

Tempo a disposizione: ore 2.

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

1. Si consideri il frammento di linguaggio di programmazione definito dalla seguente sintassi astratta:

$$b ::= \text{true} \mid \text{false} \mid x=y \mid b \text{ and } b$$

dove  $x$  e  $y$  sono due variabili intere. Definire le regole di semantica operativa strutturata per questo frammento, assumendo che gli operatori siano valutati con una disciplina di valutazione interna-destra.

2. Definire una grammatica  $G$  che generi il linguaggio  $L = \{a^n b^m c^m a^n \mid n \geq 0, m \geq 1\}$ . Il linguaggio  $L$  è libero, ma non regolare? Giustificare la risposta.
3. Si consideri la grammatica  $G$  con simbolo iniziale  $S$ :

$$\begin{aligned} S &\rightarrow SAB \mid \epsilon \\ A &\rightarrow a \mid aB \\ B &\rightarrow b \mid bA \end{aligned}$$

(i) Verificare che  $G$  non è di classe LL(1). (ii) Manipolare la grammatica  $G$ , rimuovendo la ricorsione sinistra immediata su  $S$ . (iii) Quindi fattorizzare le produzioni per  $A$  e  $B$ . (iv) Verificare se la risultante grammatica sia di classe LL(1); in caso positivo, costruire la tabella di parsing LL(1).

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

$$\begin{aligned} S &\rightarrow \text{bSc} \mid \text{bAc} \\ A &\rightarrow \text{aA} \mid \text{a} \end{aligned}$$

(i) Che linguaggio genera  $G$ ? (ii) Si costruisca il parser SLR(1). (iii) Si mostri il funzionamento del parser su input  $baac$ .

5. Si consideri il seguente frammento in uno pseudolinguaggio con parametri di ordine superiore:

```
{
int n = 10;
int x = 20;
int g(){
    return 1;
}
void foo (int f(), int n){
    int h(){
        return n;
    }
    if (n==10)    {write(f( ));
                  }
                {foo(h,n-10);
                  }
    }
    foo(g,x);
}
```

Si assuma scope statico, deep binding e passaggio di parametri per valore. Si dica cosa stampa il frammento motivando la risposta.

6. Usando uno pseudolinguaggio che ammetta l'uso dei 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. Avremmo potuto usare la tecnica del "mark and sweep" invece delle "tombstones" per risolvere questo problema? Motivare la risposta.
7. Si considerino le seguenti definizioni di classe in Java:

```
class A{
    int x = 10;
    int f (int y){return g(y);}
    int g (int k){return -k ;}
}
class B extends A{
    int x;
    int y;
    int g (int z){return z;}
    int p (int z){return g(y);}
}
class C extends B{
    int x = 100;
    int g (int k){return k*x;}
    int h (int y){return y+3;}
}
C oggi = new C;
A pippo = oggi;
int w = pippo.f(3)+ pippo.x
```

Si dica che valore viene assegnato a w motivando la risposta. Si supponga che la gerarchia delle classi sia implementata mediante lista concatenata. Si mostri la rappresentazione dell'implementazione dell'oggetto oggi.

8. Si discutano brevemente le differenze principali esistenti nei meccanismi di computazione di base dei paradigmi imperativo, funzionale e logico.