;}
}
```

Si consideri adesso il seguente frammento di codice:

```
B b = new B();
A a = b;
int zz = a.fie()+ a.x ;
```

Si dica qual è il valore di `zz` al termine dell’esecuzione del frammento.