

Prova Scritta di Linguaggi di Programmazione I

13/07/2007

Si noti che quanto messo nei riquadri è una *bozza* fornita solo a *titolo indicativo*.
Quindi **non** è un modello di soluzione completa che ci si aspetta ad un esame.

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

```
program esercizio;  
  ...  
  const N = ...;  
  var v: array [ 0 .. N ] of integer;  
  ...  
  procedure inssort( n: integer );  
  
    procedure ins( x, i: integer );  
    begin  
      while v[i] > x do  
        begin v[i+1] := v[i]; i := i-1 end;  
      v[i+1] := x  
    end;
```

```
begin  
  if n = 0 then  
    v[0] := -maxint  
  else  
    begin inssort( n-1 ); ins( v[n], n-1 ) end  
  end;
```

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

2. Sia $PE_{L_1}^{L_2}$ un valutatore parziale di L_2 scritto in L_1 , $C_{L_1}^{L_2 \rightarrow L_3}$ 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 $\llbracket PE_{L_0}^{L_1} \rrbracket (I_{L_1}^{L_2}, C_{L_2}^{L_3 \rightarrow L_4})$.
3. Sia G la grammatica individuata dalle produzioni

$$S ::= A a \mid A b$$
$$A ::= a \mid b \mid a A a \mid b A b$$

pensate per generare il linguaggio $L := \{wx \mid xw = w^R x, x \in \{a, b\}, w \in \{a, b\}^*\}$, dove w^R è la stringa w rovesciata. Si diano le stringhe di L di lunghezza ≤ 3 e un esempio dell'albero di parsing di una stringa generata per sbaglio ($\in \mathcal{L}(G) \setminus L$). Qualora non fosse possibile dare un esempio si giustifichi il perché.

```
{w ∈ L | |w| ≤ 3} = {a, b, aa, bb, aaa, abb, baa, bbb}.  
ab ∈ ℒ(G) \ L.
```

4. Si mostri l'evoluzione delle variabili e l'output del seguente frammento di programma in un linguaggio C-like con assegnamento che calcola l -value prima di r -value, valutazione argomenti chiamate *da destra a sinistra* e indici vettori iniziati da 0:

```
char x[10] = {il_proprio_cognome};  
int i=0, k=3, y[3] = {3,2,0};
```

Prova Scritta di Linguaggi di Programmazione I

13/07/2007

```
char mess(int i, ref char z) {
  x[++y[--i]] = x[--y[i--]] = x[y[i]++];
  write(y[0]+y[1]+y[2], z, k--i--);
  return x[y[i]];
}
write(mess(k++, x[++i]));
write(x[--i], k);
```

5. Utilizzando nell'Esercizio 8 la tecnica di implementazione con catena statica e display, si mostri schematicamente la situazione sul display e sullo stack di sistema quando entra in esecuzione F per la seconda volta.

6. Rappresentando QuadTrees con il tipo di dato

```
data Eq a => QT a = C a |
  Q (QT a) (QT a) (QT a) (QT a)
```

si scriva un predicato Haskell `elem_or_mele` che dati un QuadTree t e una lista di QuadTrees ts determina se t , o il QuadTree che codifica l'immagine di t ribaltata rispetto all'asse orizzontale, sono elementi della lista ts .

Si usino variabili anonime quando possibile.

Si dia esplicitamente il tipo di tutte le funzioni.

```
elem_or_mele x xs = any p xs
  where
    p y = y==x || y==flipHorizontal x
con flipHorizontal soluzione dell'Esercizio apposito dell'eserciziario.
```

7. Si scriva una funzione Haskell `overColor` che dato un colore c ed un QuadTree q determina il numero (minimo) di pixel dell'immagine codificata da q che hanno un colore maggiore di c . Ad esempio

```
let d = C 2; u = C 1; q = Q d u u u
in overColor 1 (Q q (C 0) (C 3) q)
```

restituisce 6 (visto che il QuadTree codifica almeno 16 pixel).

Si usino variabili anonime quando possibile.

Si dia esplicitamente il tipo della funzione.

8. Si mostri l'evoluzione delle variabili e l'output del seguente frammento di programma espresso in un linguaggio C-like con scoping *statico*, *deep binding*, assegnamento che calcola r -value *dopo* l -value e valutazione delle espressioni da sinistra a destra:

```
{
  int x = -3, y = 1;
  int Q(ref int v, name int y) {
    int w = y;
    v -= F(w, v-- + x);
    write(--v);
    return (--x + w);
  }
  write(Q(x, x++ + F(x, y++)));
}
int x = 5, y = 7;
int F(valueresult int v, name int u) {
  v += --x; write(x,v);
  v += u; write(x,v);
  return v++;
}
```

Prova Scritta di Linguaggi di Programmazione I

13/07/2007

```
}  
write(x,y);
```

| |
|--------------------------------------|
| 4, 2, 4, 3, 3, 3, 3, 10, -7, 3, 3, 7 |
|--------------------------------------|