

Tempo a disposizione: ore 2.

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

1. Si consideri la seguente grammatica $G = (\{S, T\}, \{a, b\}, S, P)$ dove P è dato dalle seguenti produzioni:

$$\begin{aligned} S & ::= SaS \mid TbT \mid T \\ T & ::= TaT \mid S \mid a \end{aligned}$$

Si mostri che G è ambigua.

2. Dare il DFA minimo che riconosce il linguaggio definito dalla seguente espressione regolare:

$$(ab \mid (ba)^*)ab^*$$

3. Dare un automa a pila deterministico che riconosce il linguaggio $L = \{a^n cb^{2n} \mid n \geq 0\}$. Esiste un DFA che riconosce L ? Motivare.
4. Verificare se la seguente grammatica (i caratteri a, b sono simboli terminali) è LALR(1):

$$\begin{aligned} (0) \quad S' & \rightarrow S \\ (1) \quad S & \rightarrow AaAb \\ (2) \quad S & \rightarrow BbBa \\ (3) \quad A & \rightarrow \varepsilon \\ (4) \quad B & \rightarrow \varepsilon \end{aligned}$$

5. Si consideri il seguente frammento in uno pseudolinguaggio con tipi statici, dove f è una certa funzione di due argomenti:

```
int i,j;
float y,z;
y = f(i,j);
z = f(y,i);
```

Si fornisca (i) una possibile intestazione per la funzione f e (ii) le ipotesi che occorre fare sul sistema di tipi dello pseudolinguaggio affinché l'intestazione data in (i) sia corretta.

6. Facendo riferimento a uno o più linguaggi di programmazione reali, si fornisca un esempio di due tipi di dato che sono equivalenti strutturalmente ma non per nome. Si forniscano quindi due esempi di tipi che sono compatibili ma non equivalenti.

7. Cosa stampa il seguente frammento di codice in un linguaggio con scope statico e passaggio per riferimento? L'ordine di valutazione delle espressioni influenza il risultato? Perché?

```
{int x = 1;
int A[5];
int foo () {return x++;}
for (int i=0, i<5, i++)
  A[i]=i;
int f(reference int a){
  int x = 5;
  a = x+a;
  a = x+a;
  return a;
}
write (f(A[foo()+x]) + A[x] + x);
}
```

8. Sono date le seguenti definizioni di funzione:

```
int fact_falso (int n){
  if (n == 0) return 1;
  else return fact_falso(n+1);
}
int fact_vero (int n){
  if (n == 0) return 1;
  else return n*fact_vero(n-1);
}
```

Una certa implementazione del linguaggio si comporta nel modo seguente. Alla chiamata `fact_falso(1)`, non risponde, rimanendo in un loop infinito. Alla chiamata `fact_vero(-1)` risponde dopo qualche tempo con `Stack overflow during evaluation`, abortendo l'esecuzione. Si dia una spiegazione motivata di questi due fatti.