

Tempo a disposizione: ore 2.

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

Per Paradigmi: svolgere solo: 2 (escluso iv), 5, 6, 7, 8.

1. Si consideri il linguaggio $L = \{w \in (0|1)^* \mid |w| \geq 2 \text{ ed il primo e l'ultimo simbolo di } w \text{ sono } 1\}$. Si mostri una semplice espressione regolare che denota L . Si produca un DFA che riconosca L (non è necessario partire dall'espressione regolare).
2. Si consideri il linguaggio $L = \{a^i b^j c^k \mid i = 2j \vee j = 2k, \text{ con } i, j, k \geq 0\}$. (i) La stringa vuota ϵ appartiene a L ? (ii) Elenca tutte le stringhe $w \in L$ tali che $|w| \leq 4$. (iii) Definisci una grammatica libera G tale che $L(G) = L$. (iv) Tale grammatica G è di classe $LL(1)$?
3. Si consideri la seguente grammatica G con simbolo iniziale S :

$$\begin{aligned} S &\rightarrow AB \mid D \\ A &\rightarrow \epsilon \mid Ab \mid aB \\ B &\rightarrow b \mid bB \\ C &\rightarrow c \\ D &\rightarrow aD \end{aligned}$$

Si effettuino le seguenti manipolazioni su G , nell'ordine specificato, per ottenere una grammatica equivalente G' semplificata. (i) Si eliminino i simboli inutili. (ii) Si rimuova la ricorsione sinistra presente sul nonterminale A . (iii) Si eliminino le produzioni nulle ϵ . (iv) Infine, si rimuovano le produzioni unitarie.

4. Costruire un parser $LR(0)$ per il linguaggio $L = \{a^n b^n \mid n \geq 1\}$ e si discuta il suo comportamento sugli input $aabb$ ed ϵ .

5. Si dica cosa viene stampato dal seguente frammento di codice scritto in uno pseudo-linguaggio che usa scoping dinamico e passaggio di parametri per nome. La primitiva `write(x)` permette di stampare un valore intero.

```
{int x = 2;
void pippo(name int y, name int z){
    x = x + y++ + z++;
    write(y)
}
{ int x = 5;
  pippo(x,x);
  write(x);
}
write(x)}
```

6. Si dica brevemente (ma con precisione) quali sono le differenze tra un costrutto di iterazione controllato logicamente e un costrutto di iterazione controllato numericamente.
7. La definizione di certo linguaggio di programmazione specifica che la valutazione procede da sinistra a destra. Inoltre, nel valutare un'espressione complessa, eventuali sottoespressioni che vi compaiono più di una volta devono essere valutate una sola volta, usando il valore così calcolato anche per le altre occorrenze della stessa sottoespressione. Infine, un assegnamento è una particolare forma di espressione complessa. Si consideri il seguente frammento di codice:

```
int x = 1;
int A[5];
for (int i=0; i<5; i++)
    A[i] = i;
A[x++] = A[x++] + x;
```

Qual è lo stato del vettore A dopo l'assegnamento?

8. Si dica cosa viene stampato dall'esecuzione della classe Java `SubClasse`, definita come segue:

```
class SuperClasse {
    public String s = "SUPER";
    public void stampa() {
        assegna();
        System.out.println(s);
    }
    public void assegna() {
        s = "SUPERSOPRA";
    }
}

class SubClasse extends SuperClasse {
    public String s = "SUB";
    public void assegna() {
        s = "SUPERSOSOTTO";
    }

    public static void main(String[] args) {
        SubClasse sub = new SubClasse();
        SuperClasse sup = sub;
        sup.stampa();
        sub.stampa();
        System.out.println(sup.s);
        System.out.println(sub.s);
    }
}
```