

Tempo a disposizione: ore 2.

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

1. Costruire il più semplice automa (se possibile) che riconosca il linguaggio  $\{a^k a^n a^k \mid k \geq 0, n \geq 0\}$ . È regolare tale linguaggio?
2. Si dimostri che il linguaggio  $L = \{a^{2^n} b^n \mid n \geq 0\}$  è libero deterministico, costruendo un DPDA che riconosca  $L$  per stato finale.
3. Si costruisca un parser LL(1) per il linguaggio  $L$  del punto precedente e si mostri il suo funzionamento su input  $aab$ .
4. Si costruisca un parser SLR(1) per il linguaggio  $L$  del punto 2 e si mostri il suo funzionamento su input  $aab$ . È possibile costruire un parser LR(0) per  $L$ ?
5. Si consideri un linguaggio con scope dinamico, implementato mediante CRT, nel quale tutti gli identificatori sono noti staticamente. Si prendano in considerazione le operazioni di “accesso ad una variabile non locale  $x$ ” e “accesso ad una variabile locale  $y$ ” (nel contesto di un blocco). Per ognuna delle due operazioni si dica a quale dei seguenti parametri è proporzionale il tempo necessario all’esecuzione dell’operazione, motivando brevemente la risposta.
  - (i) Il numero di variabili presenti nel programma;
  - (ii) Il numero di variabili presenti nei blocchi compresi tra quello di dichiarazione della variabile e quello in cui si accede alla variabile stessa;
  - (iii) Il numero di record di attivazione presenti sulla pila e compresi tra quello contenente la dichiarazione della variabile e quello in cui vi si accede;
  - (iv) Il tempo è costante, e quindi indipendente da questi parametri;
  - (v) nessuna delle risposte precedenti.
6. Un certo linguaggio è basato su questi tre principi: (i) *scoping statico*; (ii) *dichiarazioni implicite*: la prima volta in un blocco che un nome compare a sinistra dell’operatore di assegnamento (=) viene creata una nuova associazione per quel nome (legato al valore dell’espressione a destra dell’assegnamento); (iii) tutte le associazioni create in una funzione sono *locali* a quella funzione. Supponendo di adottare questi tre principi, si dica cosa stampa il seguente frammento:

```
A = 10;
B = 1;
void foo(){
    B = A+1;
    write(B);
}
void fie(){
    A = 5;
    B = 5;
    foo();
}
fie();
write(A,B);
```

7. Si consideri il seguente frammento di programma scritto in uno pseudo-linguaggio che usi scoping dinamico, passaggio per nome e dove la primitiva `read(Y)` permette di leggere nella variabile `Y` un intero dall'input standard, mentre `write(X)` permette di stampare il valore della variabile `X`.

```
int X = 0;
int Y;
void pippo(int Z name) {
    X++;
}
void pluto(int Z name) {
    int W;
    W = Z++;
    W = Z++;
    X = Z
    pippo;
}
read(Y);
if Y > 0 then { int X = 5;
                pluto(X);
            }
            else { pluto(X);
            }
write(X);
```

Si dica quali sono i valori stampati motivando brevemente la risposta.

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

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

class SubClasse extends SuperClasse {
    public String s = "SUB";
    public void stampa() {
        System.out.println(s);
    }
    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);
    }
}
```