Prova Scritta di Linguaggi di Programmazione I

07/09/2006

1. Con riferimento al seguente programma in Pascal, si rappresenti il P-code relativo al corpo della procedura `t`, cioè il P-code corrispondente alla sezione di codice all'interno del riquadro.

```
program esercizio;
    ...    ...
  procedure t( d, k: integer; var n: integer );
```

```
    var x, y, z: integer;
  begin
    if k = 0 then
      begin
        if d > 0 then n := 0 else n := 1
      end
    else
      begin
        t(k, n-1, x); t((k+3) mod 4, n-1, y); t((k+1) mod 4, n-1, z);
        n := 2*x + y + z
      end
  end;
```

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

2. Siano $PE^{L_2}_{L_1}$ un valutatore parziale di L_2 scritto in L_1, $C^{L_2 \to L_3}_{L_1}$ un compilatore da L_2 a L_3 scritto in L_1 e $I^{L_2}_{L_1}$ un interprete di L_2 scritto in L_1. Sia $[\![P]\!]$ la funzione calcolata dal programma P. Si dica cosa produce la valutazione $[\![I^{L_1}_{L_0}]\!](C^{L_2 \to L_1}_{L_1}, PE^{L_2}_{L_2})$.

3. Si mostri che la seguente grammatica, ideata per generare una rappresentazione lineare di alberi binari di numeri, è ambigua. ($*$ per l'albero vuoto)

$$T ::= * \mid T\ N\ T$$
$$N ::= C \mid N\ C \qquad\qquad C ::= \mathtt{0} \mid \ldots \mid \mathtt{9}$$

   Se ne costruisca una variante, sempre per generare rappresentazioni lineari di alberi, non-ambigua (non necessariamente equivalente).

4. Si mostri l'output del seguente frammento di programma in un linguaggio C-like con assegnamento che calcola *l*-value *dopo* *r*-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 = x[j++] = x[j++];
      write(c);
      x[--j] = c = x[j--];
      write(y, j);
      return c;
  }
  write(magic(1, x[i--]));
  write(x[i], i);
}
```

5. Assumendo di utilizzare nell'Esercizio 8 la tecnica di implementazione con catena statica e "display", si mostri schematicamente la situazione sul display e sullo stack di sistema al momento in cui viene eseguita la seconda Q.

Prova Scritta di Linguaggi di Programmazione I

07/09/2006

6. Si scriva una funzione `colminmax` :: (**Ord** a) =>[[a]] −> [a] che, data una matrice implementata come liste di liste per righe, calcola il vettore delle coppie (minimo,massimo) delle colonne della matrice.

   Si scriva il programma con variabili anonime ove possibile.

7. Si scriva un predicato Haskell `isRBT` che dato un albero colorato secondo la definizione di tipo seguente determina se è ben formato:

   - ogni nodo contiene un valore non minore dei valori del suo sottoalbero sinistro e minore dei valori del sottoalbero destro;
   - tutti i cammini dalla radice a una foglia hanno lo stesso numero di nodi Black;
   - i nodi Red devono avere genitore Black;
   - la radice è Black.

   ```
   data (Ord a) ⇒ RBT a = Void | Node a Color (RBT a) (RBT a)
   data Color = Red | Black
   ```

   Si dia espressamente il tipo di `isRBT`.

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

```
{ int x = 1; int y = 2;
  proc P(ref int y, name int z, int R(int, result int)) {
      y += x−− + R(z, x);
      write(x,y);  }
  {   int x = 3;
      int Q(int w, result int z) {
          z= x++ * w;
          return (w + y++);  }
      P(x, y++ + −−x, Q);
      write(−−x,y−−);
  }
  {   int y = 5;
      int Q(int w, result int z) {
          z= x++ * w;
          return (w + y++);  }
      P(x, y++ + −−x, Q);
      write(−−x,y−−);
  }
  write(x,y);
}
```

8, 11, 10, 4, 33, 33, 32, 7, 32, 3