

Prova Scritta di Linguaggi di Programmazione I

27/03/2006

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

```
program esercizio;
```

```
... ..
```

```
function f( n: integer ) : integer; ...
```

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

```
procedure p( x: integer; var y: integer );
```

```
begin
```

```
  y := f(f(x));
```

```
  if (y = 0) and (x > 0) then
```

```
    p(y+f(x-1),y)
```

```
end;
```

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

2. Sia $C_{L_1}^{L_2 \rightarrow L_3}$ un compilatore da L_2 a L_3 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 $\llbracket I_{L_0}^{L_1} \rrbracket(C_{L_1}^{L_2 \rightarrow L_1}, \llbracket I_{L_0}^{L_1} \rrbracket(C_{L_1}^{L_3 \rightarrow L_2}, C_{L_3}^{L_5 \rightarrow L_4}))$ e se tale risultato può effettivamente esistere.
3. Si stabilisca se la grammatica

$$S ::= S0 \mid R1$$
$$R ::= R0 \mid T1$$
$$T ::= T0 \mid 1 \mid S1$$

è ambigua. Si caratterizzi inoltre il linguaggio generato.

4. Si mostri l'output del seguente frammento di programma espresso in un linguaggio C-like con assegnamento che calcola l'r-value *dopo* l'l-value e indici dei vettori che iniziano da 0:

```
{
  char x[10] = il_proprio_cognome;
  int i = 1;
  char magic(int j, ref char y)
  {
    char c = y;
    x[j++] = x[j++] = x[j++];
    write(y, j);
    return c;
  }
  write(magic(0, x[++i]));
  write(x[i], i);
}
```

5. Con riferimento alla tecnica di implementazione con tecnica di 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 chiamata di Q ed al passaggio dei parametri per Q (cioè immediatamente prima della esecuzione del comando **return**).
6. Data la seguente definizione del tipo di dato astratto (polimorfo) *Matrici* **type** Matrix a = $\llbracket a \rrbracket$ si scriva un predicato Haskell diagonal :: Matrix a -> Bool che determina se una matrice (quadrata) è diagonale. Si implementino le matrici per righe come liste di liste.

Si scriva il programma con variabili anonime ove possibile.

Prova Scritta di Linguaggi di Programmazione I

27/03/2006

7. Data la seguente definizione del tipo di dato astratto (polimorfo) *Binary Search Tree*

```
data (Ord a) => BST a =  
  Void | Node a (BST a) (BST a)  
  deriving (Eq, Ord, Read, Show)
```

si scriva una funzione Haskell `limitedVisit` che dato un albero e due valori x, y costruisce la lista (ordinata) degli elementi dell'albero compresi nell'intervallo di valori da x a y .

Si dia espressamente il tipo di `limitedVisit`.

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++);  
  }  
  { int x = 3;  
    int Q(name int w)  
    { return (w + x++ + y++);  
    }  
    P(x, y, Q);  
    write(y++,x++);  
  }  
  write(y,x);  
}
```

6, 6, 19, 32, 8, 33, 5
