

Prova Scritta di Linguaggi di Programmazione I

21/09/2006

1. Con riferimento al seguente programma in Pascal, si rappresenti il P-code relativo al corpo della procedura **tr**, cioè il P-code corrispondente alla sezione di codice all'interno del riquadro.

```
program esercizio;  
    ...  
    const N = ...;  
    type matrix = array[ 0..N, 0..N ] of real;  
    ...  
    procedure swap( var x, y: real );  
        var t: real;  
    begin t := x; x := y; y := t end;  
  
    procedure tr( k: integer; var M: matrix );
```

```
        var i: integer;  
    begin  
        if k > 0 then  
            begin  
                tr( k-1, M );  
                for i := 0 to k do  
                    swap( M[k,i], M[i,k] )  
                end  
            end  
        end;  
    end;
```

```
begin ... end.
```

2. Sia $PE_{L_1}^{L_2}$ un valutatore parziale di L_2 scritto in L_1 e $I_{L_1}^{L_2}$ un interprete di L_2 scritto in L_1 . Sia $\llbracket P \rrbracket$ la funzione calcolata dal programma P . Si dica cosa produce la valutazione $\llbracket PE_{L_0}^{L_0} \rrbracket(PE_{L_0}^{L_0}, I_{L_0}^{L_1})$.
3. Si dica se le seguenti grammatiche sono ambigue (mostrando un testimone dell'ambiguità oppure argomentando opportunamente sulla non ambiguità). Si determini inoltre se sono equivalenti.
 1. $S ::= S S \mid (S) \mid ()$
 2. $S ::= A \mid A S \quad A ::= () \mid (S)$
4. Si mostri l'output del seguente frammento di programma in un linguaggio C-like con assegnamento che calcola r -value *prima* di l -value, valutazione argomenti chiamate da sinistra a destra e indici vettori iniziati da 0:

```
#define ind(e) ( (y[e--])++ )  
  
{  
    char x[10] = il_proprio_cognome;  
    int y[3] = {0,2,1};  
    int i = 2;  
    char mess(int j, ref char z)  
    {  
        char c = x[ind(j)] = x[ind(j)];  
        write(y[0]+y[1]+y[2], z, i+j);  
        return c;  
    }  
    write(mess(i--, x[i--]));  
    write(x[i--], i);  
}
```

5. Assumendo di utilizzare nell'Esercizio 8 la tecnica di implementazione con shallow binding, si mostri con un diagramma la situazione sullo stack nascosto e nel vettore centralizzato per le variabili del sistema, al momento immediatamente successivo alla prima chiamata di Q.

Prova Scritta di Linguaggi di Programmazione I

21/09/2006

6. Si scriva una funzione `colaltsums :: (Ord a) => [[a]] -> [a]` che, data una matrice implementata come liste di liste per righe, calcola il vettore delle somme a segni alternati delle colonne della matrice.

$$\text{Detto } s_j = \sum_{i=1}^n (-1)^{i+1} a_{ij}, \text{ colaltsums} \left(\begin{pmatrix} a_{11} & \dots & a_{1m} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nm} \end{pmatrix} \right) = (s_1 \quad \dots \quad s_m)$$

Si scriva il programma con variabili anonime ove possibile.

7. Si scriva una funzione Haskell `shiftToZero` che dato un BST t secondo la seguente definizione di tipo

```
data (Ord a) => BST a = Void
    | Node a (BST a) (BST a)
```

costruisce un nuovo BST isomorfo che contiene gli elementi t diminuiti del valore minimo di t .

A titolo di esempio,

```
shiftToZero (Node 5 (Node 2 Void Void) Void)
```

riscrive a

```
Node 3 (Node 0 Void Void) Void.
```

Si dia espressamente il tipo di `shiftToZero`.

8. Si mostri l'output del seguente frammento di programma espresso in un linguaggio C-like con scoping *dinamico*, *shallow binding* e valutazione delle espressioni da sinistra a destra:

```
{ int x = 5;
  int y = 7;
  proc P(ref int y, valueresult int z, int R(name int))
  { z += y — + R(++x + ++y);
    write(x,y,z);
    z -= R(z++ + —y);
  }
  { int x = 3;
    int Q(name int w)
    { return (w + y++);
    }
    P(x, y, Q);
    write(y++,x++);
  }
  write(y,x);
}
```

5, 5, 21, -8, 5, -7, 5