Prova Scritta di Linguaggi di Programmazione I

21/12/2005

1. Sia $C_{L_1}^{L_2 \to L_3}$ un compilatore da L_2 a L_3 scritto in L_1 e $I_{L_4}^{L_5}$ un interprete di L_5 scritto in L_4. Sia $[\![P]\!]$ la funzione calcolata al programma P. Si dica cosa produce la valutazione $[\![I_{L_4}^{L_5}]\!](C_{L_5}^{L_4 \to L_5}, I_{L_4}^{L_5})$ e se tale risultato può effettivamente esistere.

2. Si mostri che la grammatica $S \to SS \mid (S) \mid ()$ è ambigua. Si dia una grammatica non-ambigua che genera lo stesso linguaggio.

3. Si scriva, in un qualsiasi pseudolinguaggio, un frammento di codice tale che il numero massimo dei RdA presenti a run-time sulla pila non sia determinabile staticamente.

4. Si mostri l'output del seguente frammento di programma espresso in un linguaggio C-like con scoping *statico*:

```
{   int x[10] = #il_proprio_cognome+spazi_per_riempire#;
    int i = 1;
    void ganjaman(ref int y, ref int j)
    {   x[j] = x[j+1];
        write(y);
        j++;
        x[j] = x[j+1];
        write(y);
    }
    ganjaman(x[i], i);
    write(x[i]);
}
```

5. Si mostri l'output del seguente frammento di programma espresso in un linguaggio C-like con scoping *statico*:

```
{   int x = 3; int y = 4;
    proc P(ref int y, result int z, name int w)
    {   int x = 4;
        y = ++x + y++;
        write(y+x,w);
        z = w + 3;
        write(y++);
    }
    P(x, y, x+y);
    write(y,x);
}
```

> In questo caso il risultato è indipendente dall'ordine di valutazione delle espressioni e dall'ordine di calcolo di *l*-value e *r*-value nell'assegnamento.
>
> L'output è $13, 12, 8, 15, 9$.

6. Con riferimento alla tecnica di implementazione con catena statica e "display", per l'esercizio precedente si mostri (schematicamente) con dei diagrammi la situazione sul display e sullo stack di sistema al momento in cui viene eseguita P.

7. Data la seguente definizione del tipo di dato astratto (polimorfo) *Matrici* **type** Matrix a = [[a]] si scriva un predicato Haskell lowertriangular :: Matrix a -> **Bool** che determina se una matrice (quadrata) è triangolare inferiore. Si implementino le matrici come liste di liste, per righe o per colonne a seconda delle preferenze.

   A titolo di esempio, `lowertriangular([[1,0,0],[1,1,0],[1,1,1]])` restituisce `True`, mentre `lowertriangular([[0,0,1],[1,1,0],[1,1,1]])` restituisce `False`.

   Si scriva il programma con variabili anonime ove possibile.

Prova Scritta di Linguaggi di Programmazione I

21/12/2005

8. 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 `filtertree` p t che costruisce una lista (ordinata) di tutti gli elementi dell'albero t che soddisfano il predicato p. Si dia espressamente il tipo di `filtertree`. (Potrebbe esserci una soluzione elegante che passa per la definizione di una funzione più generale su BST)