

Tempo a disposizione: ore 2.

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

**Per Paradigmi: svolgere solo: 1,5,6,7,8.**

1. Una grammatica libera  $G$  con non terminali  $\{S, A, B\}$  ( $S$  è il simbolo iniziale) e terminali  $\{a, b\}$  contiene le seguenti produzioni

$$\begin{aligned} A &\rightarrow aA \mid \epsilon \\ B &\rightarrow AB \end{aligned}$$

Quali produzioni devono essere *aggiunte* affinché  $\mathcal{L}(G) = \{ba^n b \mid n \geq 0\}$ ?

2. Si dia il DFA minimo che riconosce il linguaggio definito dall'espressione regolare  $ab^*(ab \mid b^*a)a^*$ .
3. Si consideri la grammatica (simbolo iniziale  $S$ ; solite convenzioni per terminali/non terminali):

$$\begin{aligned} (1) \quad S &\rightarrow aDC \mid C \\ (2) \quad B &\rightarrow aBb \mid b \\ (3) \quad C &\rightarrow cS \mid \epsilon \\ (4) \quad D &\rightarrow bD \mid \epsilon \end{aligned}$$

Si dica se si tratta di una grammatica LL(1).

4. Si dia l'automa canonico LR(0) per la grammatica aumentata

$$\begin{aligned} 0 \quad S' &\rightarrow S \\ 1 \quad S &\rightarrow aScS \\ 2 \quad S &\rightarrow \epsilon \end{aligned}$$

Si tratta di una grammatica SLR(1)?

5. Si consideri il seguente frammento di codice in uno pseudo-linguaggio che ammetta passaggio dei parametri per riferimento e per nome.

```
int[] V = new int[5];
int n=0;

int f (reference int x) {
    return x++; }

void foo(reference int x, name int y){
    x++; y++; x++; y++;}

V[0]=V[1]=V[2]=V[3]=V[4]=1;

foo(V[f(n)], V[f(n)]);
```

Si dia lo stato del vettore V al termine dell'esecuzione del codice esposto (si ricordi che un comando della forma `return w++;` restituisce il valore corrente di `w` e poi incrementa `w` di uno).

6. Si assuma di avere uno pseudolinguaggio che adotti la tecnica del *reference count*; se *OGG* è un generico oggetto nello heap, indichiamo con *OGG.ref-c* il suo reference count (nascosto). Si consideri il seguente frammento di codice:

```
C foo = new C(); // oggetto OG1

C bar = new C(); // oggetto OG2

C fie = foo;

bar = fie;
```

Si dia il valore di *OG1.ref-c* e *OG2.ref-c* dopo l'esecuzione del frammento. Quali di questi due oggetti possono essere restituiti alla lista libera?

7. Si scriva un frammento di codice, nel linguaggio di programmazione preferito, che generi un *dangling reference*.
8. È dato il seguente frammento di codice in uno pseudolinguaggio con scope statico gestito mediante display

```
void P1(){
    void P2(){
        corpo di P2
    }
    void P3(){
        void P4(){
            corpo di P4
        }
        corpo di P3
    }
    corpo di P1
}
```

Si descriva graficamente il display dopo la successione di chiamate (tutte attive) P1, P3, P4, P2, P3. Si descriva inoltre brevemente come viene determinato il puntatore al record di attivazione di P2 (da inserire nel display), quando P2 è chiamata da P4.