Corso di Linguaggi di Programmazione — Paradigmi di Programmazione
Prova scritta del 5 febbraio 2013.

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

$$\begin{array}{ll} A \rightarrow & aA \mid \epsilon \\ B \rightarrow & bB \mid \epsilon \end{array}$$

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

2. Si dia il DFA minimo che riconosce il linguaggio definito dall'espressione regolare $a(a \mid b)^* c^*$.

3. Si consideri la grammatica aumentata (simbolo iniziale S; solite convenzioni per terminali/non terminali):

$$\begin{array}{lrcl} (0) & S' & \rightarrow & S \\ (1) & S & \rightarrow & CB\texttt{bb} \\ (2) & S & \rightarrow & \texttt{aab} \\ (3) & S & \rightarrow & \texttt{b}B\texttt{a} \\ (4) & B & \rightarrow & \texttt{a} \\ (5) & C & \rightarrow & \texttt{c}C \\ (6) & C & \rightarrow & \texttt{c} \end{array}$$

   Si dica se si tratta di una grammatica è LL(1), motivando adeguatamente.

4. Per la stessa grammatica dell'esercizio 3, si dica se si tratta di una grammatica SLR(1), motivando adeguatamente.

5. 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. Infine se P^L è un programma scritto in L e x un suo dato di input, $\mathcal{I}^{L_1}_L(P^L, x)$ indica l'applicazione dell'interprete a P^L e x. Si dica sotto quali ipotesi la seguente scrittura descrive un'esecuzione corretta (di uno o piu' programmi) e quale è il risultato di tale esecuzione.

   $\mathcal{I}^L_{L_1}(\mathcal{I}^{L_1}_L, (\mathcal{C}^L_{L,L_1}, P^L))$.

6. Si dica cosa è l'aliasing e perché può rendere difficoltosa la verifica della correttezza dei programmi. E' possibile inserire nel compilatore di un ipotetico linguaggio un controllo che permetta di identificare tutte le situazioni di aliasing ? Perché ?

7. Si dica cosa viene stampato dal seguente frammento di codice scritto in uno pseudo-linguaggio che usa scoping dinamico e deep binding:

```
int x = 3;
procedure ass_x(n:int)
    {x = n;
    }
procedure stampa_x
    {write_integer(x);
    }
procedure pippo(function S, P ; int n )
   { int x= 10;
    if n=1  then ass_x(n)
            else S(n);
    P;
    stampa_x
    }

{ int x = 5;
  pippo(ass_x, stampa_x, 1);
  pippo(ass_x, stampa_x, 2);
}
```

8. Sono date, in Java, le seguenti dichiarazioni di classi:

```
class A{
   int a = 10;
   void f(){g();}
   void g(){a=5;}
   }

class B extends A{
   int a = 7;
   int b = 2;
   void g(){b=15;}
   }
```

   Si dica cosa stampa, nello scope di queste dichiarazioni, il seguente frammento:

```
A y = new A();
B x = new B();
x.f();
y=x;
System.out.print(x.a);
System.out.print(y.a);
System.out.print(x.b);
```