

CORSO DI PARADIGMI DI PROGRAMMAZIONE
PROVA SCRITTA DEL 1 LUGLIO 2004.

Tempo a disposizione: ore 2.

1. Si diano almeno tre esempi di vincoli sintattici contestuali (detti anche vincoli di semantica statica), cioè vincoli sintattici non esprimibili mediante grammatiche libere.
2. Si consideri la seguente grammatica $G = (\{S, T\}, \{a, b, c\}, S, P)$ dove P è dato dalle seguenti produzioni:

$$\begin{aligned} S &::= aSa \mid bTb \mid T \\ T &::= bTb \mid cSc \mid c \end{aligned}$$

Si dia un albero di derivazione per la stringa “bbcaacaacbb”. Esiste una derivazione per “bbccccbb”? In caso positivo, si dia l’albero corrispondente; in caso negativo si dia un argomento che dimostra l’impossibilità.

3. Usando uno pseudolinguaggio che usi puntatori si fornisca un frammento di codice che generi un “dangling reference”. Si faccia quindi vedere come con la tecnica delle “tombstones” non si ha più tale problema.
4. 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. La primitiva `write(x)` permette di stampare un valore intero.

```
{int x = 2;

void pippo(value int y){
    x = x + y;
}

    { int x = 5;
      pippo(x++);
      write(x);
    }

write(x); }
```

(si ricordi che un comando della forma `foo(w++)`; passa a `foo` il valore corrente di `w` e poi incrementa `w` di uno).

5. Si dica brevemente (ma con precisione) quali sono le differenze tra un costrutto di iterazione controllato logicamente e un costrutto di iterazione controllato numericamente.
6. Si illustri graficamente come viene gestita la catena statica del seguente frammento di codice usando la tecnica del display (si assume che il linguaggio ammetta `goto` e blocchi annidati etichettati, indicati con $A : \{\dots\}$):

```
A: { int x = 5;
    goto C;

    B: {int x = 4;
        goto E }
    C: {int x = 3;

        D: { int x = 2}
            goto B;
        E: {int x = 1}
            }
    }
```

7. Sono date le seguenti definizioni Java:

```
interface A {
    int val = 1;
    int foo (int x);
}

interface B extends A {
    int val = 2;
}

class C implements B {
    int n = 0;
    public int foo (int x){ return x+val+n;}
}

class D extends C {
    int n;
    D(int v){n=v;}
    public int foo (int x){return x+val+n;}
}
```

Si consideri ora il seguente frammento di programma

```
int u, v, w, z;
A a;
B b;
C c;
D d = new D(3);
a = d;
b = d;
c = d;
u = a.foo(1);
v = b.foo(1);
w = c.foo(1);
z = d.foo(1);
```

Si dia il valore di u , v , w e z al termine dell'esecuzione.

8. **Solo per: corso AL; corso MZ a.a. 2002/03** Si dica quale sono le risposte calcolate dal seguente programma logico per il goal $p(X)$

$p(0)$.

$p(s(X)) :- q(X)$.

$q(s(X)) :- p(X)$.

9. **Solo per il corso MZ a.a. 2003/04:** Definiamo i λ -termini $\Omega \equiv (\lambda x.xx)(\lambda x.xx)$ e $I \equiv \lambda z.z$. (i) Si dica qual è la forma normale del seguente λ -termine:

$$((\lambda u.(\lambda v.u))I)\Omega$$

(ii) Qual è la forma normale se si usasse una valutazione *call by value*?