

Tempo a disposizione: ore 2.

**Svolgere gli esercizi 1–4, 5–6 e 7–8 su tre fogli differenti.**

1. Siano date due macchine astratte  $M_0$  e  $M_1$ , con relativo linguaggio  $L_0$  e  $L_1$ , rispettivamente. Inoltre, siano dati un interprete  $\mathcal{I}_{L_2}^{L_0}$ , che realizza la macchina astratta  $M_2$  sopra la macchina ospite  $M_0$ , e un compilatore  $C_{L_2, L_1}^{L_2}$ . Utilizzando il software disponibile, è possibile implementare una macchina astratta per  $L_2$  sopra la macchina  $M_1$ ?
2. Definire una grammatica libera per  $L_1 = \{b^{2n}a^{2m} \mid n \geq m \geq 0\}$  ed una grammatica libera per  $L_2 = \{b^{2n+1}a^{2m+1} \mid n \geq m \geq 0\}$ . È vero che  $L_1 \cup L_2 = \{b^{n+k}a^{n-k} \mid n \geq k \geq 0\}$ ? Giustificare la risposta.
3. Si consideri la grammatica  $G$  con simbolo iniziale  $S$ :

$$\begin{aligned} S &\rightarrow SA \mid A \\ A &\rightarrow a \end{aligned}$$

(i) Determinare il linguaggio generato  $L(G)$ . (ii) Verificare se  $G$  sia di classe LL(1). (iii) Mostrare che  $G$  è di classe LR(0).

4. Si costruisca un parser LL(1) per il linguaggio  $L = \{a^n b^{n+1} \mid n \geq 0\}$  e si mostri il suo funzionamento su input  $abb$ .
5. Si consideri il seguente frammento in uno pseudolinguaggio con scope dinamico e parametri di ordine superiore:

```
int x = 7;
int n = 30;
void g(){
    write(n+x)
}
void foo (int f(), int n){
    if (n==0) f();
    else
    {int x = 20
    foo(f,0);
    g();
    }
}
{
    int x = 5;
    foo(g,1)
}
```

Si dica cosa stampa il frammento con shallow binding.

6. Un certo linguaggio adotta un'allocazione della memoria completamente statica. Quali delle seguenti caratteristiche *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 riferimento; (e) passaggio dei parametri per risultato; (d) comandi allocazione esplicita della memoria.

7. È dato il seguente frammento di codice in un linguaggio con variabili a riferimento e garbage collection con contatori dei riferimenti; nel linguaggio, `new` alloca dinamicamente memoria nello heap e l'espressione `x = e` assegna la valutazione di `e` a `x` e ne ritorna il valore:

```
type A = struct { A next; }
A f(){
  A a = new A();
  A b = a.next = new A();
  return a;
}
A g( A a ){
  A b = new A();
  return b.next = a;
}
A u = g( f() );
u = g( u.next );
```

(i) Quanti oggetti di tipo `A` sono creati sullo heap in totale? (ii) Per ciascuno di essi si dia il valore del contatore dei riferimenti al termine del frammento.

8. Cosa stampa il seguente frammento di codice Java?

```
class A {
  float a = 0;
  float a(){ return ++a; }
  float f(){
    if( ( a = a() + a ) < 27 ){
      System.out.println( a );
      return f();
    }
    return a;
  }
}

class B extends A {
  float a = 1;
  float a(){ return a++; }
}

new B().f();
```