

Tempo a disposizione: ore 2.

Svolgere gli esercizi 1-4 e 5-8 su due fogli differenti.

1. Nella seguente espressione

$$\mathcal{I}_{L_1}^{L_0}(\mathcal{C}_{L_0, L_1}^{L_1}, \mathcal{C}_{L_0, L_1}^{L_1})$$

l'interprete su quale macchina ospite si basa e quale la macchina astratta realizza? Quale risultato produce l'interprete?

2. Gli identificatori di un ipotetico linguaggio di programmazione sono definiti come sequenze nonvuote di lunghezza arbitraria di lettere o cifre, che cominciano con una cifra, contengono almeno una lettera e terminano con la cifra 0. Fornire una definizione regolare per questi identificatori.
3. Si consideri la seguente grammatica G con simbolo iniziale S :

$$\begin{aligned} S &\rightarrow BA \mid A \\ A &\rightarrow \mathbf{a} \mid \mathbf{a}A \\ B &\rightarrow \epsilon \mid \mathbf{b}CB \\ C &\rightarrow \mathbf{c}C \mid B \end{aligned}$$

(i) Si calcolino i First e i Follow per tutti i nonterminali. (ii) La grammatica G è di classe LL(1)? (iii) Si rimuovano le produzioni unitarie per ottenere una grammatica G' senza produzioni unitarie, che sia equivalente a G .

4. Si consideri la grammatica G con simbolo iniziale S :

$$S \rightarrow (S) \mid ()$$

(i) Costruire l'automa canonico LR(1). (ii) Costruire la tabella di parsing LR(1) e verificare se ci sono conflitti. (iii) Mostrare il funzionamento del parser LR(1) per l'input $((()))$.

5. Si considerino le seguenti definizioni di classe in Java:

```
class A{
    int x;
    int y;
    int f (int y){return y+1;}
    int g (int k){return k+2;}
}
class B extends A{
    int x;
    void g (int z){return y;}
    int f (int y){return y+3;}
    int h (int y){return y+3;}
}
class C extends B{
    int y;
    int f (int y){return y+3;}
    int m (int y){return y+3;}
}
```

Si supponga che la gerarchia delle classi sia implementata mediante vtable. Qual è la struttura della vtable di C ? Se nella classe A eliminiamo la definizione del metodo f cambia la vtable di C ? Motivare la risposta.

6. Si dica cosa stampa il seguente frammento di programma assumendo scope dinamico e shallow binding:

```
{int x = 10;
 int n = 100;
 int stampa(){
     write(x);
 }
 void foo (int f(), int g()){
     int x = 30;
     int n = 300;
     stampa()
     f();
     g();
 }
     {int x = 20;
      int n = 200;
      int ass_x(){
          x= x+n;
      }
      int stampa(){
          write(x - 1);
      }
      foo(ass_x(), stampa());
 }
 }
```

7. È dato il seguente frammento di codice in uno pseudolinguaggio con variabili a riferimento e garbage collection con contatori dei riferimenti:

```
type A = struct{
    int x;
    A next;
}
A foo(){
    A a = new A();
    A c = new A();
    a.next= new A();
    b.next = a;
    c.next = a;
    return a;
}

A u = foo();
u = foo();
```

- (i) Quanti oggetti di tipo A sono creati sullo heap? (ii) Per ciascuno di essi si dia il valore del contatore dei riferimenti al termine del frammento.
8. Un certo linguaggio adotta un'allocazione della memoria completamente statica. Quali delle seguenti caratteristiche *non sono compatibili* con tale politica di allocazione? (a) Blocchi annidati; (b) Definizione di funzioni annidate; (c) Definizione di funzioni ricorsive; (d) Passaggio dei parametri per valore; (e) Passaggio dei parametri per riferimento; (d) comandi allocazione esplicita della memoria.