Corso di Linguaggi di Programmazione
Prova scritta del 24 luglio 2012.

Tempo a disposizione: ore 2.
Svolgere gli esercizi 1-4 e 5-8 su due fogli differenti.

1. Si dia la definizione di linguaggio generato da una grammatica $G = (NT, T, S, P)$.

2. Con la notazione $\mathcal{C}^{L}_{L_1,L_2}$ indichiamo un compilatore da L_1 a L_2 scritto in L. Con $\mathcal{I}^{L}_{L_1}$ indichiamo un interprete scritto in L per il linguaggio L_1; se P è un programma in L_1 e x un suo dato, $\mathcal{I}^{L}_{L_1}(P, x)$ indica l'applicazione dell'interprete a P e x. Si dica se la valutazione di $\mathcal{I}^{L}_{L_1}(\mathcal{C}^{L_1}_{L_1,L_2}, \mathcal{C}^{L_1}_{L_1,L_2})$ produce un qualche risultato motivando la risposta.

3. Si dia il DFA minimo che riconosce il linguaggio definito dall'espressione regolare $a(a \mid ab)^*(ba)^*$.

4. Si consideri la grammatica aumentata

$$\begin{array}{lrcl} (1) & S' & \rightarrow & S \\ (2) & S & \rightarrow & Ab \\ (3) & S & \rightarrow & Bc \\ (4) & A & \rightarrow & aA \\ (5) & A & \rightarrow & \epsilon \\ (6) & B & \rightarrow & aB \\ (7) & B & \rightarrow & \epsilon \end{array}$$

   Mostrare che non è SLR(1) (usare l'argomento più economico). Verificare se è LL(1), dando esplicitamente l'automa.

5. Si consideri la seguente definizione di tipo record:

```
type S = struct{
   int x;
   char y;
   };
```

Si supponga che un `int` sia memorizzato su 2 byte, un char su 1 byte, su un'architettura a 16 bit con allineamento alla parola. In un blocco viene dichiarato un vettore:

```
S A[10];
```

Indicando con `PRDA` il puntatore all'RdA di tale blocco, e con `ofst` l'offset tra il valore di `PRDA` e l'indirizzo iniziale di memorizzazione di `A`, si dia l'espressione per il calcolo dell'indirizzo dell'elemento `A[4].y`.

6. Si dica cosa viene stampato dal seguente frammento in un linguaggio con eccezioni:

```
void f() throws X {
   throw new X();
   }

void g (int sw) throws X {
   if (sw == 0) {f();}
   try {f();} catch (X e) {write("in␣g");}
   }
...
try {g(1);}
   catch (X e)  {write("in␣main");}
```

7. Si consideri il seguente frammento di programma scritto in uno pseudo-linguaggio che usa scope statico.

```
{
void f() {
        void g() {
        corpo_di_g;
        }

        void h() {
                void l(){
                corpo_di_l;
                }
        corpo_di_h;
        }
corpo_di_f;
}
```

Si descriva graficamente l'evoluzione del display nella sequenza di chiamate f, g, h, l, g, supponendo che tutte le chiamate rimangano attive (ossia nessuna funzione ha restituito il controllo).

8. Si consideri il seguente frammento in uno pseudolinguaggio con scope dinamico e parametri di ordine superiore:

```
int x = 700;
int n = 30;
void g(){
    write(n+x)
}
void foo (int f(), int n){
    if (n==0) f();
    else foo(f,0);
    g();
}
 {
  int x = 5;
  foo(g,1)
 }
```

Si dica cosa stampa il frammento con (i) shallow binding; (ii) deep binding.