Corso di Paradigmi di Programmazione
Prova scritta del 6 Giugno 2006.

Tempo a disposizione: ore 2.

1. Si dia la definizione di grammatica ambigua e si spieghi brevemente perché la grammatiche ambigue non sono adatte a descrivere i linguaggi di programmazione.

2. La tabella che segue riporta la semantica operazionale dei comandi di un semplice linguaggio imperativo (coincide con la Figura 2.14 del testo).

$$\langle \textbf{skip}, \sigma \rangle \to \sigma \quad (c1)$$

$$\langle X := n, \sigma \rangle \to \sigma[X \leftarrow n] \quad (c2) \qquad \frac{\langle a, \sigma \rangle \to \langle a', \sigma \rangle}{\langle X := a, \sigma \rangle \to \langle X := a', \sigma \rangle} \quad (c3)$$

$$\frac{\langle c_1, \sigma \rangle \to \sigma'}{\langle c_1; c_2, \sigma \rangle \to \langle c_2, \sigma' \rangle} \quad (c4) \qquad \frac{\langle c_1, \sigma \rangle \to \langle c_1', \sigma' \rangle}{\langle c_1; c_2, \sigma \rangle \to \langle c_1'; c_2, \sigma' \rangle} \quad (c5)$$

$$\langle \textbf{if tt then } c_1 \textbf{ else } c_2, \sigma \rangle \to \langle c_1, \sigma \rangle \quad (c6) \qquad \langle \textbf{if ff then } c_1 \textbf{ else } c_2, \sigma \rangle \to \langle c_2, \sigma \rangle \quad (c7)$$

$$\frac{\langle b, \sigma \rangle \to \langle b', \sigma \rangle}{\langle \textbf{if } b \textbf{ then } c_1 \textbf{ else } c_2, \sigma \rangle \to \langle \textbf{if } b' \textbf{ then } c_1 \textbf{ else } c_2, \sigma \rangle} \quad (c8)$$

$$\langle \textbf{while } b \textbf{ do } c, \sigma \rangle \to \langle \textbf{if } b \textbf{ then } c; \textbf{while } b \textbf{ do } c \textbf{ else skip}, \sigma \rangle \quad (c9)$$

Si descriva la computazione corrispondente al comando

$$\textbf{while } \neg(X == Y) \textbf{ do } X := X + 1;$$

nello stato $\sigma = [(X, 4), (Y, 3)]$. Le espressioni aritmetiche e booleane possono essere valutate con il loro significato intuitivo.

3. In un linguaggio con array dinamici (di forma fissata al momento di elaborazione della dichiarazione), sono date le seguenti dichiarazioni:

```
type T = struct{
           int x;
           char c;
           }
T V[0..n, 0..m];
```

(la seconda dichiarazione introduce la variabile `V` come un array a due dimensioni con elementi di tipo `T`). Al momento in cui la dichiarazione di `V` è elaborata si ha `n=20` e `m=10`. Si dia: (i) la struttura del *dope vector* per `V`; (ii) L'offset dell'elemento `V[2,5]` rispetto all'inizio del vettore, supponendo che gli interi siano memorizzati su due byte, i caratteri su un byte, e l'architettura richieda allineamento alla parola di 16 bit.

4. Si consideri la seguente definizione di funzione

```
int f(int n){
   if (n==0) return 1;
   else return n + f(n-1);
```

Si fornisca la definizione di una funzione `g(n)` che calcoli lo stesso valore di `f` e che possa essere calcolata usando a run-time un numero costante di record di attivazione, indipendentemente dal valore di `n`.

5. Facendo riferimento ad un qualsiasi (pseudo-)linguaggio di programmazione si faccia un esempio di una funzione che esibisce polimorfismo universale parametrico e di una che esibisce polimorfismo universale di sottotipo.

6. Si dica cosa stampa il seguente frammento in uno pseudolinguaggio con passaggio per riferimento e scope dinamico. Ai fini dell'esercizio si assuma che un parametro attuale della forma `x++` corrisponda allo l-valore di `x` e, contemporaneamente, all'incremento del suo r-valore di di uno.

```
int x = 4;
void foo(name int y) {
     int x = 6;
     int w;
     x = x + y;
     w = y;
     write(x);
     write(y);
}
{ int x = 10;
   foo(x++);
write(x);
}
```

7. (i) Si descriva la struttura delle vtable corrispondenti alle seguenti dichiarazioni di classi (si assuma ereditarietà singola):

```
class A{
   int a = 1;
   int f(int n){return n+1;}
}
class B extending A{
   int a = 2;
   int g(){return f(3);}
   }
```

(ii) Supponiamo di compilare le classi A e B. Se successivamente la classe A è modificata come segue

```
class A{
   int a = 1;
   int h(){return 4;}
   int f(int n){return n+1;}
}
```

possiamo ricompilare solo A e continuare a usare la classe B già compilata? Motivare la risposta.

8. **Solo corso A-L** Si consideri il seguente programma logico

```
member (X, [X| Xs).
member(X,  [Y|Xs]):- member(X, Xs).
```

dove il costruttore `[X | Xs]` indica la lista con testa X e coda Xs. Si dica quale è il risultato della valutazione del goal `member(g(X), [ f(1), h(2), g(3)] )` (dove `[f(1), h(2), g(3)]` indica la lista che contiene i tre elementi `f(1), h(2), g(3)`.

9. **Solo corso MZ** Si considerino i seguenti lambda-termini:

$$I = \lambda x.x \qquad S = \lambda y.y + y \qquad K = \lambda u.(\lambda v.u)$$

Si diano i passi di riduzione in strategia *applicativa (call by value)* del seguente termine:

$$(I\ S)(K\ 3\ I)$$