

Tempo a disposizione: ore 2.

1. 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 . Infine se P^L è un programma scritto in L e x un suo dato di input, $\mathcal{I}_{L_1}^L(P^L, x)$ indica l'applicazione dell'interprete a P^L e x . Si dica se la seguente scrittura ha senso

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

Se la risposta è "no" si motivi tale fatto; se è "sì", si dica qual è il risultato ottenuto e se tale risultato sia ottenibile in modo più semplice (sotto le stesse ipotesi che permettono di ottenerlo con l'espressione precedente).

2. Sui terminali $\{0, 1\}$ e non terminali $\{A, B\}$ sono date due grammatiche, entrambe aventi A come simbolo iniziale: G_1 ha come produzioni

$$A ::= 0A0 \mid 0B \quad B ::= 1A$$

la seconda grammatica G_2 ha come produzioni

$$A ::= 0A0 \mid 1B \quad B ::= 1B \mid 1$$

Qual è il linguaggio generato da G_1 ? Qual è il linguaggio generato da G_2 ?

3. Cosa stampa il seguente frammento in un linguaggio con scope statico e passaggio dei parametri per riferimento?

```
int x = 4;
void f (reference int z){
    int x = 0;
    void g (reference int w){
        x = z + w + 1;
    }
    z++;
    g(z);
    x++;
}
f(x);
write(x);
```

4. Si considerino le seguenti definizioni in Java:

```
class A{
    int n;
}
class B{
    int foo (A x, A y) {
        y.n = 0;
        x.n = 1;
        if (x.n == y.n) return 1;
        else return 0;
    }
}
A a1 = new A();
A a2 = new A();
B b = new B();
int w = ****
```

Si dia un'espressione che utilizzi in modo non banale il nome `b` e che, sostituita al posto di `****`, assegni a `w` il valore 1.

5. Si consideri uno pseudolinguaggio con gestione delle eccezioni, scope statico e passaggio per valore. Cosa stampa il seguente frammento?

```

int x = 1;

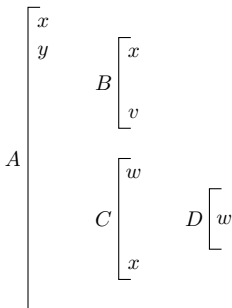
void g(int z){
    x = z+1;
    throw E;
    x = z+1;
}

void foo(int y){
    try{g(x+y);} catch E {write(x);}
}

{int x = 10;
  x = x+1;
  try {foo(x);} catch E {write(x);}
  write(x);
}
write (x);

```

6. Si consideri la struttura di blocchi schematizzata nella figura seguente; i nomi all'interno di un blocco indicano una dichiarazione di quel nome.



Si rappresenti graficamente l'ambiente per il blocco D di tale figura, dopo la sequenza di chiamate A,B,C,D, con scope dinamico realizzato mediante tabella centrale dell'ambiente (CRT), supponendo che, ove possibile, ogni chiamata venga chiusa prima della chiamata successiva.

7. Si fornisca una funzione f che non ha la ricorsione in coda. E' possibile scrivere una funzione equivalente a f che abbia la ricorsione in coda? Motivare la risposta.
8. Si supponga che un ipotetico analizzatore semantico per un linguaggio L prenda in input un programma P e un indirizzo di una locazione di memoria l . L'ipotetico analizzatore dovrebbe eseguire le seguenti operazioni: 1) controllare se il valore v , memorizzato nella cella di indirizzo l , è un valore di tipo reale oppure intero; 2) verificare che P non contenga identificatori che non siano stati prima dichiarati; 3) verificare che tutti e soli gli identificatori del programma P che a run time sono effettivamente usati sono stati prima dichiarati; 4) identificare tutti i comandi while del programma P che, con input v , producono un ciclo infinito. Si dica quali di queste operazioni sono possibili e quali no, motivando la risposta (se occorre, si facciano delle ipotesi sul linguaggio L).