

Tempo a disposizione: ore 2.

1. Si forniscano tre esempi di vincoli sintattici contestuali e si dica brevemente come questi vengono controllati.
2. Con la notazione  $\mathcal{C}_{L_1, L_2}^L$  indichiamo un compilatore da  $L_1$  a  $L_2$  scritto in  $L$ . Con  $\mathcal{I}_{L_1}^L$  indichiamo un interprete scritto in  $L$  per il linguaggio  $L_1$ ; se  $P$  è un programma in  $L_1$  e  $x$  un suo dato,  $\mathcal{I}_{L_1}^L(P, x)$  indica l'applicazione dell'interprete a  $P$  e  $x$ . Si dica se la seguente scrittura ha senso

$\mathcal{I}_{L_1}^L(\mathcal{C}_{L, L_1}^L, \mathcal{C}_{L, L_1}^L)$ .

Se la risposta è "no", si motivi tale fatto; se è "sì" si dica qual è il risultato ottenuto.

3. Si dica cosa stampa il seguente frammento in uno pseudolinguaggio con passaggio per nome e scope statico

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

4. Il linguaggio imperativo Ric è costituito dagli usuali comandi (assegnamenti, controllo di sequenza ecc.), non permette comandi di allocazione (e deallocazione) esplicita della memoria, ammette funzioni, ma, nel caso di funzioni ricorsive, queste devono essere ricorsive in coda. Si dica, motivando la risposta, qual è la più semplice forma di gestione della memoria utilizzabile nell'implementazione di Ric.
5. Si consideri il seguente frammento in uno pseudolinguaggio con scope statico e parametri di ordine superiore:

```
{void foo (int f(), int n){
  int fie(){
    write(n);
  }
  if (n==0) f();
  else foo(fie,0);
}
int g(){
  write(10);
}
foo(g,1);
}
```

Si dica cosa stampa il frammento con (i) shallow binding; (ii) deep binding.

6. Sono date le seguenti dichiarazioni in un linguaggio ad oggetti con classi:

```
class A {
  int x=0;
  void f(){write(x);}
  void g(){f();}
}
class B extending A {
  int x=1;
  void f(){write(x);}
}
```

Si dica cosa stampa il seguente frammento, nel contesto di tali dichiarazioni:

```
B b = new B();
A a = b;
b.g();
b.f();
a.g();
a.f();
```

7. Si discuta brevemente la differenza esistente fra polimorfismo universale parametrico e polimorfismo universale di sottotipo.
8. **Solo corso AL** Sia  $P$  un programma logico e supponiamo che la valutazione del goal  $p(\mathbf{a})$  in  $P$  produca (almeno) una computazione che non termina. È possibile che la valutazione del goal  $p(\mathbf{X})$  produca solo computazioni che terminano? Se sì fornire un esempio di  $P$ , se no motivare brevemente la risposta. (Ricordiamo che le lettere minuscole indicano costanti e le maiuscole variabili).
9. **Solo corso MZ** Si considerino i seguenti lambda-termini:

$$I = \lambda x.x \quad S = \lambda y.y + y \quad K = \lambda u.(\lambda v.u)$$

Qual è il primo redex che viene ridotto nel termine

$$(I S)(K 3 I)$$

se si segue la strategia *applicativa (call by value)*?